

Gamberi

R. Hoffmann

Wikibooks.org

5 gennaio 2013

Il 28 aprile 2012 i contenuti dei progetti Wikibooks e Wikipedia in inglese e in tedesco sono stati concessi in licenza con licenza Creative Commons Attribution-ShareAlike 3.0 Unported. Un URI di questa licenza è fornito nell'elenco delle figure a pagina 435. Se questo documento è un lavoro derivato dal contenuto di uno di questi progetti e il contenuto era ancora concesso in licenza dal progetto con questa licenza al momento della derivazione questo documento deve essere concesso in licenza con la stessa licenza, una licenza simile o compatibile, come indicato nella sezione 4b della licenza. L'elenco dei contributori è incluso nel capitolo Collaboratori a pagina 433. Le licenze GPL, LGPL e GFDL sono incluse nel capitolo Licenze a pagina 441, poiché questo libro e / o parti di esso possono o non possono essere concesse in licenza in base a uno o più di questi licenze, e quindi richiedono l'inclusione di queste licenze.
UN.TE Software per impaginazione X. Lun.TE Il codice sorgente X è incluso come allegato (source.7z.txt) in questo file PDF. Per estrarre la fonte dal file PDF, si consiglia di utilizzare <http://www.pdflabs.com/tools/pdfkit-the-pdf-toolkit/> o facendo clic sul simbolo dell'allegato della graffetta in basso a sinistra del visualizzatore PDF, selezionando Salva allegato. Dopo averlo estratto dal file PDF devi rinominarlo in source.7z. Per decomprimere l'archivio risultante si consiglia di utilizzare <http://www.7-zip.org/>. L^{UN}.TE Lo stesso codice sorgente X è stato generato da un programma scritto da Dirk Hünniger, disponibile gratuitamente con una licenza open source di http://de.wikibooks.org/wiki/Benutzer:Dirk_Huenniger/wb2pdf. Questa distribuzione contiene anche una versione configurata del compilatore pdflatex con tutti i pacchetti e i caratteri necessari per compilare il file Lun.TE Sorgente X inclusa in questo file PDF.

Contenuti

0.1	Cosa sono i gamberi?	2
0.2	Perché Gambas?	2
0.3	Come puoi imparare i gamberi?	3
0.4	Scarica Gambas.	3
0.5	Installazione di Gambas.	3
0.6	Descrizione dettagliata.	4 °
0.7	Iniziare i gamberi.	7 °
0.8	Ciao mondo!	10
		13
0.9	Secondo programma a Gambas.	°
	Terzo programma: programmi nella finestra del terminale senza grafica.	14
0.10	th
0.11	Come posso copiare il codice del programma da questo corso?	16
0.12	Crea file eseguibile.	16
		17
0.13	Posiziona il programma sul desktop.	°
	Assegnare al programma la propria icona (simbolo del programma).	18
0.14	..	°
		18
0.15	Avvio del programma.	°
		19
0.16	Fine del programma.	°
0.17	Programma di stallo.	21
0.18	Funzione di aiuto in dotazione.	21
0.19	Aiuto su Internet.	24
		25
0.20	Domande aperte.	th
		25
0.21	Cursore alla fine di un testo?	th
0.22	Come posso scrivere più variabili in un file e, se necessario (ad es.	26
	Avvio del programma) leggere di nuovo?	th
0.23	Come puoi usare gli appunti per salvare la grafica?	27
0.24	Come posso interrogare quale scheda è attiva in una tabstrip?	27
0.25	Cosa significa la h in molti esempi di GambasCode?	27
0.26	Come posso scoprire la dimensione dei caratteri?	27
0.27	Come puoi dividere una riga in un editor e rimetterla insieme?	28
0.28	Come puoi programmare una funzione in modalità Terminal?	28
	Come puoi scrivere in una casella di testo usando solo lettere maiuscole?	
0.29	28
		30
0.30	Riduci la finestra di output diretto.	th
		30
0.31	Crea una nuova directory, copia la directory.	th
0.32	Come puoi interrogare la home directory dell'utente corrente?	34
0.33	Come si utilizza il comando DateDiff?	35
0.34	Virgolette attorno a un testo.	35

0.35	Gambas Editor, funzioni non documentate.	36
0.36	Indice di tabulazione in gambas.	36
0.37	File zip.	37
0.38	Elaborazione della query con Gambas. Come funzionano i tasti di scelta rapida?	37
0.39	.	38

0.40 Converti immagine in immagine e viceversa.	38
Come puoi assegnare di nuovo il contenuto di una DrawingArea a un'immagine?	
0.41	39
0.42 Segno dell'euro in Gambas.	39
0.43 Puoi dichiarare in Gambas come in VB con i caratteri \$ ecc?	39
La riga IF non viene compilata su una riga. Come mai?	
0.44	39
0.45 Sostituisci comando.	39
0.46 Sistema di aiuto programmabile?	40
Ricorda la finestra di dialogo del percorso.	
0.47	40
Database: variabili con il nome del campo invece del nome del campo?	
0.48	40
0.49 Gamberi.	41
0.50 Tutorial di base.	42
0.51 Vorrei creare un indice.	43
0.52 Taglia linee o stringhe dal file.	43
0.53 Il Gambas Wikibook come libro HTML sul tuo computer.	45
0.54 Programma Gambas.	46
0.55 Chiama tramite Konqueror.	47
0.56 Finestra in Gambas.	47
0.57 Passare da una finestra all'altra con ALT + TAB.	53
0.58 Raccolta di strumenti.	53
0.59 Il menu Gambas.	56
0.60 I tasti funzione in Gambas.	59
I controlli più importanti (raccolta strumenti / cassetta degli attrezzi).	
0.61	59
0.62 Pulsante di comando (pulsante).	67
0.63 Area di testo.	67
Barra di avanzamento = barra di avanzamento.	
0.64	67
0.65 Il timer (timer, timer).	69
0.66 Il ColumnView.	69
0.67 Lo slider.	69
0.68 Modulo	70
0.69 La vista a griglia.	70
0.70 Tableview.	70
0.71 Visualizzazione elenco.	70
0.72 Il pulsante di comando.	70
0.73 Numeri fino a 27.	76
0.74 Le proprietà, i metodi e gli eventi dell'area di testo.	77
0.75 Come puoi saltare alla fine del testo nella TextArea con il cursore? .	78
0.76 Stile di testo Gambas.	78
0.77 Il programma timer più semplice.	80
Modifica il ritardo in fase di esecuzione.	
0.78	80
0.79 Modifica il ritardo con numeri casuali.	82
0.80 Conto alla rovescia.	82
0.81 Testare il tempo di risposta.	84
0.82 Il comando wait, un'alternativa al timer.	84
0.83 ScrollBar (dispositivo di scorrimento).	85
0.84 Programma di esempio.	86
0.85 Come si crea un nuovo modulo?	86
0.86 Semplice in materia di forma.	87

0.87 Chiedi	88
-------------------	----

0.88	Modulo di avvio.	89
0.89	Forma modale.	89
0.90	Proprietà di una forma.	89
0.91	Metodi di una forma.	90
0.92	Eventi di una forma.	90
0.93	Costanti della forma.	90
0.94	Ordine di tabulazione del modulo.	90
0.95	Nuova forma.	90
0.96	Trasferimento dei dati da un modulo a un altro modulo.	91
0.97	Variabili globali in Gambas.	91
0.98	Copia il layout del modulo.	93
0.99	Ridimensiona un'area di testo in una forma.	95
	Modifica della larghezza di un campo di testo quando si modifica la larghezza	
0.100	del modulo.	96
0.101	Generale.	97
0.102	Programma di esempio.	97
0.103	Teoria di GridView.	98
0.104	Riempì GridView con i valori.	98
0.105	Imposta il colore di sfondo per qualsiasi cella.	100
0.106	Carica il componente TableView.	101
0.107	Usa TableView.	101
0.108	Modifica celle.	102
0.109	Celle colorate.	102
0.110	Naviga con i tasti cursore.	102
0.111	Tabella Wiki.	103
0.112	Messaggio semplice.	105
0.113	Messaggio con diverse possibili risposte.	105
0.114	Introduzione.	107
0.115	Espressioni aritmetiche.	107
0.116	Panoramica dei simboli matematici in Gambas.	113
0.117	Rappresentazione numerica.	113
	Conteggio in avanti, conto alla rovescia, troncatura, arrotondamento ecc. INC,	
0.118	DEC, INT, FRAC.	116
0.119	Segno e valore assoluto Sgn, Abs.	117
0.120	Il comando Val: crea qualcosa di significativo da un testo.	120
0.121	Calcola il numero di Eulero.	122
0.122	Calcola la sequenza di Fibonacci.	122
0.123	Calcola la sequenza di Lucas.	123
0.124	Programmare il simbolo della somma å.	124
0.125	Contare i numeri.	128
0.126	Rappresentazione funzionale.	130
0.127	Funzioni trigonometriche nei gambas.	130
0.128	Logica.	132
0.129	Calcola i numeri primi.	132
0.130	Logaritmo.	133
0.131	Conto interessi.	139
0.132	Statistiche descrittive.	144
0.133	Distribuzione gaussiana.	150
0.134	Testing.	150
0.135	Entropia per sequenze 01 semplici.	166

0.136 Calcoli casuali con la funzione casuale.	179
1 DrawingArea, Immagine	189
1.1 Le coordinate di DrawingArea1.	190
fotogr	
2 afie	191
3 Colori	193
4 ° Funzioni matematiche	195
5 Punto, linea, cerchio	197
5.1 Punti	197
5.2 Linee.	198
5.3 Rettangolo.	201
5.4 Cerchio.	202
5.5 Grafico a torta.	208
5.6 Ellisse.	208
6 ° La stella marina	213
6.1 Disposizione.	214
6.2 Il codice	214
6.3 Picture, Image, Drawingarea, Picturebox ecc.	216
6.4 Area grafica (DrawingArea).	216
6.5 Campo immagine (Picturebox).	216
6.6 Campo di scorrimento.	216
Carica un'immagine in una casella delle immagini.	
6.7	218
6.8 Converti un'immagine in un'immagine.	219
Carica un'immagine con finestra di dialogo in una casella delle immagini.	
6.9	219
Carica un'immagine nel Picturebox senza finestra di dialogo.	
6.10	220
Salva immagine senza finestra di dialogo.	
6.11	221
Salva immagine con finestra di dialogo.	
6.12	221
6.13 Ottieni un'immagine dagli appunti.	221
6.14 Carica l'immagine negli appunti.	221
6.15 Ridimensiona.	221
6.16 Memorizzato nella cache.	222
6.17 Ricaricare.	222
6.18 Immagine dello schermo	222
6.19 Colori	222
6.20 Gamma di colori	224
6.21 Come usare MouseDown, MouseMove e MouseUp.	225
6.22 Le coordinate del mouse.	227
6.23 Trascina e rilascia, prendi e trascina con il mouse.	229
6.24 Interroga i pulsanti del mouse.	231
Definire una forma diversa del cursore utilizzando le proprietà.	
6.25	231
6.26 Il mouse casuale.	232
Un'altra immagine del cursore con il comando del cursore.	
6.27	233
6.28 Domanda.	234

6.30 L'immissione di testo.	237
6.31 Elimina testo.	238
6.32 Incolla il testo insieme a &.	238
6.33 Nuova riga con & Chr \$ (10) e Chr \$ (13).	239
Etichetta di testo in grassetto, corsivo e colorato.	
6.34	239
6.35 Il comando di divisione.	240
6.36 Il comando Sostituisci: sostituisci i caratteri.	242
6.37 Instr Qual è la posizione della parte?	245
6.38 RInStr.	246
6.39 Segnaposto secondario nel testo.	246
Determina la lunghezza di una stringa (una sequenza di testo) Len.	
6.40	247
6.41 CHR () Converte il codice numerico ASCII in caratteri.	247
6.42 ASC () Converte i caratteri alfanumerici in codici numerici ASCII.	248
6.43 Str () converte altre variabili in una stringa di testo.	249
6.44 Set di caratteri.	249
Seleziona il carattere tramite una finestra di dialogo standard.	
6.45	250
6.46 Elenca tutti i caratteri disponibili.	250
6.47 Testo negli appunti.	250
6.48 Teoria degli appunti.	252
6.49 Come puoi copiare la grafica negli appunti e farla uscire di nuovo? .	252
6.50 Come posso modificare gli appunti di Klipper (KDE)?	253
6.51 Panoramica dei comandi.	253
6.52 Comando di attesa (pausa).	253
6.53 Tempo	254
6.54 Data.	260
6.55 Chiedi	263
6.56 Salvare il file	265
6.57 Salva file senza finestra di dialogo.	266
6.58 Carica file 1.	266
6.59 Carica file 2.	267
Carica il file con la finestra di dialogo standard.	
6.60	267
6.61 Carica e salva i file con Addyplus.	268
6.62 Il tuo comando.	270
6.63 Crea una nuova directory.	270
6.64 Copia il file.	271
6.65 Come puoi interrogare la home directory dell'utente corrente?	272
6.66 Esiste un file?	273
6.67 Carica e salva il file in un piccolo programma combinato.	273
6.68 Scrivi file con WRITE.	275
6.69 Leggi i dati con READ.	275
6.70 Finestre di dialogo standard.	277
6.71 Teoria.	277
Modifica le dimensioni di una finestra di dialogo.	
6.72	278
6.73 Primo menu.	278
6.74 Menu delle opzioni.	281
6.75 Menu dinamico.	282
6.76 Menù contestuale.	283
6.77 Programmazione database solo con Gambas.	284

6.78	File CSV.....	285
6.79	Programmazione di database con Gambas e MySQL.....	287
6.80	Programmazione di database con Gambas e SQLite.....	291
6.81	Crea una connessione al database.....	291
6.82	Mostra i database disponibili.....	292
6.83	Sinistra	293
6.84	Poche righe in stampa diretta.....	293
6.85	Stampa con finestra di dialogo.....	294
6.86	Stampa in orientamento orizzontale.....	295
6.87	Stampa un'immagine.....	295
6.88	Stampa su un file PS Postscript.....	295
6.89	Stampa Annulla.....	295
6.90	Teoria delle classi di stampanti.....	295
6.91	Printer.Print.....	297
6.92	Browser HTML.....	297
6.93	Browser avanzato.....	299
6.94	Browser web di Gambas.....	300
6.95	Esempio di codice per l'interfaccia RS 232.....	300
6.96	È sotto i gamberi.....	301
6.97	wc = conteggio parole sotto Gambas.....	302
6.98	Carica il componente di compressione.....	303
6.99	Programma di esempio.....	303
6.100	Ci sono diversi giochi negli esempi forniti.....	304
6.101	Le torri di Hanoi.....	305
6.102	Sprint un mini-gioco per 2.....	305
6.103	TicTacToe.....	307
6.104	Sinistra	309
6.105	Convertitore (testo in codice HTML).....	309
6.106	Youtube Downloader.....	310
6.107	Aggiunta di tutorial.....	312
6.108	Sottrai (sottrai) tutorial.....	314
6.109	Tutorial sulla moltiplicazione.....	315
6.110	Tutorial per i caratteri di confronto> <ed è lo stesso.....	317
	L'insegnante di matematica di Harry.....	
6.111	.	319
6.112	Scarica un file.....	319
6.113	Calcolatrice avanzata.....	320
6.114	Calcolo del canone di locazione.....	324
6.115	Loop semplice.....	327
6.116	Espandi loop.....	327
6.117	Analisi dei gas nel sangue.....	327
6.118	Diagnosi ECG.....	331
6.119	Esame radiografico del torace.....	334
6.120	Diagnosi ECG a lungo termine.....	334
6.121	Rapporto di endoscopia.....	334
6.122	Diagnosi ecografica.....	334
6.123	Risultati di ammissione.....	334
	Breve lettera di dimissione del medico.....	
6.124	334
6.125	Diagnosi TC.....	334

6.126 Tempo QT.....	334
6.127 Conversione frequenza cardiaca - distanza RR.....	334
6.128 Gambas e SVG.....	334
6.129 Blights.....	335
6.130 Serpente..... Programma di navigazione in rete.....	336
6.131	336
6.132 Connessione al database.....	336
6.133 Apostrofo come segno di commento.....	336
6.134 Prove di programma con commenti.....	337
6.135 Messaggio semplice.....	338
6.136 Messaggio con diverse possibili risposte.....	339
6.137 Teoria dell'assegnazione.....	341
6.138 Tipi di dati semplici.....	341
6.139 Tipi di dati composti.....	342
6.140 Conversione dei tipi di dati.....	342
6.141 Molto facilmente	342
6.142 Sinistra	343
6.143 Variabili.....	343
6.144 val	352
6.145 str \$.....	354
6.146 asc.....	355
6.147 chr \$.....	356
6.148 int.....	357
6.149 aggiustare.....	358
6.150 cosiddetto	359
6.151 bst.....	359
6.152 cbool.....	359
6.153 cbyte.....	360
6.154 cdate.....	360
6.155 cfloat.....	360
6.156 cstr.....	360
6.157 cint.....	360
7° Matrici semplici	361
Riempি più velocemente con il comando array.....	
7.1	363
7.2 Routine di array semplici.....	363
7.3 Errori con gli array.....	365
7.4 Split >> Divide un testo in un array di stringhe con molti singoli elementi. . .	367
7.5 Join Il comando join.....	368
7.6 Per ciascuno..... Routine di array Aggiungi, Rimuovi, Ridimensiona, Ripristina, Cancell, 7.7 Ordina.....	369
7.8 Riempি la matrice con numeri casuali.....	373
7.9 Numeri della lotteria (7 su 49). . .	374
7.10 Getta i doppelganger fuori dall'array.....	375
7.11 Teoria.....	376
8° Tipi di array	377

9	Crea una matrice bidimensionale	379
10	Routine di array	381
10.1	Raccolta di materiali e domande sugli array	381
10.2	Indice dei comandi alfabetici	382
10.3	E e)	383
10.4	O o)	384
10.5	XOR	384
10.6	NON	385
10.7	Operatori bit per bit	385
10.8	Panoramica delle strutture di Gambas	386
10.9	Rami	388
10.10	Macinare	391
11/10	Ordinamenti	395
10.12	Generale	395
10.13	Un esempio molto semplice	395
10.14	Esempi di funzioni :	396
10.15	Il programma funziona da solo	400
10.16	Costanti predefinite	406
10.17	Ripplesort	408
11	Eventi di controllo	413
12	° Programma eventi con EVENT, STOP EVENT, RAISE	415
12.1	Cattura gli errori con Catch	416
12.2	Individua gli errori con TRY	418
12.3	F11	419
	Accesso diretto nella finestra diretta senza un programma	
12.4	419
12.5	Uscita nella finestra diretta con il comando PRINT	419
	Programmazione senza grafica solo nella finestra diretta	
12.6	420
12.7	Domanda:	421
12.8	Tipi di dati	421
12.9	Tipi di file	422
12.10	Costanti stringa (testo)	422
11/12	Tipi di ordinamento	422
12.12	Giorni della settimana	422
12.13	Formati numerici predefiniti	422
12.14	Formati di data e ora predefiniti	423
12.15	Diversi formati	423
12.16	Più costanti	423
12.17	Costanti	423
12.18	Sequenze di fuga	424
12.19	Componenti GB	424
12.20	Come funziona Gambas?	428
12.21	Differenze tra Visual Basic e Gambas	430
	Panoramica delle estensioni di file a confronto :	
12.22	430
12.23	Importazione di moduli VB	431
12.24	Variabili globali in Gambas	431

X

12.25 IsNumber.	431
Comando di scaricamento non disponibile in Gambas.	
12.26 . . .	432
13	
° Autori	433
Lista delle figure	435
14	
th Licenze	441
14.1 LICENZA PUBBLICA GENERALE GNU.	441
14.2 GNU Free Documentation License.	442
14.3 GNU Lesser General Public License.	442

Dopo alcuni anni di sviluppo, Gambas è il primo ambiente di sviluppo simile a VB per Linux che offre una versione 2.0 completamente sviluppata.

La base di questo libro sono i mini-programmi Gambas che utilizzo per il mio sito web¹ creato. Dopo aver scoperto i Wikibook, ho deciso di pubblicarli anche qui.

Questi mini programmi si sono dimostrati molto efficaci sin dai tempi di Visual Basic. Questo rende l'apprendimento del linguaggio di programmazione molto più semplice. È disponibile anche un semplice libro di consultazione per aiutare. Chiunque sia interessato dovrebbe essere in grado di imparare facilmente i gambas senza molta conoscenza preliminare. Ovviamente, tutti possono aggiungere nuovi capitoli, modificare e migliorare quelli esistenti. Posso ancora imparare molto a Gambas e ne sono grato!

Cerca sempre di trovare la soluzione in assoluto più semplice e didatticamente più comune per un problema. Solo allora si dovrebbero fornire soluzioni un po' più complicate come ulteriori esempi. Questo è molto utile per i principianti perché consente loro di comprendere il nucleo di una soluzione di programmazione e di non essere distratti da inutili zavorre dall'esterno. Per i programmatore esperti, potrebbe esserci del codice banale nel libro. Ciò che ad alcuni sembra banale, è l'inizio di una vera comprensione per gli altri.

Per rendere appetibile il linguaggio Gambas al programmatore in erba, questo libro inizia con piccoli programmi applicativi. La teoria più complicata viene dopo. All'inizio servono alcune storie di successo visibili e discussioni teoriche non lunghe.

Per facilitare il principiante, i nomi dei controlli e delle classi sono per lo più lasciati con i nomi dati. I pulsanti di comando vengono quindi sempre chiamati Button1 o Button2, i moduli Form1, Form2 ecc.

Scegli un'area di applicazione che fa per te e inizia da questa. Si spera che i primi esempi siano abbastanza semplici.

L'intero libro è sotto la GNU Free Documentation License ed è quindi disponibile gratuitamente e può essere ridistribuito fintanto che i termini della licenza² essere rispettato.

Auguro a tutti lo stesso divertimento di apprendimento e possibilmente di miglioramento che avevo durante la scrittura!

¹ <http://www.madeeasy.de/7/prgmini.htm>

² http://de.wikibooks.org/wiki/Condizioni_di_licenza

Rho³ nell'ottobre 2004 (UTC)

Infine, una citazione. Non è inteso molto sul serio: più stupido è il programmatore, più densi sono i programmi. Autore sconosciuto.

w: Gambas (linguaggio di programmazione)⁴ °

0.1 Che cos'è Gambas?

Gambas è un linguaggio di programmazione per lo sviluppo di software.

Indica un ambiente di programmazione visuale open source (IDE) per Basic sotto Linux. Inoltre rende disponibili i vantaggi di Visual Basic sotto Linux KDE e Gnome. Esiste una versione per Windows che, a differenza di Visual Basic, è concessa in licenza gratuitamente.

Gambas è sviluppato da Benoît Minisini, che vive vicino a Parigi e si assicura che Gambas stia diventando migliore e più sofisticato.

La versione attuale è Gambas 2.0. Le versioni stabili sono state fornite con Linux Professional Distribution 9.x da Suse e installate automaticamente. È disponibile anche un pacchetto per Debian (Sarge).

Con i gamberi puoi facilmente

- sviluppare un programma con un'interfaccia grafica utente GUI
- accedere a database come MySQL o PostgreSQL
- Sviluppa applicazioni KDE con DCOP
- traduci il tuo programma in molte lingue.
- Rendi disponibili i programmi Visual Basic su Linux.
- Crea soluzioni di rete.

Gambas è in una fase di sviluppo stabile. La superficie IDE stessa è scritta in Gambas-BASIC.

Secondo questa mappa del mondo⁵ per chiudere, sarà ulteriormente sviluppato e utilizzato in tutto il mondo.

0.2 Perché Gambas?

Molti programmatori Visual Basic amano programmare in Gambas.

Se ti sei lasciato alle spalle il mondo Microsoft e sei passato con successo a Linux, stai cercando disperatamente un linguaggio di programmazione facile da usare come Visual Basic. Hai fatto una buona scelta con i gamberi, perché i gamberi sono molto stabili e confortevoli. Inoltre, Gambas ha risolto alcune incongruenze di VB e ti allena costantemente alla programmazione orientata agli oggetti. Quindi impari ancora.

3 <http://de.wikibooks.org/wiki/Benutzer%3ARho>

4 <http://de.wikipedia.org/wiki/Gambas%20%28Programmiersprache%29>

5 <http://www3.clustrmaps.com/counter/maps.php?user=d7b3ed83>

0.2.1 Citazioni

"Questo progetto mira a creare un ambiente di sviluppo grafico basato su un interprete Basic, in modo da avere un linguaggio come Visual Basic sotto Linux."

"Tuttavia, questo software basato su Qt è uno degli strumenti RAD open source più avanzati disponibili.

È quasi completo e molto utilizzabile e stabile. L'obiettivo di Benoit è evitare gli errori di sviluppo di Microsoft Visual Basic. Il concetto moderno di Gambas è disponibile come grafico sul sito web. Gambas è tradotto in molte lingue. Per eseguire le applicazioni Gambas è necessario che il componente interprete Gambas sia installato sulla macchina dell'utente."

0.3 Come puoi imparare i gamberi?

Per chiunque abbia esperienza di programmazione con Basic o Visual Basic, passare a Gambas è piuttosto semplice.

Per la pura programmazione si possono consultare libri su Visual Basic. Molti dei programmi possono essere avviati anche con Gambas.

Un buon libro per principianti è: VISUAL BASIC 6 FOR DUMMIES. Sembra un po 'strano, ma è abbastanza utile. Presto ci sarà sicuramente un libro GAMBAS FOR DUMMIES.

Oppure puoi lavorare su questo libro wiki e contribuire a migliorarlo.

Anche alcuni corsi di programmazione Visual Basic sono molto utili. Vedere i collegamenti alla fine di questo libro.

0.4 Scarica Gambas

Gambas può essere scaricato come file tar zippato e compilato da solo, ma i pacchetti già pronti sono più convenienti da usare.

Su gambas.sourceforge.net⁶ puoi trovare tutto ciò di cui hai bisogno per questo.

0.5 Installazione di Gambas

Il modo più semplice per ottenere Gambas è installarlo dalla propria distribuzione.

Quindi dalla 9.1. una versione utilizzabile di Gambas già inclusa. Devi solo essere finito come root

Ottieni Yast2 e il software nel sistema attuale.

Se non hai una versione di Gambas finita nella tua distribuzione Linux o se vuoi aggiornare la tua versione, puoi anche generare Gambas dal codice sorgente. È più facile di quanto pensi.

L'installazione di Gambas è su questa pagina⁷ esattamente descritto.

⁶ <http://gambas.sourceforge.net/en/main.html>

⁷ <http://gambas.sourceforge.net/compilation.html>

Di seguito ho tradotto la compilation per la versione 1.0 in tedesco:.

0.6 Descrizione dettagliata

Prima di compilare, dovresti controllare se i componenti importanti sono disponibili sul tuo sistema:

- I pacchetti per sviluppatori X11.
- Il pacchetto per sviluppatori Qt 3.
- Il pacchetto per sviluppatori KDE 3 (se vuoi usare anche i componenti KDE).
- I pacchetti per sviluppatori PostgreSQL, MySQL o SQLite (se si desidera compilare driver di database).
- I pacchetti di sviluppo libcurl (versione 7.10.7 o successiva), se si desidera utilizzare il componente network-curl.
- I pacchetti per sviluppatori SDL e SDL_mixer (se si desidera utilizzare i componenti SDL).
- I pacchetti per sviluppatori libxml e libxslt (se si desidera utilizzare i componenti XML).

Quindi dovresti andare su Internet.

È necessario che siano installati i seguenti pacchetti di sviluppo:

- X11,
- QT3,
- KDE3,
- PostgreSQL,
- MySQL.

0.6.1 Scarica

Il modo in cui lo fai dipende dalla distribuzione.

Scarica il codice sorgente di Gambas. Per fare ciò, puoi fare clic sul file di download di Gambas in Konqueror e copiarlo nella tua directory home, oppure puoi avviare il comando wget dalla console nella tua directory home.

```
[ linux @ home ] $ wget  
http://gambas.sourceforge.net/gambas-1.0.tar.bz2
```

Il pacchetto ha una dimensione di circa 4 MB. A seconda del tipo di connessione Internet, il caricamento sul computer potrebbe richiedere del tempo.

0.6.2 Decomprimere

È meglio decomprimere il pacchetto a casa:

Qui si passa alla directory home:

```
[ linux @ home ] cd ~
```

Disimballare:

```
[ linux @ home ] $ bunzip2 gumbas-1.0.tar.bz2
```

Copia Gumbas1.0 in una nuova directory:

```
[ linux @ home ] $ tar xf gumbas-1.0.tar
```

```
[ linux @ home ] $ ls  
... gumbas-1.0 ...
```

Ora vai in questa nuova directory

```
[ linux @ home ] $ cd gumbas-1.0
```

Ora compila gumbas:

0.6.3 Verifica della configurazione

Configure è usato per esaminare il tuo sistema al fine di adattare il codice sorgente al tuo sistema. È in esecuzione un programma script, che si trova nella directory Gumbas. Quindi inserisci ./ davanti al comando di compilazione:

```
[ linux @ home gumbas-1.0] $ ./configure  
...
```

Molti messaggi vengono ora riprodotti sulla riga di comando. Se qualcosa non funziona, vengono visualizzati messaggi di errore. È quindi necessario passare alla risoluzione dei problemi.

È possibile escludere i componenti gumbas durante la compilazione scegliendo le seguenti opzioni:

--disable-kde-component per disattivare il componente KDE --disable-db-component per disattivare il componente database

E così via: digita ./configure --help per un elenco completo delle opzioni.

Esempio:

```
[ linux @ home gumbas-1.0] $ ./configure --disable-db-component
```

Inoltre, tieni presente che i componenti vengono automaticamente disattivati se i pacchetti per sviluppatori corrispondenti non possono essere trovati sul tuo computer; viene quindi visualizzato un avviso nel pannello Configura output.

Se la configurazione avviene senza errori, è possibile avviare la compilazione con make:

0.6.4 Compilazione con make

```
[ linux @ home gambas-1.0] $ make  
...
```

La compilazione richiede circa 20 minuti su un vecchio Pentium-II 400 Mhz e più di 5 minuti su un nuovo Athlon XP 2000+. Quindi bisogna essere pazienti. Puoi usare questo tempo per rovistare un po 'nella documentazione.

Se si verifica un errore durante la compilazione, ovviamente verrà nuovamente visualizzato un messaggio di errore. Poiché un tale bug è insolito, dovrebbe essere segnalato nella mailing list di Gambas.

0.6.5 Installazione come root make install

Se la compilazione è terminata senza problemi, puoi installare Gambas completamente. Per farlo, devi avere i privilegi di root. Quindi usa il comando Linux su con la tua password di root in seguito.

```
[ linux @ home gambas-1.0] $ su -c "make install"  
Parola d'ordine:  
..."
```

Inserisci la tua password di root e l'installazione inizierà. Di solito è installato in / opt / gambas. Se vuoi usare una directory di installazione diversa, devi usare l'opzione di configurazione --prefix. Fare riferimento al file INSTALL nella directory del file sorgente per maggiori dettagli.

Per ottenere gamberi z. B. per installare in / usr, devi inserire quanto segue:

```
[ linux @ home gambas-1.0] $ ./configure --prefix = / usr
```

Notare che Gambas crea collegamenti simbolici nella directory / usr / bin in modo che tu abbia sempre i programmi Gambas nel tuo percorso e che possa essere facilmente trovato dall'ambiente di sviluppo.

Se installi Gambas direttamente in / usr, questi collegamenti simbolici non vengono creati.

Quando tutto è pronto, puoi avviare la GUI di Gambas:

```
[ linux @ home gambas-1.0] $ gambas
```

Per esperienza, il modo più semplice per imparare un linguaggio di programmazione è seguire semplici esempi.

Scegli un'area che ti interessa, che si tratti di grafica, modifica del testo o applicazioni di database. Se inizi a programmare in quest'area, inizierai con successo, perché hai già alcune conoscenze precedenti.

Per tutti i matematici, i piccoli programmi per computer sono il posto giusto da cui iniziare. Per un artista, dovrebbe essere un piccolo programma di pittura. Gli appassionati di Internet hanno bisogno del proprio browser che possono programmare da soli. Ci sono buoni esempi e possibilità per tutto questo in Gambas.

0.7 Avvio di Gambas

Dopo l'installazione, avvia Gambas con Alt e F2 dalla finestra di comando diretto. Basta inserire gambas.

Viene quindi visualizzata la seguente schermata di benvenuto:



Fig. 1 La schermata di benvenuto

Puoi anche avviare Gambas usando il pulsante del menu di KDE in basso a sinistra. Quindi vai a Sviluppo e ambiente di sviluppo. Se ti piacciono i gamberi B. installato come file rpm di Suse, Gambas appare lì come una voce.

Puoi anche mettere i gamberi sul desktop. Fare clic con il pulsante destro del mouse nell'area libera del desktop. Quindi scegli Crea nuovo, File e collega a un programma. Ci sono alcune cose che devi aggiungere alle proprietà.

Ecco come appare un progetto Gambas:



Fig. 2 Un progetto tra Gambas

Se stai cercando un'icona Gambas, puoi trovarla qui:

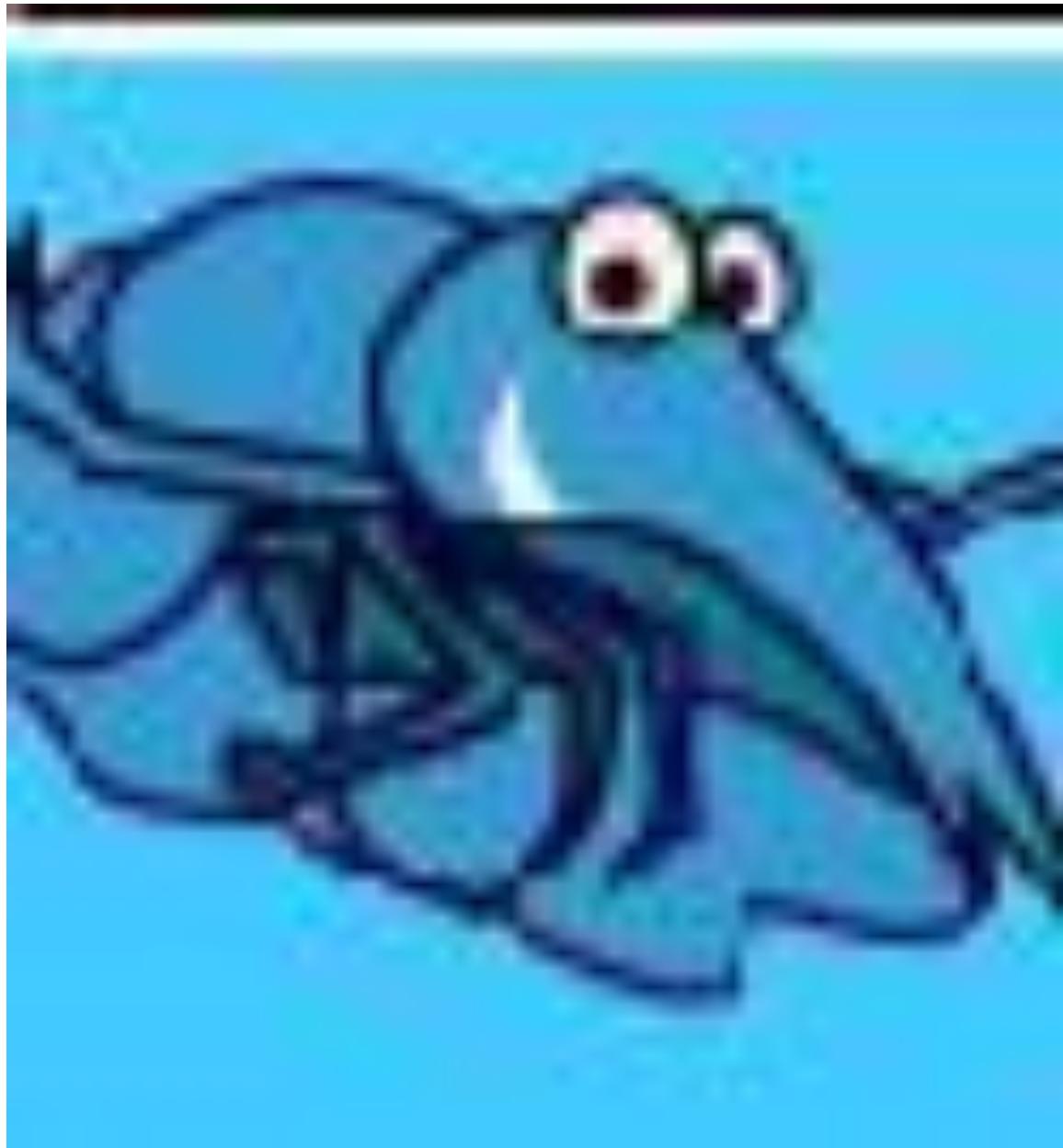


Fig. 3 L'icona Gambas

Fino ad ora non è presente alcuna icona Gambas nella raccolta di icone KDE di Linux.

0.8 Ciao mondo!

Come ogni introduzione a un linguaggio di programmazione, anche questa non arriva senza Hello World! su.

0.8.1 Per prima cosa creiamo un nuovo progetto:

- Avvia Gambas
- Seleziona Nuovo progetto nella finestra iniziale che appare

- Viene visualizzata la finestra della procedura guidata, fare clic su Avanti
- Seleziona la voce Crea progetto grafico e clicca nuovamente su Avanti
- Assegna un nome al progetto Hello_World - puoi lasciare vuoto il campo del titolo - fai di nuovo clic su Avanti
- Ora seleziona una directory in cui salvare il tuo progetto. (Puoi, tuttavia, lasciare l'impostazione predefinita poiché ogni progetto arriva in una nuova sottodirectory.)
- Fare clic su Avanti
- Fare clic su OK per uscire dalla procedura guidata e iniziare lo sviluppo

0.8.2 Ora crea un nuovo modulo:

- Si apriranno tre finestre
 - La finestra del progetto
 - La finestra degli strumenti
 - La finestra delle proprietà
- Nella finestra Progetti, fai clic con il pulsante destro del mouse su Moduli e seleziona Nuovo, quindi Modulo
 - Lascia il nome del modulo in Form1 come suggerito e fai clic su Avanti.
- Ora si apriranno più finestre.
 - la finestra del modulo su cui viene creata la superficie
 - la finestra del codice in cui è inserito il codice del programma

0.8.3 Ora è programmato:

- Immettere il seguente programma nella finestra del codice:

```
PUBLIC SUB Form_Open ()  
    STAMPA "Hello world!"  
END
```

0.8.4 Ci siamo

- Premere il tasto F5 per compilare ed eseguire il programma
- Ora vedrai una finestra in Hello World! stand

A differenza di VB, il comando di stampa non viene emesso nella forma grafica ma nella finestra diretta.



Fig. 4

Se non vedi la finestra immediata, devi spostare la finestra del modulo con il mouse o passare ad essa con ALT e Tab.

Arresta il programma facendo clic nell'angolo in alto a destra della finestra del modulo.

Prova le seguenti varianti che funzionano senza una finestra diretta:

```
PUBLIC SUB Form_Open ()  
    ME.Caption = "Hello world!"  
    FINE
```



Fig. 5

O come messaggio:

```
PUBLIC SUB Form_Open ()
    Message.Info ("Hello world!", "OK")
FINE
```



Fig. 6

Esci da Gambas con File Esci o Ctrl + Q. Il tuo programma verrà salvato automaticamente. Vedi anche: madeeasy.de^{8 °}

0.9 Secondo programma a Gambas

Fai tutto come hai fatto con il primo programma. Ma dai al progetto un nome diverso, ad es. Breve e dolce 2.

Crea un nuovo modulo come descritto nella sezione Primo programma e chiamalo come preferisci. Puoi anche lasciare il Form1 predefinito così com'è.

Ora premi F6.

Viene visualizzata la casella degli strumenti.

Selezionare un pulsante di comando (pulsante OK) e una casella di testo (campo abc) dalla casella degli strumenti. Fare doppio clic sui simboli per copiarli nel modulo.

Ora fai clic sul pulsante di comando e inserisci il seguente codice:

⁸ <http://www.madeeasy.de/7/prghallowelt.htm>

```
Pulsante PUBLIC SUB1_Click ()  
    Textbox1.Text = "Funziona"  
    FINE
```

Le righe superiore e inferiore sono già presenti. Quindi devi solo copiare la riga centrale

```
    Textbox1.Text = "Funziona"
```

Quindi premere F5.

Il programma dovrebbe iniziare.

Fare clic sul pulsante di comando.

Il testo viene visualizzato nella casella di testo.



Fig. 7

Arrestare il programma con la casella di controllo X in alto a destra del modulo.

Con F6 puoi ancora modificare le proprietà dei tuoi tre oggetti. Per fare ciò, prova qualcosa con proprietà diverse.

0.10 Terzo programma: programmi nella finestra del terminale senza grafica

È inoltre possibile utilizzare Gambas per programmi senza interfaccia utente grafica. Questi programmi vengono eseguiti nella finestra diretta (= finestra del terminale) sulla console. Con Visual Basic non c'è più questa possibilità.

Avvia Gambas come al solito.

Scegli Nuovo progetto.

Segui la procedura guidata e premi Avanti.

In Seleziona tipo di progetto, seleziona Crea progetto terminale.

Continua a seguire la procedura guidata e premi Avanti.

Dai un nome al tuo progetto, ad es. 1mal1.

Segui la procedura guidata e premi Avanti.

Seleziona la directory in cui salvare il tuo progetto come sottodirectory.

Segui la procedura guidata e premi Avanti.

I dati del tuo progetto vengono riepilogati.

Premere OK

Viene ora visualizzata la finestra del progetto. Fare clic con il pulsante destro del mouse sulla directory Classes.

Quindi premere il pulsante sinistro del mouse su Nuovo e Classe.

Lascia il nome a Class1 come classe iniziale e premi OK.

Immettere il seguente programma come esempio:

```
STATICO PUBBLICO SUB Principale ()  
DIM x AS intero  
DIM y AS intero  
For x = da 1 a 10  
    FOR y = da 1 a 10  
        PRINT x; "volte"; y; "="; X*y  
    NEXT  
NEXT  
END
```

```
9 mal 8 = 72  
9 mal 9 = 81  
9 mal 10 = 90  
10 mal 1 = 10  
10 mal 2 = 20  
10 mal 3 = 30  
10 mal 4 = 40  
10 mal 5 = 50  
10 mal 6 = 60  
10 mal 7 = 70  
10 mal 8 = 80  
10 mal 9 = 90  
10 mal 10 = 100
```

Fig. 8 Output nella finestra immediata

Premi F5 per avviarlo.

Dai un'occhiata al risultato nella finestra diretta. Possibilmente passare avanti e indietro tra le finestre con Alt - Tab.

Esci dal programma cliccando sul quadrato rosso nella finestra del progetto. O con ALT - F4

Esci da Gambas con File Esci o con Ctrl + Q o continua a programmare.

Domanda: come è possibile rendere disponibile un programma non grafico come file eseguibile sul desktop?

0.11 Come posso copiare il codice del programma da questo corso?

Istruzioni per scaricare i programmi da Internet. Ecco come va:

- Segna il codice su queste pagine con il mouse.
- Copiarlo negli appunti con Ctrl + C.
- Avvia il tuo ambiente di programmazione Gambas.
- Inizia un nuovo progetto.
- Inizia con una nuova forma.
- Ottieni i controlli ei campi di comando necessari dalla barra degli strumenti con F6.
- Assegnare un nome ai controlli come vengono chiamati nel codice. Con F4 sulle proprietà.
- Fare clic sul campo del comando principale in modalità progettazione.
- Copia il codice nella subroutine del campo di comando con Ctrl + V.
- Avvia il programma con F5.

0.12 Crea file eseguibile

Sotto la voce di menu PROGETTO, seleziona Crea file eseguibile. Questo è possibile solo se il programma funziona correttamente. Provalo in anticipo con F5.

Il programma viene quindi salvato nella directory in cui si trova il testo di origine.

Con Gambas2 versione 1.9.49 viene creato un file eseguibile con estensione .gambas. Questo file può essere avviato con il terminale e il comando /home/xxx/gambas2/programm.gambas, ad esempio. xxx è il nome della directory home. Si presume che il programma da avviare sia memorizzato nella directory / home / xxx / gambas2 e che il nome del file eseguibile sia programm.gambas.

Con la vecchia versione 1.0.15 di Gambas (installata con Ubuntu 7.04) non è così facile avviare il programma. Di seguito una soluzione che funziona con Ubuntu e Gnome e probabilmente anche con il desktop KDE:

In Ubuntu 6.10 e 7.04 e con Gnome ricevo sempre il seguente messaggio di errore all'avvio di un programma Gambas:

```
X Errore: BadDevice, dispositivo di input 169 non valido o non inizializzato
Codice operativo principale: 147
Codice operativo minore: 3
ID risorsa: 0x0
Impossibile aprire il dispositivo
X Errore: BadDevice, dispositivo di input 169 non valido o non inizializzato
Codice operativo principale: 147
Codice operativo minore: 3
ID risorsa: 0x0
Impossibile aprire il dispositivo
```

I programmi (propri e anche programmi di esempio) funzionano perfettamente nell'ambiente di sviluppo Gambas, nonostante un messaggio di errore. Anche i comandi Gambas "compila tutto" e "crea file eseguibile" funzionano senza un messaggio di errore, ma non sono mai riuscito ad avviare il file eseguibile. C'erano vari messaggi di errore all'avvio del programma. Anche con il desktop KDE (ho installato il mega pacchetto Kubuntu desktop), il successo non è arrivato.

Dopo molti esperimenti e per caso, ho finalmente raggiunto il mio obiettivo: ho chiamato il file eseguibile "something.exe", dove .exe è il fattore decisivo. Con un doppio clic sul file in Nautilus, il programma è stato avviato al di fuori dell'ambiente di sviluppo Gambas. L'intera cosa mi lascia perplesso, inizialmente ho pensato che per iniziare potrebbe essere necessario Wine. Non è così perché stavo sperimentando una seconda macchina su cui non era installato Wine.

Con l'editor esadecimale ghex ho confrontato i file eseguibili "qualcosa" e qualcosa.exe "creati da Gambas. I file sono in realtà leggermente diversi, "qualcosa" è di pochi byte più corto di "qualcosa.exe"

0.13 Posiziona il programma sul desktop

Hai creato un piccolo programma con Gambas e lo hai salvato come eseguibile. Ora vuoi aviarlo direttamente dal desktop KDE.

- Per fare ciò, fai clic con il pulsante destro del mouse su uno spazio vuoto dell'interfaccia di KDE.
- Seleziona quindi la voce di menu Crea nuovo, File e Collega al programma ...
- Nel seguente menu delle proprietà, seleziona la voce Generale e assegna un nome al collegamento.
- Quindi selezionare la voce Programmi e nella riga di comando premere il tasto Seleziona.
- Trova la directory corretta in cui si trova il tuo file eseguibile, ad esempio home / name / gam-bas99 / program1
- Selezionare il programma corretto da questa directory e confermare le immissioni con OK.

Il tuo programma apparirà ora sul desktop con una ruota dentata blu come simbolo (= icona).

Guarda anche: <http://www.madeeasy.de/7/prgkde.htm>

Per rendere un programma non grafico accessibile come file eseguibile sul desktop sotto KDE, dopo aver selezionato il programma, fare clic sul pulsante Opzioni avanzate ... in basso a destra e selezionare Avvia in Terminale lì.

0.14 Assegnare al programma una propria icona (simbolo del programma)

Le icone sono file bitmap con un massimo di 64 * 64 pixel. Una versione più recente del programma KlconEdit è adatta per creare un'icona. Dopo aver disegnato l'icona del programma, è meglio salvarlo nella stessa directory del file eseguibile. Quindi devi ancora collegare il file desktop con la nuova icona. Per fare ciò, fai clic con il tasto destro del mouse sulla ruota dentata blu del tuo programma e seleziona Proprietà del punto più basso. Fai clic sull'ingranaggio blu e cerca la tua icona in Altri simboli e Seleziona.

Guarda anche: <http://www.madeeasy.de/7/icon.htm>

0.15 avvio del programma

Un programma finito può essere avviato in diversi modi:

1. Dalla GUI dell'ambiente di programmazione con F5
2. Dall'ambiente di programmazione GUI tramite la barra dei menu
3. Dal desktop con un click sul programma, una volta compilato e salvato sul desktop.
4. Con Alt + F2 e inserisci il nome del programma dopo averlo compilato.

Opzioni di avviamento speciali:

- Facendo clic su un pulsante di comando.
 - Questa versione è la più facile da capire per i principianti.
- Con Form_Open ()
 - Avvio automatico del programma quando viene caricato un modulo
- Quando fai doppio clic sulla forma (evento = doppio clic)
- Quando si tocca la forma con il mouse (Evento = Invio)

0.15.1 Caratteristiche speciali

Avvia un programma ed esci immediatamente

Resta da vedere se questo abbia senso. L'esempio seguente mostra come funziona. Hai solo bisogno di una forma vuota per farlo funzionare.

```
PUBLIC SUB
Form_Open() DIM a AS
String[]
DIM x AS
Integera = NEW
String[]
FOR x = 0 TO 4
    a.Add(Str(x))
NEXT
a[0] = "Anna"
a[1] = "Anton"
a[2] = "Toni"
```

```
a[3] = "Max"
a[4] = "Klaus"FOR
x = 0 TO 4PRINT
  a[x]
NEXT
ME.CloseEND
```

Con

```
PUBLIC SUB Form_Open ()
```

il programma viene avviato direttamente.

Con

```
ME.Chiudi
```

nella penultima riga termina di nuovo immediatamente.

Domanda: come è possibile ridurre la finestra di output diretto a livello di codice?

Il programma inizia con il trasferimento dell'argomento

Su cui lavorare

impedire l'avvio del secondo programma

Su cui lavorare

0.16 fine programma

- Con il pulsante di chiusura, di solito una croce nell'angolo in alto a destra della finestra
- Termina con ME.Close non con END come in VB!

```
PUBLIC SUB Button1_Click()
  ME.Close
END
```

0.16.1 Fine del programma con ME.Close

In VB potresti terminare un programma con il comando END. Non funziona a Gambas.

Il comando Quit produce anche messaggi di errore. Utilizzare quindi il comando ME.Close.

Il programma di esempio ha un solo pulsante di comando.

Inizia un nuovo progetto, ottieni una nuova forma.

Ottieni un pulsante di comando dalla casella degli strumenti = casella degli strumenti.

Se si preme il pulsante, il programma dovrebbe essere interrotto immediatamente.

Ecco come appare:



Fig. 9

Il codice seguente è dietro Button1 e dietro il form: The following code is behind Button1 and behind the form:

```
PUBLIC SUB Button1_Click()  
    ME.Close  
END
```

0.16.2 Terminare con il tasto ESC

```
PUBLIC SUB Form_KeyPress()  
    IF Key.Code = Key["Esc"] THEN ME.CloseEND  
END
```

o semplicemente impostare la proprietà Annulla su True nelle proprietà del pulsante di uscita. Quando si preme il tasto ESC, ciò ha lo stesso effetto di fare clic sul pulsante con il mouse

0.16.3 problemi

- Ci sono problemi con il comando STOP.
- Con il comando Esci, ci sono spesso problemi.
- Con Window.Close (non è statico).
- Le finestre possono essere chiuse con la combinazione di tasti ALT + F4.
- Termina e chiudi il database prima

0.17 Programma di stallo

- Termina il programma facendo clic sul punto rosso nella GUI
- Esci dal programma facendo clic sulla X nell'angolo in alto a destra dello schermo
- Termina con ME.Chiudi
- Uccidi il programma con Alt + F4
- Programma di stallo: chiama con Ctrl + ESC, quindi kpm.
- Inizia l'input diretto con Alt + F2, quindi avvia kpm (kpm + Return)
 - Ottieni un elenco di tutti i processi in esecuzione
 - Segna il processo difettoso e premi il tasto kill in basso.
- O ancora più semplice: apri una finestra della console, inserisci killall gambas.
- Diverse possibilità:
 - Apri una finestra della console, ps -d | Immettere grep gambas e annotare il numero del processo.
 - Segue il comando kill -9 PROCESS NUMBER. Con questo l'intero gambero viene abbattuto.
 - Se vuoi solo terminare un certo processo (dialogo, ecc.), Cercalo con: ps -ef | grep gambas scoprirà il processo corretto e lo ucciderà come descritto sopra. Nessun processo sopravvive a queste varianti di tiro. Non è quindi necessario riavviare il sistema!
- L'arresto del programma con Ctrl + Pausa o Ctrl + C non funziona in Gambas.
- Il programma di stallo con Ctrl + Alt + Canc non funziona in Gambas.

Cosa puoi fare se rimani bloccato a programmare con i gamberetti?

0.18 Funzione di aiuto fornita

Con ogni installazione di Gambas viene installato un ampio sistema di aiuto. Questo browser della guida può essere utilizzato bene se non sai cosa fare con i problemi di programmazione. Questo browser della guida viene richiamato con F1:

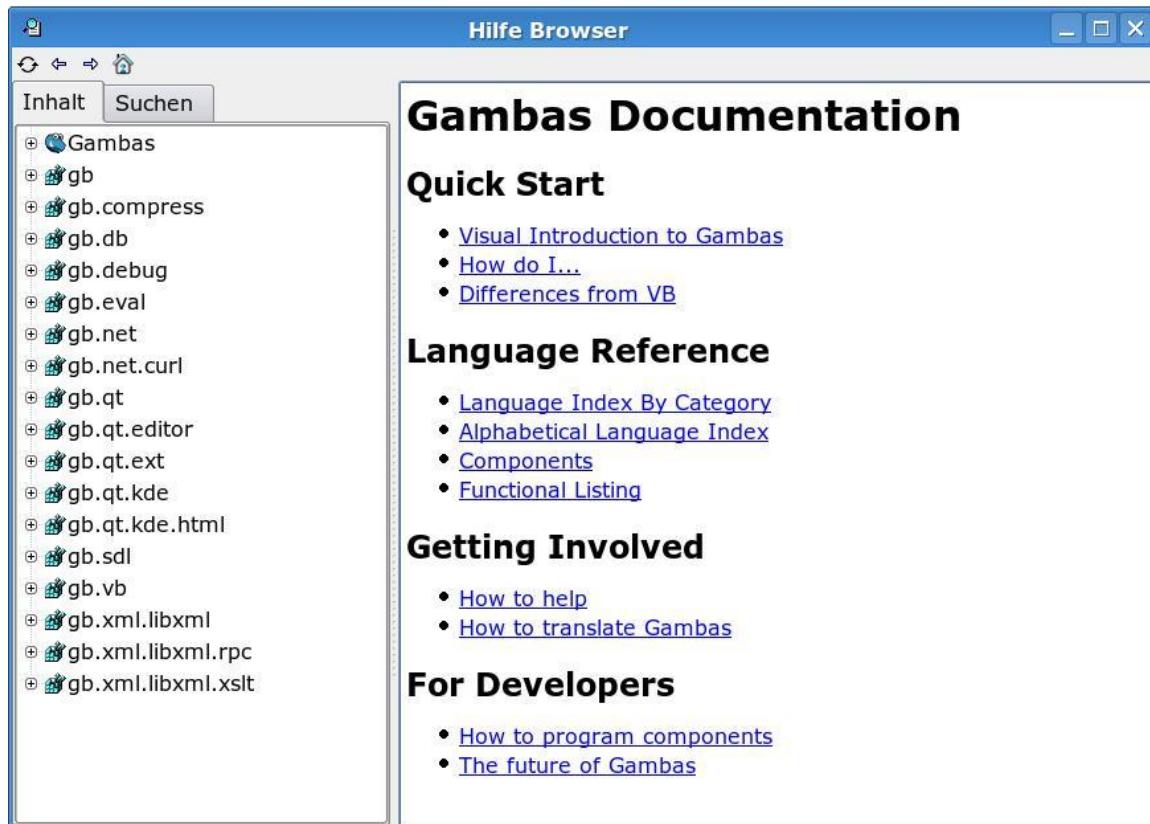


Fig. 10 Il browser della guida di Gambas

L'indice alfabetico dei comandi Gambas è molto utile. C'è spesso un piccolo programma di esempio per un comando.

Anche il richiamo dei componenti nel browser della guida è molto utile.

Ad esempio, se fai clic sul componente `gb.qt`, otterrai una panoramica di tutte le classi di questo importantissimo componente grafico di Gambas.

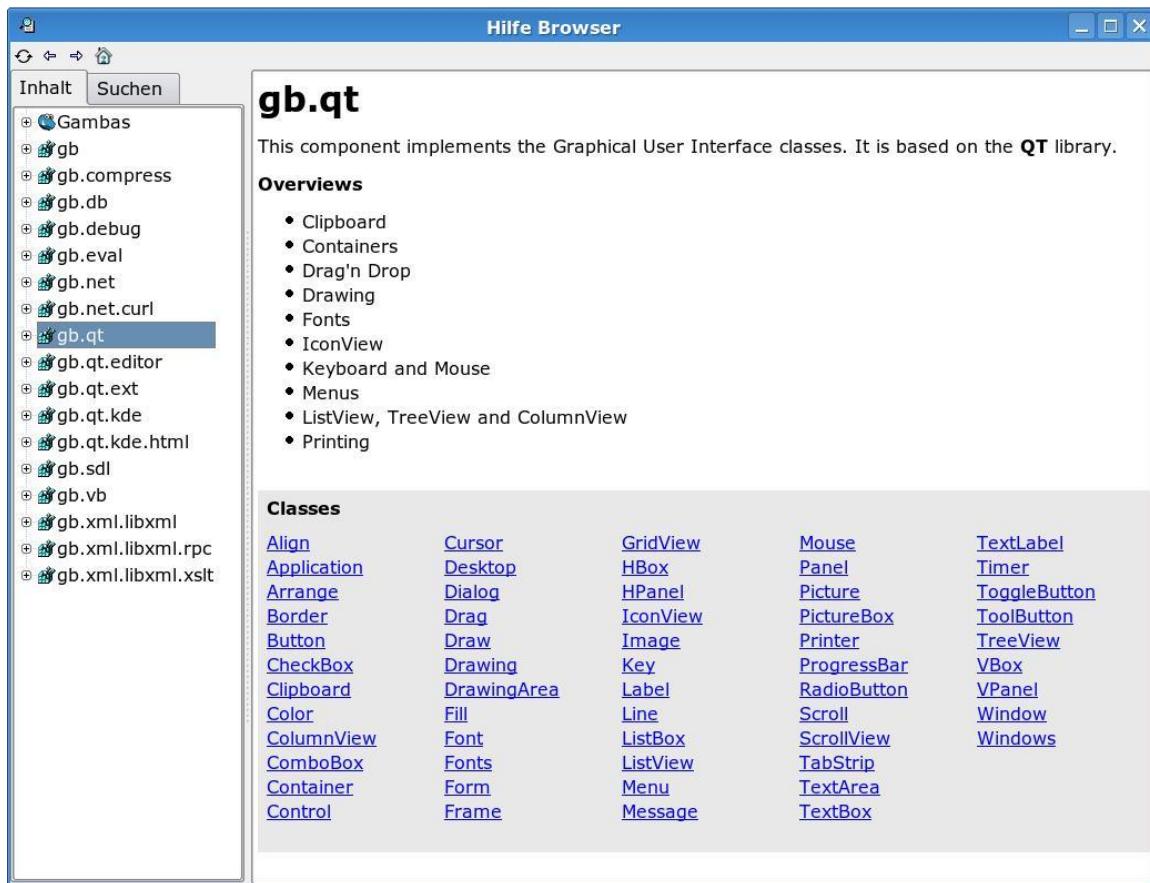


Fig. 11 Il componente gb.qt nel browser della guida

Ora puoi ottenere ulteriore aiuto facendo clic sui corsi offerti lì.

Ad esempio, se hai fatto clic sulla classe Appunti, viene visualizzato un testo di aiuto sull'argomento.

Questo testo elenca quindi tutte le proprietà, tutti i metodi (eventi) e tutte le costanti di questa classe.

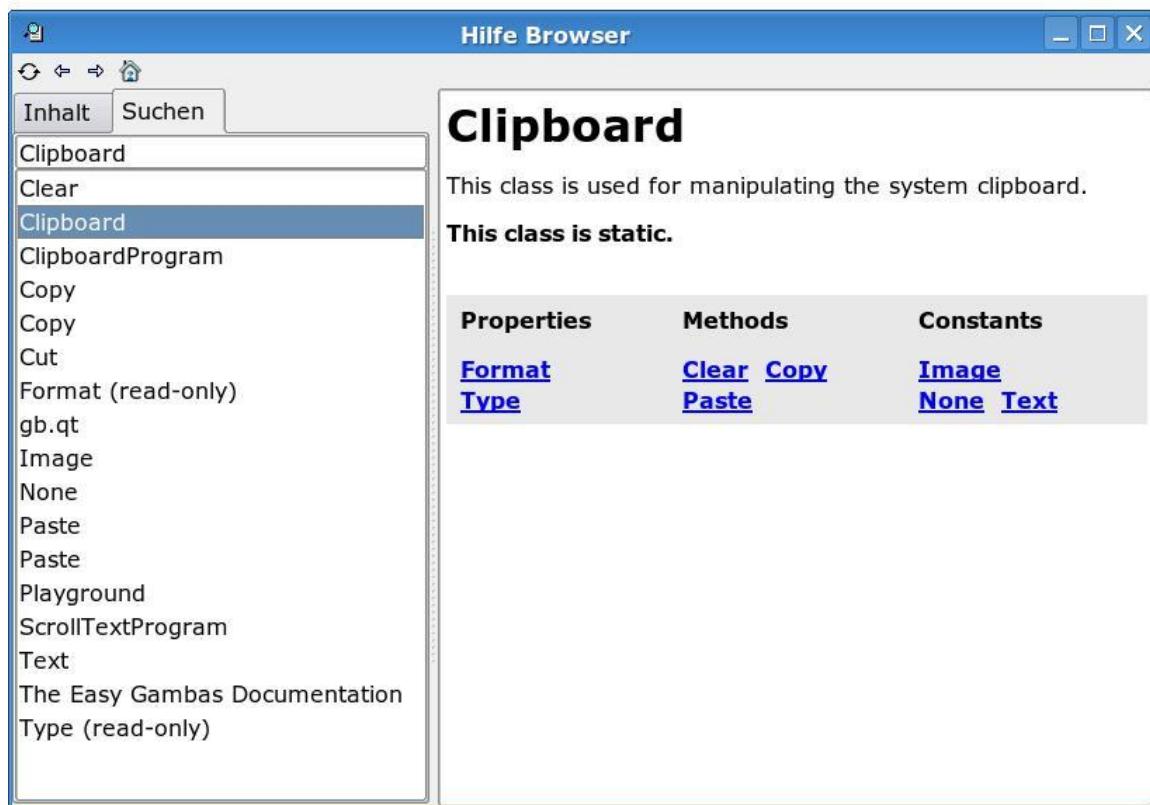


Fig. 12 La classe Appunti nel browser della Guida

Uno svantaggio del browser della guida di Gambas è che tutti i testi sono scritti in inglese.

0.19 Aiuto su Internet

0.19.1 gambasdoc.org

Il browser della guida viene mantenuto e migliorato dagli utenti Gambas stessi. Puoi scaricarlo da [gambasdoc.org](http://www.gambasdoc.org)⁹. Richiamare su Internet e visualizzare tutti gli argomenti.

0.19.2 Forum di discussione

A questo indirizzo web¹⁰ c'è un forum di discussione in lingua inglese molto ampio sul tema dei gambas. Anche Benoit Minisini è regolarmente rappresentato in questo forum.

Qui¹¹ è attualmente in fase di creazione un forum di lingua tedesca.

⁹ <http://www.gambasdoc.org/help/>

¹⁰ <http://sourceforge.net/mailarchive/forum.php?forum=gambas-user>

¹¹ <http://www.gambas-club.de/>

0.19.3 Domande frequenti su Gambas Wikibook

Puoi anche ottenere aiuto sull'argomento dei gamberi da questo libro. Descrivi il tuo problema qui: Gambas: Domande¹²

0.19.4 Indice Gambas

Un'opzione di ricerca per parola chiave per Gambas Wikibook può essere trovata qui: Google Wikibook Gambas¹³

0.20 Domande aperte

Se hai difficoltà con i gamberi, scrivi semplicemente il tuo problema alla fine di questo testo. Altri possono imparare dalle tue difficoltà e puoi sviluppare lezioni ed esempi per chiarire i problemi. Il modo migliore per imparare è attraverso i problemi che sono stati superati e nelle discussioni con utenti esperti che possono essere guardati alle spalle durante la programmazione.

Discussioni sui gamberi possono essere trovate anche qui su Internet:
<http://sourceforge.net/mailarchive/forum.php?forum=gambas-user>

Alcune delle domande che vengono poste saranno trattate nella parte successiva e tradotte in tedesco.

C'è un modo per controllare un sito web con Gambas? Ad esempio, accedi con il tuo nome utente e password, quindi aggiungi alcuni link a un campo di testo, inserisci un numero nativo e quindi fai clic sul link invia ...

0.21 cursore alla fine di un testo?

Come posso posizionare il cursore alla fine del testo in una TextArea?
 (TextArea1.Selection = Len(TextArea1.Text) non funziona)

Risposta Vedi Gambas: _Textarea #

How_can_man_jump_with_the_Cursor_to_the_End_des_Text_in_the_Textarea_.3F^{14th}

```
PUBLIC SUB Button1_Click() textareal.pos
= Len(textareal.text)'Der Cursor geht
ans Ende des Textes
END
```

¹² Capitolo 0.19.4 a pagina 25

¹³ <http://www.google.de/custom?sa=Google+Search&domains=de.wikibooks.org/wiki/>

¹⁴ Gambas & sitesearch = de.wikibooks.org / wiki / Gambas 14
 Capitolo 0.75 a pagina 78

0.22 Come posso scrivere più variabili in un file e leggerle di nuovo quando necessario (es. Avvio del programma)?

Ecco una soluzione completa e funzionante. Ciò significa che 5 parti di testo salvate separate da virgole vengono richiamate in seguito e assegnate ai campi:

0.22.1 Salvataggio in un file

"ZusFeld" è una TextArea nascosta per accogliere l'array. È riempito con la stringa "ZUS". Il contenuto del primo campo è la variabile per il nome file "NAME". Il file "NAME" viene quindi salvato nella directory home dell'utente, che contiene i valori dei 5 campi separati da virgole.

```
Dim ZUS AS String
DIM NAME AS
String

ZusFeld.Clear
ZUS = Name.Text & Chr(44) & Feld2.Text & Chr(44) & Feld3.Text
&Chr(44) & Feld4.Text & Chr(44) & Feld5.Text & Chr(10)
ZusFeld.Text = ZUS
NAME = Name.Text

File.Save(system.home & "/" & NAME ,ZusFeld.Text)
```

Puoi anche separare i campi con Chr (10) per leggerli singolarmente con LineInput e assegnarli ad altre variabili.

0.22.2 Caricamento dei valori dal file

In questo caso il ComboBox contiene i nomi di diversi elenchi di valori. "Name.Text = name" invece di "Name.Text = List [0]" funzionerebbe anche a questo punto tramite il trasferimento di variabili dal ComboBox.

```
Nome DIM AS String
Elenco DIM come stringa []
DIM i AS intero

nome = ComboBox2.Text
ZusFeld.Text = File.Load (system.home & "/" & name)
List = Split (AddField.Text, ",")
PER i = da 0 a 4
Name.Text = List [0]
Field2.Text = List [1]
Field3.Text = List [2]
Field4.Text = List [3]
Field5.Text = List [4]
IL PROSSIMO
FINE
```

0.23 Come puoi usare gli appunti per salvare la grafica?

0.24 Come posso interrogare quale scheda è attiva in una tabstrip?

Il modulo del test è costituito da un pulsante, una tabstrip e una casella di testo. Dopo aver fatto clic sul pulsante, il nome della scheda attiva apparirà nella casella di testo.

```
Pulsante PUBLIC SUB1_Click ()  
    Textbox1.Text = Tabstrip1.Text  
FINE
```

0.25 Cosa significa la h in molti esempi di GambasCode?

```
"File di classe Gambas  
PRIVATO $hProcess AS Process
```

Innanzitutto, suggerisco di mettere i commenti prima del codice che spiegano. Penso che sia la tradizione.

- "h" ti dice che la variabile riceverà un handle di oggetto.
- h ti dice che la variabile otterrà un handle di oggetto
- "Processo" ti dice che sarà un processo.
 - Processo significa che questa sarà la maniglia di un processo.
- E il tipo di variabile conferma che sarà un oggetto della classe Process.
 - E il TIPO della variabile conferma che diventa un oggetto di una classe di processo.

0.26 Come posso determinare la dimensione dei caratteri?

```
come posso ottenere i valori pixel effettivi di font.height e font.width (nello stesso valore  
pixel di
```

ME.x e ME.y) in modo da poter impostare una dimensione corretta per il pulsante o le etichette

```
x = font.width ("Questa è una stringa di prova per vedere quanti pixel è presente  
le stringhe sono lunghe ")  
y = font.height ("H")
```

Questo dipende dal dispositivo (cioè è diverso tra lo schermo e la stampante) quindi usi Draw.TextHeight e Draw.TextWidth per ottenere quei valori. Penso che potresti dover avere un Draw.Begin attivo quando chiavi quei metodi.

0.27 Come puoi dividere una riga in un editor e rimetterla insieme?

Da: Maria Graciela Cesar < mgcesar @ ya ... > come tagliare una riga nell'editor? Come puoi dividere una riga in un editor e rimetterla insieme? 21-10-2004 03:50

Ciao a tutti. Uso i gambas da un po 'e riesco ancora a trovare il modo per fare il prossimo: voglio scrivere una frase su più righe. In visual basic puoi usare i connettori + e &. Quindi scrivi una lunga riga in più righe. Non so se puoi farlo in gambas (e se è una domanda stupida) Grazie

Vorrei assemblare una frase da più righe. In VB puoi usare i comandi + e & usa. In questo modo puoi assemblare una lunga fila da diversi corti. Puoi farlo anche a Gambas?

Risposta: In linea di principio sì, ma: il segno più + è usato solo in Gambas per operazioni matematiche. Tutte le aggiunte di stringhe (= testo) vengono eseguite con il segno &.

Esempio:

Stampa "Questa è una frase" & "che è stata completata con una clausola subordinata."

Stampa 3 + 5

Guarda anche <http://www.madeeasy.de/7/prgtxtplus.htm>

0.28 Come si può programmare una funzione in modalità terminale?

Quando si programmano le funzioni in modalità terminale, i messaggi di errore si verificano ripetutamente. Funziona bene in modalità grafica.

Confronti http://de.wikibooks.org/wiki/Gambas:_Erste_Schritte#Terzo_programma:_Programmi_nella_finestra_TERMINALE senza_grafica

0.29 Come puoi scrivere in una casella di testo usando solo lettere maiuscole?

Ciao ! Ho un vero problema a cui sto aggrappato per diverse ore e anche su Internet Non ho trovato niente

Vorrei scrivere in una casella di testo utilizzando solo lettere maiuscole. Quindi, se inserisco "a1b2c3d", ad esempio, "A1B2C3D" dovrebbe apparire nella casella di testo ... Come si fa?

Primo suggerimento <http://www.madeeasy.de/7/prgklein.htm>

In Terminalmdous un codice di programma per Ucase e Lcase ha questo aspetto:

```
STATICO PUBBLICO SUB Principale ()  
DIM una stringa AS  
a = "test"  
STAMPA a  
STAMPA LCase (a)  
STAMPA ASase (a)  
FINE
```

Risultato nella finestra diretta:

```
test  
test  
TEST
```

Potrebbe essere un po 'più difficile separare i numeri dalle lettere. Ma Ucase e Lcase si riferiscono ovviamente a Gambas solo a lettere e non a numeri, come mostra il seguente esempio:

"File di classe Gambas

```
STATICO PUBBLICO SUB Principale ()  
DIM una stringa AS  
a = "Test1234"  
STAMPA a  
STAMPA LCase (a)  
STAMPA ASase (a)  
FINE
```

Risultato nella finestra diretta:

```
Test1234  
test1234  
TEST1234
```

È un po 'più complicato se vuoi che le maiuscole appaiano durante la digitazione. Puoi renderlo facile se prima consenti l'inserimento con lettere minuscole e poi imposti l'intero testo, ad esempio in una casella di testo con Ucase, e lo elabori ulteriormente.

È necessario un pulsante di comando e una casella di testo per avviare il programma. L'esempio viene eseguito in modalità grafica.

```
Pulsante PUBLIC SUB1_Click ()  
DIM gros AS String  
gros = TextBox1.Text  
' Stringa da cambiare.  
Texbox1.Text = UCASE $(grande)  
' Cambia in maiuscolo.  
FINE
```

Forse è ancora più semplice:

```
Pulsante PUBLIC SUB1_Click ()  
Textbox1.Text = UCASE $(Textbox1.Text)  
FINE
```

Ed ecco, funziona.

La soluzione più semplice è utilizzare l'evento CHANGE:

```
PUBLIC SUB textbox_Change ()  
textbox.text = UCASE $(textbox.text)  
FINE
```

L'evento di modifica viene chiamato ogni volta che viene modificata la casella di testo e il testo viene immediatamente scritto in lettere maiuscole. Ciò significa che pulsanti aggiuntivi o simili non sono necessari, poiché il testo viene convertito immediatamente.

0.30 Riduce la finestra di output diretto

Come puoi posizionare la finestra di output diretto e la finestra del modulo in programmi che eseguono automaticamente dall'interno del codice in modo che non si sovrappongano?

Un semplice trucco è spostare la finestra del modulo da in alto a sinistra a destra in modalità Progettazione. Oppure dai

```
PUBLIC SUB Form_Open ()  
ME.X = 500  
ME.Y = 1  
FINE
```

Nel tuo codice. Inoltre non so come rimpicciolire la finestra diretta.

0.31 Crea nuova directory, copia directory

Come puoi prendere Gambas

- creare una directory
- copiare una directory

Copia .. in ... purtroppo funziona solo con i file. Con Dir (...,"*.*") Ottieni solo i file.

Poiché anche la GUI di Gambas è stata creata con Gambas e viene creata una nuova directory per ogni progetto Gambas, in linea di principio è possibile farlo. Ma come?

0.31.1 Crea una nuova directory

È possibile creare una nuova directory con il comando MakeDir. Attenzione: funziona solo

- se hai inserito un percorso corretto,
- se hai il permesso di scrittura per questo percorso
- se vuoi scrivere una directory nella finestra aperta di Gambasprojet. (vedi Application.Name)
- nessun file con lo stesso nome esiste in questo percorso.

Esempio:

È necessario un pulsante di comando per avviare il programma:

```
Pulsante PUBLIC SUB1_Click ()
  MKDIR "/ home / tuonome / test"
  FINE
```

Sostituisci il percorso tuonome con il nome della tua directory.

Controlla con Konqueror se la directory è stata realmente creata.

È preferibile inserire una routine di errore nel caso in cui il percorso esista già o non sia disponibile l'accesso in scrittura.

```
Pulsante PUBLIC SUB1_Click ()
  PROVA MKDIR "/ home / tuonome / test"
  SE ERRORE ALLORA
    message.box ("Crap, non ha funzionato!")
  FINISCI SE
  FINE
```

0.31.2 Copia directory

Il comando COPY copia un file da una directory a un'altra.

COPIA la directory / file di origine NELLA directory / file di destinazione

Il comando precedente copia un file dalla directory di origine alla directory di destinazione. Il file non deve necessariamente avere lo stesso nome del file di destinazione. Tuttavia, le sottodirectory non vengono copiate.

(Notare che non è possibile copiare le directory in modo ricorsivo con questa funzione.)

Esempio:

```
' Esegue il backup del file di configurazione gambas nascosto
' La directory deve esistere
' e bisogna avervi accesso
COPIA "/home/tuonome/.gambas/gambas.conf" A
"/mnt/save/gambas.conf.save"
```

Nell'esempio, sostituisci il tuo nome con il nome della tua home directory.

L'esempio produce un messaggio di errore se:

- La directory non esiste
- Non c'è accesso in scrittura alla directory
- Il file di destinazione esiste già.

Esempio2:

È necessario un pulsante di comando nel modulo per avviare l'esempio.

```
Pulsante PUBLIC SUB1_Click ()  
    COPIA "/home/tuonome/.gambas/gambas.conf" A  
    "/home/tuonome/test/gambas.conf.save"  
FINE
```

Se vuoi copiare più file contemporaneamente, il comando Dir ti aiuterà. Vedi Gambas:
File¹⁵.

Di seguito è mostrato come è possibile copiare o spostare intere directory con Gambas2:

Abbiamo bisogno di un modulo con due display DirView per la selezione della directory di origine e di destinazione. Abbiamo anche bisogno di 4 pulsanti di comando.

Con il primo pulsante copiamo tutti i file della directory selezionata. Il secondo pulsante copia solo i file più recenti e il terzo pulsante sposta una directory in un'altra posizione. Il quarto pulsante viene utilizzato solo per uscire dal programma.

Poiché Gambas non sovrascrive alcun file esistente con il comando copy, qualsiasi directory di destinazione esistente viene prima eliminata per la copia di tutti i file. I file vengono quindi copiati.

Per motivi di tempo, ha senso copiare solo file più recenti con directory di grandi dimensioni. In questo caso, i file nella directory di destinazione non vengono eliminati. Se Gambas rileva che un file esiste già, i dati dell'ultimo salvataggio vengono confrontati tra loro. Se il file da copiare è più recente di quello esistente, il file esistente viene eliminato e sostituito da quello più recente.

Spostare una directory equivale a copiare tutti i file. Viene eliminata anche la directory di origine.

Ecco il codice completo del modulo:

```
PUBLIC SUB Form_Open ()  
    DirView1.Root = "/" 'viene visualizzata la directory iniziale  
    DirView2.Root = "/"  
FINE
```

```
PUBLIC SUB cmdEnd_Click ()  
    ME. Chiudi  
FINE
```

15 Capitolo 6.55 a pagina 265

```
PUBLIC SUB cmdCopyAll_Click ()
  'se disponibile: elimina la directory di destinazione
  BasisModule.Folder_delete (DirView2.Current e DirView1.Current)
  ' Copia la directory di origine nella directory di destinazione
  BasisModule.Folder_kopieren (DirView1.Current, DirView2.Current)
FINE
```

```
PUBLIC SUB cmdCopyNew_Click ()
  ' Copia la directory di origine nella directory di destinazione
  ' vengono copiati solo i file più recenti
  BasisModule.Folder_kopieren (DirView1.Current, DirView2.Current)
FINE
```

```
PUBLIC SUB cmdMove_Click ()
  'Elimina la directory di destinazione se disponibile
  BasisModule.Folder_delete (DirView2.Current e DirView1.Current)
  ' Copia la directory di origine nella directory di destinazione
  BasisModule.Folder_kopieren (DirView1.Current, DirView2.Current)
  ' Elimina la directory di origine
  BasisModule.Folder_delete (DirView1.Current)
FINE
```

Inoltre, è richiesto un modulo con il nome BasisModule, in cui sono memorizzate tutte le routine per le attività di cui sopra. (Le routine possono anche essere contenute nel modulo, penso sia più chiaro con il modulo) Genera il modulo e copia il codice seguente.

```
'File del modulo Gambas
PUBLIC SUB Folder_kopieren (folderFrom AS String, folderTo AS String)

File DIM AS String
DIM fileTemp AS String
Info DIM AS stat
DIM info1 AS stat
File DIM da AS String
File DIM To AS String
Cartella DIMTemp AS String []
DIM j AS Numero intero

  PER OGNI file IN RDir (folderFrom, "*")
    fileFrom = folderFrom & "/" & File
    fileTo = folderTo & fileFrom
    PROVA A COPIARE file Da A file A
    SE ERRORE Codice = 38 ALLORA 'File esistente' Controlla se il
      file di origine o di destinazione è più recente info = Stat
      (fileFrom)

      info1 = Stat (fileTo)
      IF info.LastModified> info1.LastModified THEN
        ' Il file di origine è più recente
        ' Il file di destinazione verrà
        eliminato KILL fileTo
        ' Copia il file sorgente
        COPIA file Da A file A
      FINISCI SE
    FINISCI SE
    SE ERRORE Codice = 45 ALLORA "Le directory di destinazione non esistono
      folderTemp = Split (fileTo, " / ")

    FOR j = 0 TO folderTemp.Count - 2
      fileTemp = fileTemp & "/" & folderTemp [j]
      TRY MKDIR Destra $ (fileTemp, Len (fileTemp) - 1)
      PROVA A COPIARE file Da A file A
```

Contenuti

```
IL PROSSIMO  
FINISCI SE  
fileTemp = ""  
IL PROSSIMO  
FINE
```

```
PUBLIC SUB File_delete (cartella AS String)  
File DIM AS String File  
DIM_ AS String 'elimina tutti  
i file  
    PER OGNI file IN RDir (cartella, "") sarà      '.* solo file con estensione  
cancellato  
        file_ = cartella & "/" & Il file verrà      '.* anche file senza estensione  
eliminato  
        PROVA A UCCIDERE file_  
IL PROSSIMO  
FINE
```

```
PUBLIC SUB Folder_delete (cartella AS String)  
File DIM AS String  
File DIM_ AS String  
Cartella DIMTemp AS String []  
DIM i AS intero  
    ' prima elimina tutti i file File_delete  
    (cartella)  
    ' elimina tutte le sottodirectory PER OGNI  
    file IN RDir (cartella)  
        file_ = cartella & "/" & File folderTemp =  
        Dividi (file_, "/") PROVA RMDIR file_  
  
    SE codice.errore = 0 E file_ = cartella ALLORA  
        RITORNA  
  
    FINISCI SE  
    IF error.code = 0 ALLORA "La directory è stata eliminata  
        io = 1  
        FARE  
            'prova a eliminare le directory principali file_ = Left (file_, Len (file_) -  
            Len (folderTemp [folderTemp.Count - i]) - 1)  
            PROVA RMDIR file_  
            SE codice.errore = 0 E file_ = cartella ALLORA  
                RITORNO  
            FINISCI SE  
            io = io + 1  
        LOOP UNTIL error.code = 42 OR error.code = 38 'Directory non vuota  
  
    FINISCI SE  
IL PROSSIMO  
FINE
```

0.32 Come si può interrogare la directory home dell'utente corrente?

Sfortunatamente, il comando Application.home non funziona per questo. Il comando Application.Name fornisce anche il percorso del programma attualmente in esecuzione, ma non la home directory.

La directory home dell'utente corrente può essere ottenuta con il comando system.home

Esempio 1

```
PUBLIC SUB Form_Open ()  
Sistema PRINT Home  
FINE
```

Esempio2:

```
PUBLIC SUB Form_Open ()  
ME.Text = "Informazioni di sistema"  
textarea1.text = "Charset =" & system.Charset & Chr (13) & Chr (10) textarea1.text =  
textarea1.text & "Domain =" & system.Domain & Chr (13) & Chr (10)  
  
textarea1.text = textarea1.text & "Home directory =" & system.Home & Chr  
(13) & Chr (10)  
textarea1.text = textarea1.text & "Host =" & system.Host & Chr (13)  
& Chr (10)  
textarea1.text = textarea1.text & "Language =" & system.Language & Chr (13) & Chr (10)  
  
textarea1.text = textarea1.text & "path =" & system.path & Chr (13)  
& Chr (10)  
textarea1.text = textarea1.text & "User =" & system.User  
' Sistema PRINT  
' Sistema PRINT Dominio  
' Sistema PRINT Home  
' Sistema di stampa Host  
' Sistema PRINT Lingua  
' Sistema PRINT Path  
' Sistema PRINT Utente  
FINE  
PUBLIC SUB Button1_Click ()  
    ME.Close  
FINE
```

Hai bisogno di un'area di testo e di un pulsante di comando per avviare il programma.

0.33 Come si usa il comando DateDiff?

... Non so come chiamare la funzione DateDiff () Qualcuno potrebbe darmi un breve esempio?

Di quale classe DateDiff è un metodo?

Il comando DateDiff proviene da Visual Basic. Calcola la differenza tra due date inserite. È necessario il componente Gambas gb.vb Vedere Gambas: Componenti¹⁶

0,34 virgolette attorno a un testo

Vorrei produrre un output di testo racchiuso tra virgolette. Esempio:

/Home/user/file_name.asm dovrebbe diventare "home / user / file_name.asm". Come funziona?

16 Capitolo 12.18 a pagina 424

Soluzione:

```
button1.text = chr (34) & "home / user / file_name.asm" & chr (34)
```

0,35 Editor di Gambas, funzioni non documentate

Ciao, sto lavorando a un piccolo programma che utilizza GambasEditor. Purtroppo questo è in L'aiuto di Gamba è appena documentato. Chi può spiegarmi le seguenti proprietà e metodi:

- - Congelato
- - evidenziare
- - Altezza della linea
- - Posizioni
- - Showchange
- - Mostra corrente
- - Mostra proc
- - simbolo
- - Sollievo dell'utente
- - Analizza
- - Garantirevisibile

Indice di scheda 0,36 in gambas

Come funziona il tasto Tab in una forma? Puoi cambiare l'ordine degli elementi di controllo saltati? Esiste una cosa come l'indice di tabulazione in VB?

È meglio lavorare con il comando: Setfocus

Esempio: hai diverse caselle di testo e vuoi saltarne una con il tasto Tab.

```
PUBLIC SUB TextBox1_KeyPress ()  
    TextBox2.SetFocus  
    FINE
```

Secondo B.Minisini, il "TabIndex" in Gambas si chiama ZOrder. Questo ZOrder viene modificato tramite l'IDE.

Esempio: contrassegnare la casella di testo durante la creazione della forma. Quindi fare clic con il pulsante destro del mouse nel menu> "Disposizione"> "In primo piano" e selezionare la casella di testo successiva. Il tasto Tab viene utilizzato per saltare nello stesso ordine.

ZOrder non è una proprietà Object (textBox nell'esempio sopra).

Se vuoi cambiare ZOrder in base al codice, devi usare il metodo Raise () o Lower () della classe Control.

Un'altra possibilità per influenzare il passaggio del focus quando si preme il tasto tab: si esce da Gambas e si apre il form da modificare con un editor (nome file: form name.form nella directory del progetto corrispondente). Il codice sorgente XML può essere visto qui. Ora si modifica l'ordine degli oggetti in base all'ordine di esecuzione del tabulatore desiderato spostando un blocco completo di un oggetto con le sue proprietà. Un blocco può essere riconosciuto dalle parentesi. È necessario prestare attenzione a non spostare il blocco al di fuori di un blocco di livello superiore (ad es. Creato da un frame). Gambas elabora il testo di origine in relazione allo ZOrder dall'alto verso il basso. Questo metodo ha il vantaggio rispetto al metodo piuttosto confuso tramite l'IDE Gambas,

Esempio (ZOrder / TabIndex: 1 ° cognome, 2 ° nome):

```
{Cognome TextBox
  Sposta (128,32,448,24)
  Testo = ("")
}
{First name TextBox
  Sposta (128,80,448,24)
  Testo = ("")
}
```

Riordinato (ZOrder / TabIndex: primo nome, secondo cognome):

```
{First name TextBox
  Sposta (128,80,448,24)
  Testo = ("")
}
{Cognome TextBox
  Sposta (128,32,448,24)
  Testo = ("")
}
```

0,37 file zip

Come si accede a un archivio (zip, tar, jar ecc.) O ai file in esso contenuti con Gambas?

Per la risposta, vedere Gambas: compressione¹⁷

0.38 Elaborazione della query con Gambas

Se vuoi leggere l'output di un comando contenente Process_Read (), ottieni solo un prompt (input).

¹⁷ Capitolo 6.97 a pagina 303

Esempio:

```
Comando SHELL WAIT FOR READ WRITE AS myProcess  
PUBLIC SUB Process_Read ()  
    legge la riga in una variabile  
  
        se la variabile è "prompt utente:" allora  
            WRITE getUserInput ()  
        finisci se  
    FINE
```

Questo esempio non funziona, come puoi farlo funzionare?

0.39 Come funzionano i tasti di scelta rapida?

Se vuoi usare i tasti di scelta rapida in un progetto, potresti usare il seguente codice:

Sottofondo pubblico Form_KeyPress ()

```
Se codice.chiave = ... allora  
    Qualcosa accade
```

Sfortunatamente, questo non funziona correttamente. Se il modulo è vuoto, funziona. Se sono presenti altri controlli, non funzionerà più. Come mai???

Fino ad ora non esiste un'interfaccia per le interruzioni globali con l'evento chiave in QT. QT è basato su gambas.

Tuttavia, c'è un trucco per risolvere il problema.

Se hai creato un menu nel modulo, anche le scorciatoie da tastiera del menu funzionano globalmente. Quindi puoi aggiungere un menu nascosto al modulo in cui puoi inserire i tasti di scelta rapida desiderati.

Vedi Gambas: Menu¹⁸

0.40 Converti immagine in immagine e viceversa

In Gambas ci sono 2 modi per modificare le immagini: Immagine e Immagine. Qual è la differenza tra i due? Come puoi convertire un'immagine in un'immagine e viceversa? Il problema sorge, ad esempio, quando si vuole visualizzare il contenuto del buffer (in questo caso un'immagine) in Gambas? Vedi Gambas: Pictures¹⁹

18 Capitolo 6.72 a pagina 278

19 Capitolo 6.2 a pagina 216

0.41 Come è possibile assegnare nuovamente il contenuto di una DrawingArea a un'immagine?

0,42 Euro accedi Gambas

Se leggi un segno di euro in una TextArea e poi prova quanto segue

```
t $ = Conv $(TextArea.Text, Desktop.Charset, "ISO-8859-1")
```

viene visualizzato il messaggio di errore "Conversione stringa errata".

Soluzione: il simbolo dell'euro non esiste nel set di caratteri ISO-8859-1, può essere trovato solo in ISO-8859-15

0.43 È possibile dichiarare in Gambas come in VB con i caratteri \$ etc?

0.44 La riga IF non viene compilata su una riga. Come mai?

Il seguente comando IF x = 3 THEN TRY MKDIR "temp_folder" non viene compilato in una riga. Perché sta succedendo?

0.45 Sostituisci comando

Perché il codice seguente non funziona?

```
Dim a come stringa  
Dim b come stringa  
a = "test" t "  
b = Sostitisci $(a, "" "", "\")
```

b dovrebbe essere "test \" t "

Il carattere \ è un carattere speciale. Provare

```
b = Sostitisci $(a, "" "", "\\")
```

Vedi Gambas: _Textverarbeitung # Der_Replace_Befehl _-_ Sostitisci caratteri^{20th}

20 Capitolo 6.36 a pagina 242

0.46 Sistema di guida programmabile?

Esiste un sistema di guida programmabile in Gambas come .chm in vb?

No, non direttamente. GNU / Linux offre alcune possibilità per questo

- il sistema di aiuto di KDE,
- il sistema di aiuto di GNOME,
- pagine man (Konqueror le mostra molto bene!)
 - Inserisci man: / una volta nella barra degli indirizzi di Konqueror.
 - Guarda anche: http://de.wikibooks.org/wiki/Linux-Kompendium:_Linux_auf_dem/Desktop # Browser web
- pagine html che chiamano direttamente un browser web

0.47 Finestra di dialogo Ricorda percorso

Come puoi far ricordare a Gambas la finestra di dialogo Path una volta che è stata aperta?

Vedi ad esempio Gambas: _Bilder # Laden_Sie_ein_Bild_mit_Dialog_in_eine_Picturebox²¹

0.48 Database: variabili con il nome del campo invece del nome del campo?

Quando apri un database, puoi accedere ai singoli campi con un handle di risultato, soprattutto con il metodo .edit. Sfortunatamente, l'handle dei risultati sembra supportare solo i nomi, non le variabili.

Esempio (non eseguibile, solo per dimostrazione):

Nome DIM AS String

DIM sFeldVorname AS String = "nome"

DIM sFeldNachname AS String = "cognome"

.

.

sql_string = "SELECT * FROM customer WHERE first name LIKE" & name

²¹ Capitolo 6.9 a pagina 219

```
rs = cn.EXEC (sql_string)
```

```
rs! sFeldVorname = "Hugo"
```

```
rs! sFeldlastname = "Müller"
```

I campi "nome" e "cognome" esistono nella tabella.

L'istruzione SQL è solo a scopo dimostrativo, in realtà non funziona nemmeno con il metodo .edit. rs! first name e rs! surname funzionano, ma non se i loro valori devono essere trasferiti nelle variabili 'sFeldVorname' o 'sFeldNachname'.

Allora come si usano le variabili come nomi di campo?

Già respinto come try & error: rs! "Nome campo", rs! ["Nome campo"], rs! [Nome campo], rs! (Nome campo), rs! ("Nome campo")

Una soluzione potrebbe assomigliare a questa:

rs [nome campo] tralascia semplicemente il nominativo e le virgolette.

0,49 gamberi

- Il lato Gambas²² (Inglese)
- Forum di Gambas²³
- Forum di Gambas²⁴
- Forum degli utenti di Gambas^{25th} (Inglese)
- Tutti gli ordini^{26th}
- programmi e codice gambasforge.net in cambio²⁷ (Inglese) [non disponibile]

²² <http://gambas.sourceforge.net/>

²³ <http://www.gambas-club.de>

²⁴ <http://gambas.roemi.de/>

²⁵ <http://sourceforge.net/mailarchive/forum.php?forum=gambas-user>

²⁶ <http://www.gambasdoc.org/help/lang>

²⁷ <http://www.gambasforge.net/>

- Buon articolo di revisione in LinuxUser²⁸
- Articolo di revisione precedente in LinuxUser²⁹
- Alcune cose sul tema dei gamberi in tedesco, miniprogrammi^{30th}
- Un'introduzione con un semplice esempio³¹
- Slashdot ha preso atto di Gambas con un'ottima risposta³²
- Tutorial³³ (Inglese e spagnolo) [non disponibile]
- Vantaggi e svantaggi di Gamba rispetto a VB³⁴ (inglese, pdf, 126kB)
- Alcuni bei programmi³⁵ (Spagnolo)
- Il blog di un attivista³⁶ (Inglese)

0.50 Tutorial di base

- Bel tutorial per vb4 (adatto anche per versioni superiori)³⁷
- Corso VB ben fatto³⁸
- Informazioni dettagliate su VB (FAQ, esempi, suggerimenti)³⁹
- Bella introduzione VB⁴⁰
- Tutorial VB in inglese⁴¹

Se hai un'idea di programma o una richiesta di programma, perché non puoi realizzarla da solo in Gambas perché non hai il tempo o non abbastanza know-how Gambas, allora scrivilo qui.

Forse ci sarà qualcuno che farà accadere l'intera cosa.

Illustrazione: la g piccola e la G grande

28 http://www.linux-user.de/ausgabe/2005/04/066-gambas/index_html?print=y

29 <http://www.linux-user.de/ausgabe/2002/11/050-gambas/>

30 <http://www.madeeasy.de/7/gambas.htm>

31 <http://linuxfocus.org/Deutsch/March2004/article329.shtml>

32 <http://slashdot.org/article.pl?sid=04/10/26/142212>

33 http://www.laclavedefa.org/codigo_abierto/gambas_tutorial/

34 http://geocities.com/arjun_jain/resources/gambas-paper.pdf

35 <http://wiki.gnulinex.org/gambas>

36 <http://www.kudla.org/>

37 <http://www.mozilla.at/user/herf/lv/vb4einf/Inhalt.html>

38 <http://www.rhirte.de/vb/home2.htm>

39 <http://VB-Tec.de/>

40 <http://www.vb-seminar.de/index.htm>

41 http://www.techiwarehouse.com/cms/engine.php?page_id=c6985589



Fig.13

In qualche modo, il piccolo g sentiva di essere osservato tutto il tempo.

0.51 Vorrei creare un indice.

Come posso fare questo?

Che cosa dovrebbe indicizzare l'indice? Un database? Sono necessarie informazioni un po 'più dettagliate per aiutare qui.

-L'indice, come la guida di Gambas, dovrebbe visualizzare un tutorial in una finestra della guida.

Sotto http://de.wikibooks.org/wiki/Gambas:_Html c'è un browser Gambas Html.
Se crei le pagine della guida come pagine HTML, puoi usarle come tutorial della guida.

Taglia 0,52 linee o stringhe dal file

Ho letto un intero file con READ e successivamente cerco una stringa specifica. Quando si verifica, ho letto l'intera riga. Ora voglio eliminare questa riga e aggiungere quanto segue alla riga precedente in modo che il mio file non diventi inutilmente grande. Può essere fatto con il file

gestire il codice corrente? Inoltre non riesco a trovare un comando con il quale posso indirizzare una certa riga per aggiungere una stringa di caratteri ad essa.

Rho: Inizierò analizzando il problema. Per prima cosa ho descritto il comando READ in modo più dettagliato.

Per favore, riferisci http://de.wikibooks.org/wiki/Gambas:_Datei#Daten_einlesen_mit_LEGGERE

Per risolvere ulteriormente il problema, dovrebbe essere descritto in termini più concreti. Sarebbe utile un esempio.

Sono necessari: CommandButton, TextArea e ID.txt file.

Ad esempio, il seguente programma:

```
i AS intero
ii AS intero
hfile AS file
PUBLIC SUB Form_Open ()
APRIRE "ID.txt" PER LEGGERE COME #hFile
LEGGI # hFile, tar1.Text, Lof (hFile)
FINE
PUBLIC SUB cmd1_Click ()
i = Instr (tar1.Text, "A")
SE i> 0 ALLORA
ii = Instr (tar1.Text, "\ n", i) SE ii> 0
ALLORA
tar1.Text = Mid (tar1.Text, i, ii-i)
FINISCI SE
FINISCI SE
FINE
```

"ID.txt" contiene diverse stringhe di caratteri, ciascuna con una CR successiva. Ora voglio ritagliarne uno. Tuttavia, non salvare il testo modificato in una TextArea, ma salvarlo di nuovo in ID.txt. Il messaggio di errore quando si apre il file con CREATE o WRITE viene visualizzato come "Accesso vietato". Tuttavia, i diritti di accesso al file erano impostati su "all". Ci sono comandi che lo semplificano, o piuttosto salta alla riga richiesta, taglia e sposta la riga successiva?

Un altro esempio potrebbe essere:

ListBox, CommandButton

```
txt come stringa
PUBLIC SUB cmd1_Click ()
txt = libx1 [0] .Text
libx1.Remove (0)
FINE
PUBLIC SUB Form_Open ()
libx1.Add ("aaaaa \ n")
libx1.Add ("bbbbbb \ n")
libx1.Add ("aaaaa \ n")
libx1.Add ("bbbbbb \ n")
libx1.Add ("aaaaa \ n")
FINE
```

Questa mi sembra essere l'opzione migliore, ma sorge la domanda su quale metodo richiede il minor tempo di elaborazione o quale rallenta il programma se il ListBox è ancora visualizzato. L'elenco potrebbe contenere anche fino a 100 voci.

0.53 Gambas Wikibook come libro HTML sul tuo computer

Puoi configurare Gambas Wikibook abbastanza facilmente come pagine HTML collegate sul tuo computer.

0.53.1 Configurare la directory / wiki /

Affinché il collegamento funzioni correttamente, è necessario impostare una directory / wiki sul computer. Per fare ciò, apri una console in KDE e crea una directory lì:

```
mkdir / home / USERS / wiki
```

Sostituisci USER con il tuo nome utente. Quindi in quello nuovo Cambia directory:

```
cd / home / USER / wiki
```

0.53.2 Copiare tutti i file da Gambas Wikibook

Quindi copia il seguente file da Wikibook in questa directory:

```
wget -nv -O index.html 'http://de.wikibooks.org/w/index.php?title=Spezial%3APrefixindex&namespace=0&from=
```

Estrai e scarica gli indirizzi delle pagine rimanenti da questo file:

```
wget -nv -c $(cat index.html | tr ""'\n' | egrep '^ / wiki / Gambas' | sort -u | sed's #^# http://de.wikibooks.org#')
```

Infine, regola i link:

```
for i in $(ls); esegui # href = "/ wiki / Gambas" # href = ". / Gambas" # g '$i > $i.temp; mv $i.temp $i; fatto
```

Quindi puoi impostare la panoramica come browser della guida di Gamba sul tuo computer. Per scaricare sempre le ultime pagine da GambasWikibook, ripetere gli ultimi tre passaggi.

0,54 Programma Gambas

La procedura sopra descritta funziona abbastanza bene, ma è scomoda. Infine, questo è un libro di testo sulla programmazione Gambas, quindi cosa potrebbe essere più naturale che scrivere un programma per farlo.

Crea un nuovo progetto grafico con la procedura guidata, chiamalo Wikibook_Download, scegli una posizione di archiviazione e completa la procedura guidata.

Crea una nuova forma, Form1. Posiziona i seguenti controlli sulla forma:

- Pulsante 4
- 2 TLabel
- 2 TextBox
- 1 TextArea
- 1 ProgressBar

Infine apri la finestra del codice e inserisci il seguente codice sorgente: Wikibook Scarica Form1.class⁴²

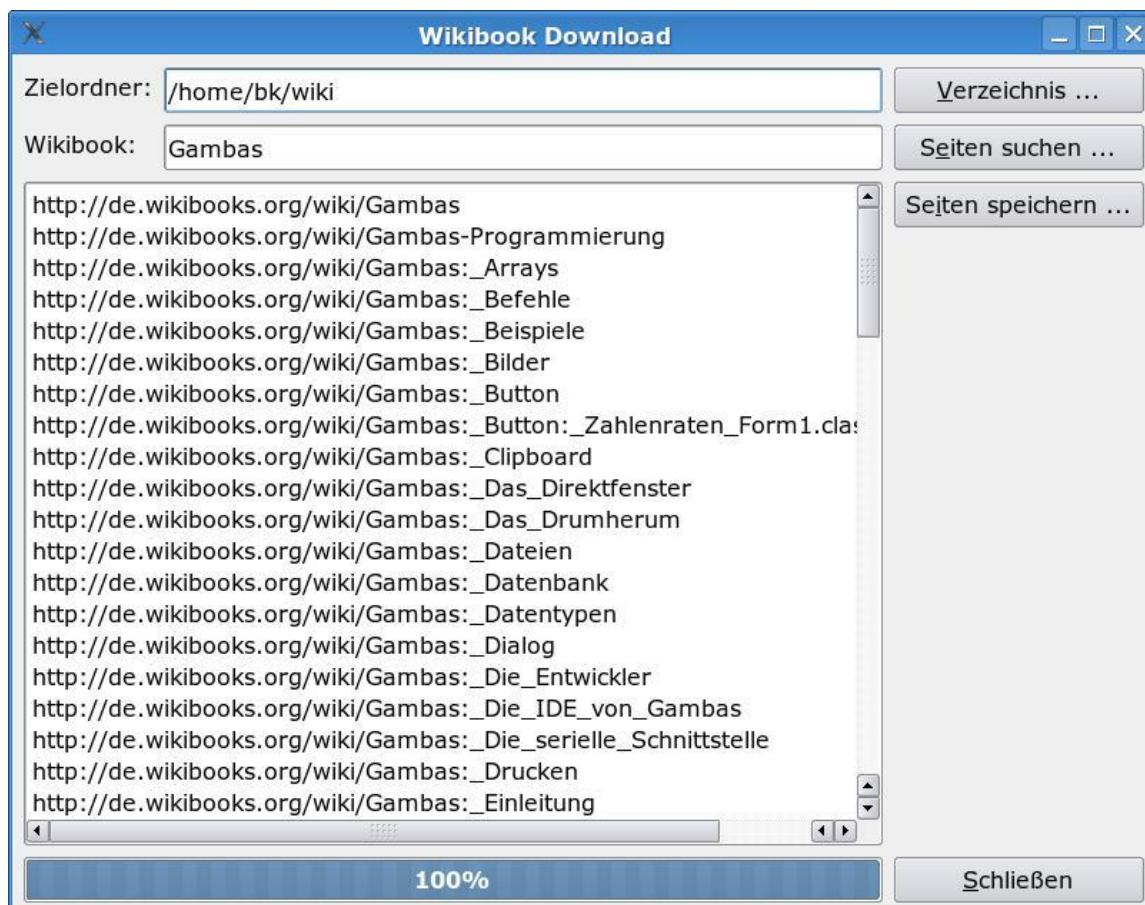


Fig.14 Scarica Wikibook

⁴² <http://de.wikibooks.org/wiki/Gambas%3A%20Lokal%3A%20Wikibook%20Download%20Form1.class>

Quindi puoi avviare il programma.

0.55 chiamata tramite Konqueror

Il tutto viene quindi chiamato tramite Konqueror. In questo modo puoi anche rendere disponibili altri libri wiki localmente sul tuo computer.

Per fare ciò, imposta la pagina / wiki / Gambas sul tuo desktop con il tasto destro del mouse come Crea nuovo, file e link a indirizzo (URL). Immettere / home / USER / wiki / Gambas all'indirizzo.

Forse allora cercherai un bel simbolo nelle proprietà.

Richiamarlo con Explorer non funziona correttamente in Windows.

IDE è un'abbreviazione dall'inglese e sta per: Integrated Development Environment. In tedesco, questo è chiamato ambiente di sviluppo integrato o, in modo un po' più breve, ambiente di lavoro di Gambas.

0,56 finestra in Gambas

0.56.1 Finestra del progetto

La finestra del progetto è la panoramica più importante in Gambas. Dalla finestra del progetto puoi raggiungere tutte le altre finestre e dalla finestra del progetto di solito termini anche la tua sessione di gioco d'azzardo. Pertanto dovresti familiarizzare con le voci di menu⁴³ familiarizzare con la finestra del progetto. All'inizio ti servono solo pochi importanti.

43 Capitolo 0.59 a pagina 56

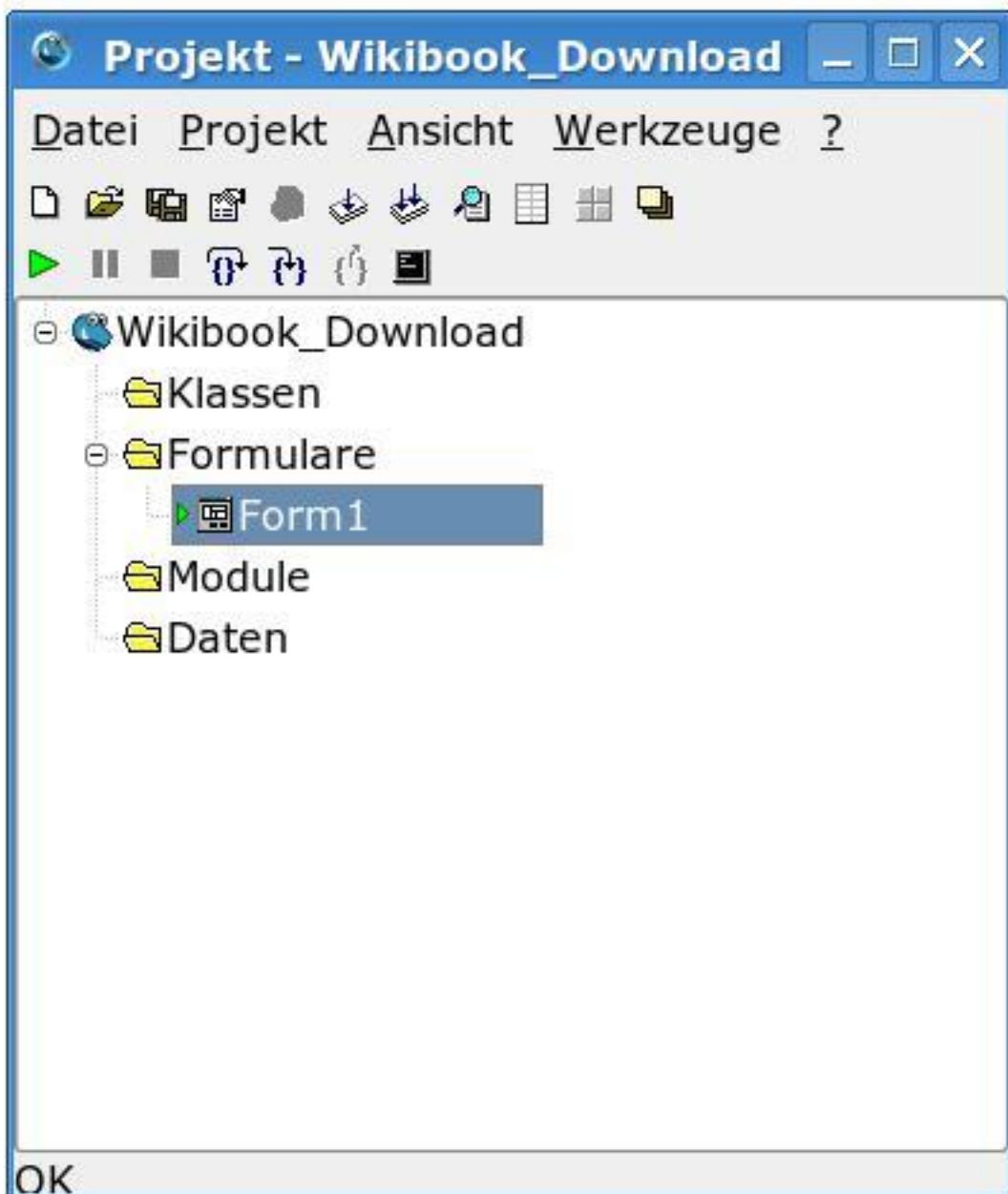


Fig.15 La finestra del progetto

0.56.2 Finestra forma grafica

La finestra della forma grafica mostra la forma corrente ed i suoi controlli in modalità progettazione I controlli possono essere spostati, ingranditi e ridotti, selezionati, copiati e cancellati con il mouse. Facendo doppio clic su un controllo si accede alla finestra del codice associato.

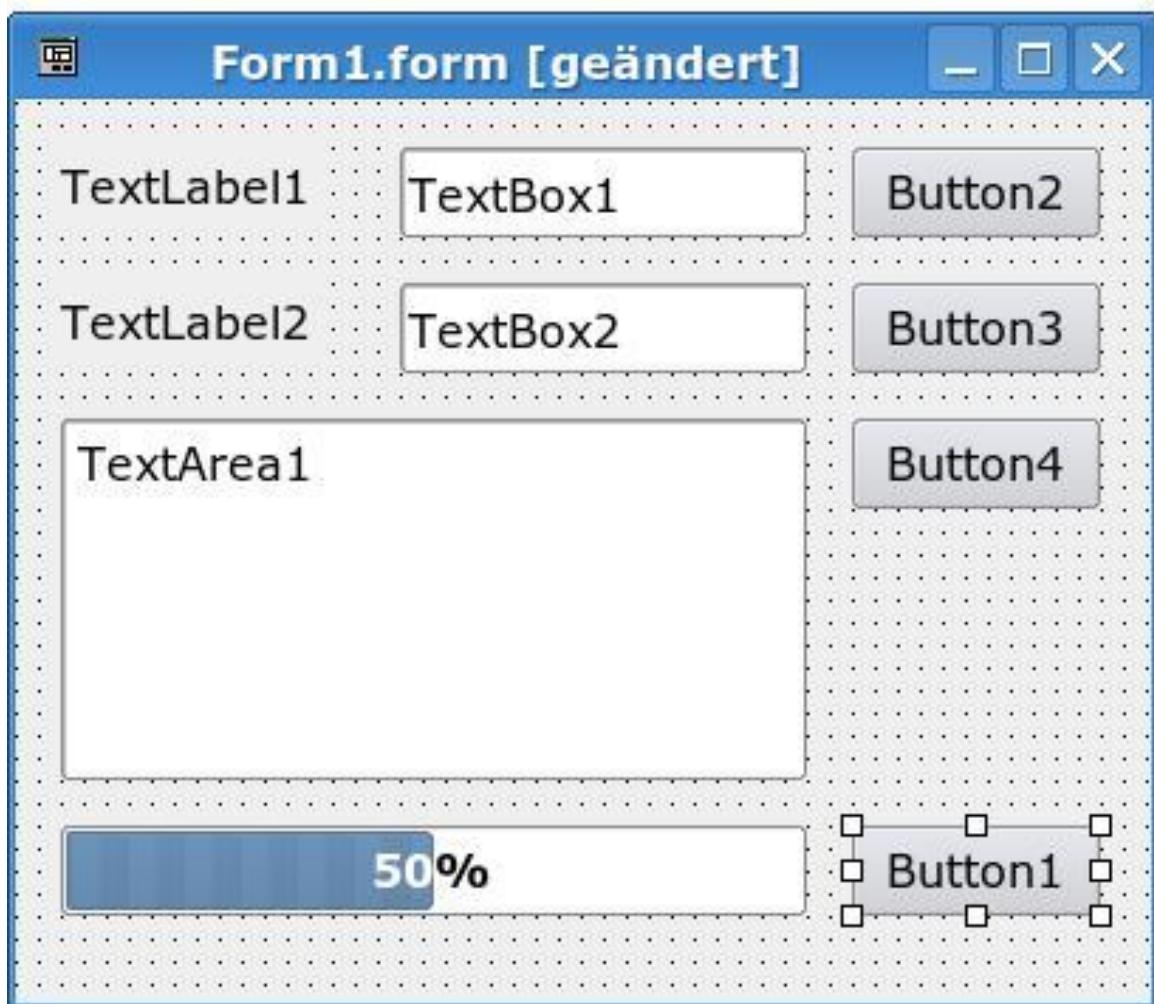
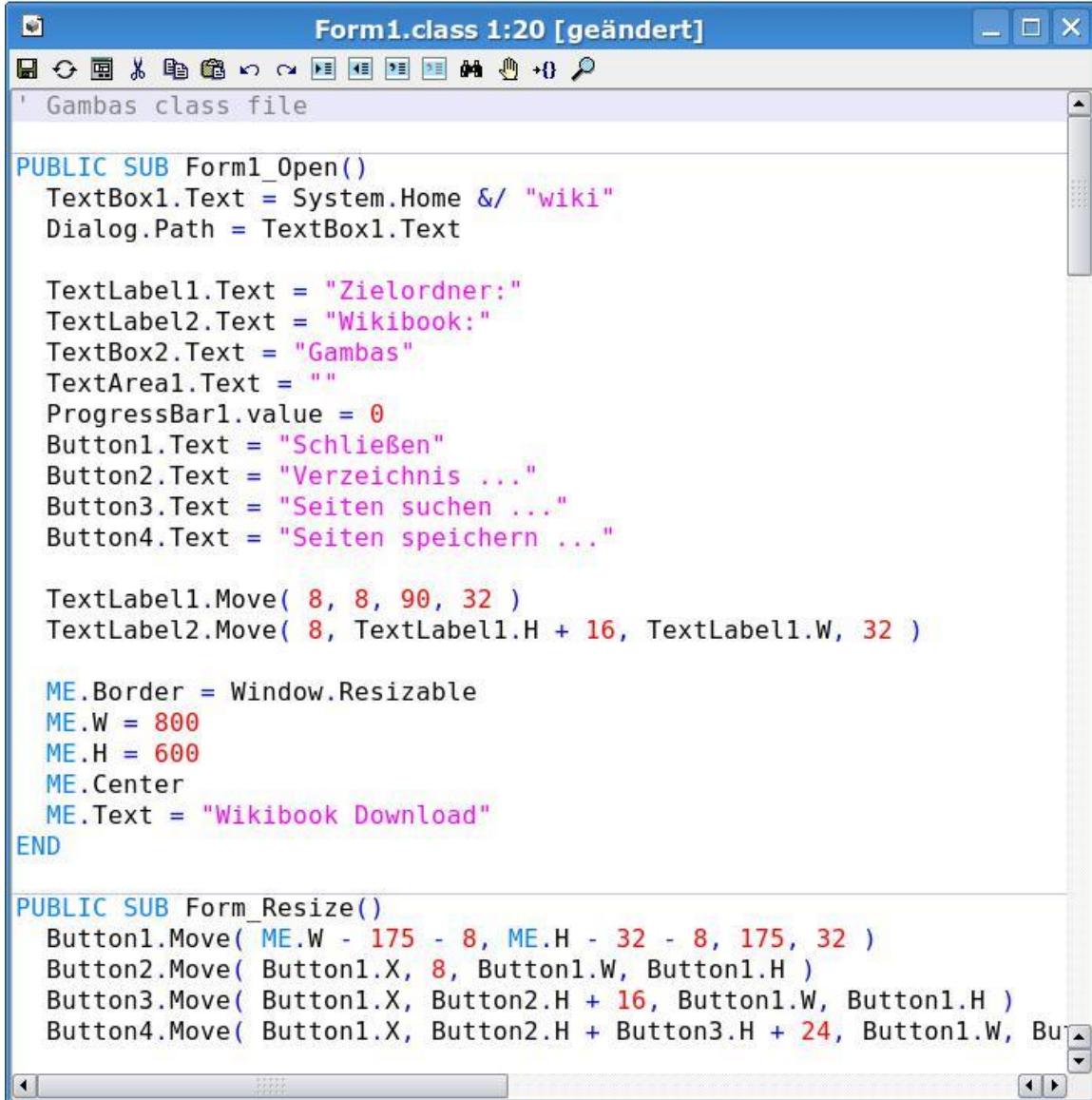


Fig.16 La finestra della forma

0.56.3 Finestra del codice

La programmazione viene effettivamente eseguita nella finestra del codice. La finestra del codice è un pratico editor con una speciale evidenziazione dei colori dei comandi Gambas.



The screenshot shows the Gambas IDE interface with the title bar "Form1.class 1:20 [geändert]". The main window displays the following code:

```
PUBLIC SUB Form1_Open()
    TextBox1.Text = System.Home &/ "wiki"
    Dialog.Path = TextBox1.Text

    TextLabel1.Text = "Zielordner:"
    TextLabel2.Text = "Wikibook:"
    TextBox2.Text = "Gambas"
    TextArea1.Text = ""
    ProgressBar1.value = 0
    Button1.Text = "Schließen"
    Button2.Text = "Verzeichnis ..."
    Button3.Text = "Seiten suchen ..."
    Button4.Text = "Seiten speichern ..."

    TextLabel1.Move( 8, 8, 90, 32 )
    TextLabel2.Move( 8, TextLabel1.H + 16, TextLabel1.W, 32 )

    ME.Border = Window.Resizable
    ME.W = 800
    ME.H = 600
    ME.Center
    ME.Text = "Wikibook Download"
END

PUBLIC SUB Form_Resize()
    Button1.Move( ME.W - 175 - 8, ME.H - 32 - 8, 175, 32 )
    Button2.Move( Button1.X, 8, Button1.W, Button1.H )
    Button3.Move( Button1.X, Button2.H + 16, Button1.W, Button1.H )
    Button4.Move( Button1.X, Button2.H + Button3.H + 24, Button1.W, Bu
```

Fig.17 La finestra del codice

0.56.4 Raccolta di strumenti

La raccolta di strumenti fornisce i controlli grafici per un nuovo progetto. Fare doppio clic su un elemento di controllo per trasferirlo nella finestra del modulo grafico. Il modo più rapido per accedere alla raccolta di strumenti è con il tasto funzione F6.



Fig.18 La collezione di strumenti

0.56.5 Finestra delle proprietà

Ogni forma e controllo ha proprietà diverse. Sono elencati nella finestra delle proprietà e possono essere modificati. In alternativa, puoi modificare o definire le proprietà nel codice. Il modo più rapido per accedere alla finestra delle proprietà è utilizzare il tasto funzione F4.

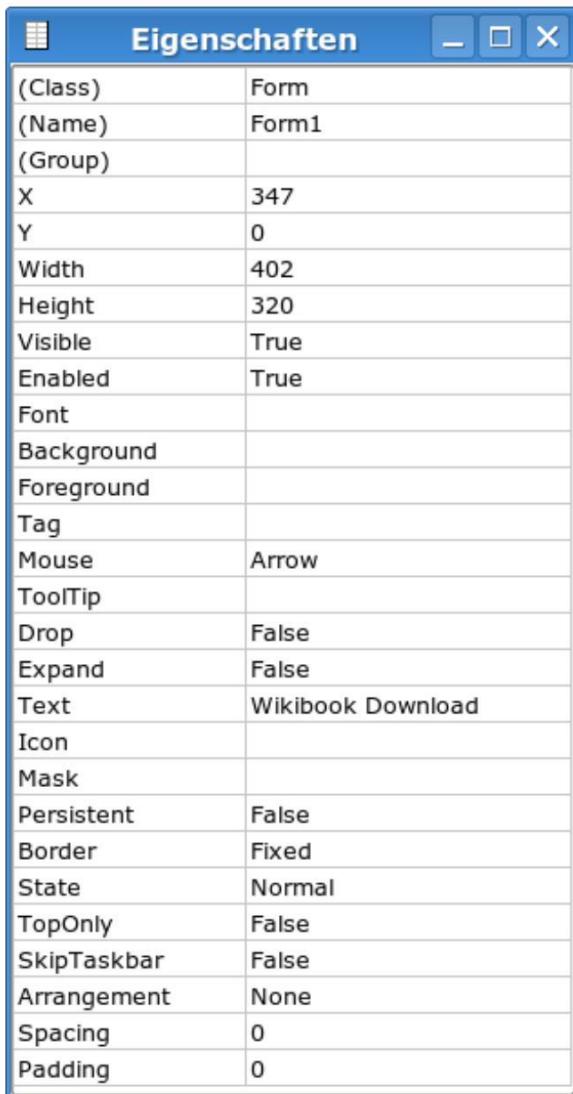


Fig.19 La finestra delle proprietà

0.56.6 Finestra diretta (finestra terminale)

La finestra diretta è un'innovazione di Gambas rispetto a VB. Corrisponde alla finestra del terminale di Linux e viene utilizzato principalmente come output diretto per i comandi PRINT nel codice. È particolarmente utile per la risoluzione dei problemi. Alcuni programmi vengono eseguiti anche completamente nella finestra del terminale. Il modo più rapido per accedere alla finestra diretta è utilizzare il tasto funzione F11.

0.56.7 Finestra di esecuzione

0.56.8 Finestra della Guida

Con F1 si arriva alla finestra della guida. Qui puoi trovare tutti i comandi in elenco alfabetico e funzionale. Spesso c'è anche un piccolo esempio di programmazione in modo che la finestra della guida debba essere utilizzata ampiamente.

0.56.9 e pochi altri

Schermate aggiuntive con tutte le finestre possono essere trovate su gambas.sourceforge.net⁴⁴

0.57 Passare da una finestra all'altra con ALT + TAB

Se hai più finestre aperte contemporaneamente su uno schermo, puoi passare da una finestra all'altra con la combinazione di tasti ALT + TAB. Ovviamente puoi anche usare il mouse e fare clic sulla finestra visualizzata nella barra di controllo.

Poiché Gambas utilizza più finestre, è meglio metterle una accanto all'altra in modo che possano essere sempre visualizzate o passare da una all'altra con ALT + TAB.

0.58 raccolta di strumenti

La cassetta degli attrezzi contiene tutti gli elementi di controllo grafici (= controlli) che possono essere utilizzati su un modulo Gambas. Questo va dai campi di designazione ai campi di testo ai numeri LCD e ai timer.

La casella degli strumenti viene richiamata con il tasto funzione F6. È inoltre possibile accedervi tramite il menu File. Se hai una forma aperta e fai doppio clic con il mouse su un elemento di controllo, salta automaticamente alla forma e può essere ulteriormente modificata lì.

Le proprietà di un elemento di controllo vengono chiamate con F6.

All'inizio sono sufficienti alcuni strumenti: ad esempio il pulsante di comando, la casella di testo, il campo immagine, l'area di disegno e il timer. Il resto può essere risolto poco a poco.

Vedi anche Gambas: Controls⁴⁵.

44 <http://gambas.sourceforge.net/screenshots.html>

45 Capitolo 0.60 a pagina 59

0.58.1 Contenuto della raccolta di strumenti



Fig.20 La collezione di strumenti

(Ordina come in foto)

Controllo	descrizione
Freccia di selezione	(Nessun elemento di controllo! Serve per Passare da già nel modulo elementi di controllo posizionati)
Etichetta	Etichetta
TextLabel	Etichetta di testo campo
PictureBox	Scatola per foto
Barra di avanzamento	Barra di avanzamento
Pulsante	Pulsante di comando
CheckBox	Pulsante di selezione

Controllo	descrizione
RadioButton	Pulsante di selezione
Interruttore	interruttore on / off
ToolButton	Pulsante di comando
Casella di testo	Campo di testo
Combo box	Elenco di combinazioni campo
TextArea	campo di testo grande
ListBox	Casella di riepilogo
Visualizzazione elenco	Visualizzazione elenco
Visualizzazione ad albero	Visualizzazione ad albero
IconView	Visualizzazione a icone
Vista a griglia	Vista a griglia
ColumnView	Vista a colonne
HBox	Organizza il contenuto nei controlli orizzontalmente in uno Avanti
VBox	Organizza il contenuto nei controlli verticalmente in uno Colonna attiva
HPanel	Organizza il contenuto nei controlli orizzontalmente se necessario in diversi Righe
VPanel	Organizza il contenuto nei controlli verticale quando necessario in diversi colonne
Telaio	Cornice, ad es. Intorno Pulsanti radio insieme per cogliere
Pannello	Riassumere sen di insieme

	elementi di controllo associati menti
TabStrip	Tab
ScrollView	Visualizza con diapositiva regolatore orizzontale e verticale
DrawingArea	Campo carattere
Timer	Timer

0.58.2 Elementi ricaricabili

Alcuni altri elementi possono essere usati, ma devono essere ricaricati in componenti extra, vedi anche Gambas: Components⁴⁶.

- GambasEditor
- LCDNumber
- Comporre
- SpinBox
- ScrollBar (cursoro)⁴⁷
- Slider
- TableView
- Schegge
- Area di lavoro

0.59 Il menu Gambas



Fig.21 Il menu

46 Capitolo 12.18 a pagina 424

47 Capitolo 0.82 a pagina 85

0.59.1 Menu "File"



Fig.22 II menu "File"

0.59.2 Menu "Progetto"



Fig.23 II menu "Progetto"

0.59.3 Menu "Visualizza"

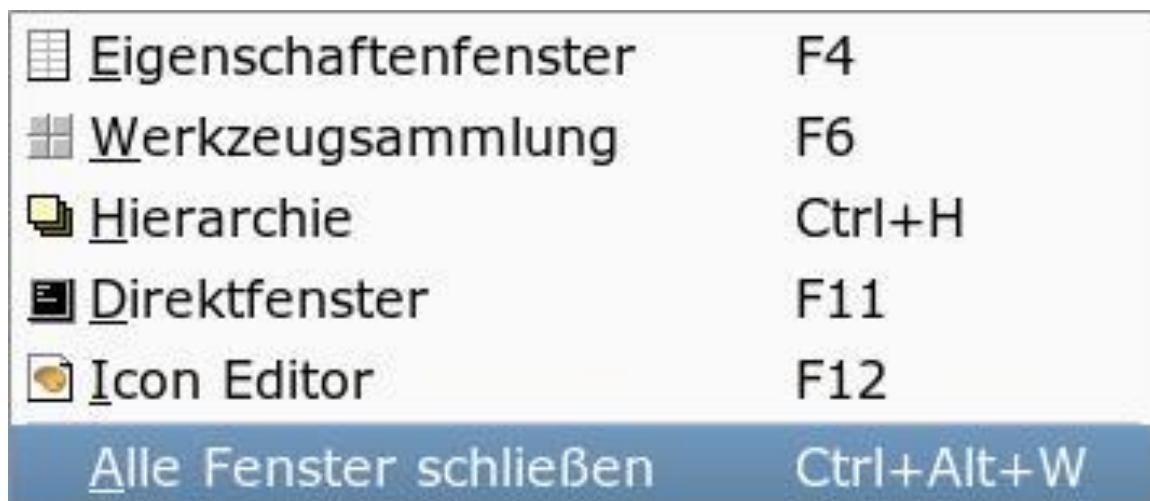


Fig.24 Il menu "Visualizza"

0.59.4 Menu "Strumenti"

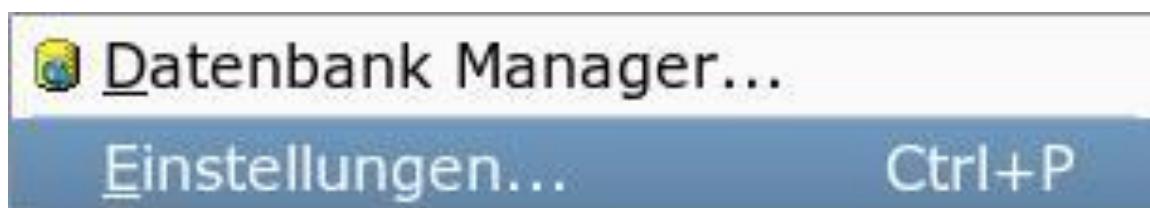


Fig.25 Il menu "Strumenti"

0.59.5 Menu "?"



Fig.26 Il menu "?"

0.60 I tasti funzione in Gambas

Alcuni tasti funzione sono molto utili in Gambas:

F1	Questo richiama il sistema di aiuto come al solito. Ci sono anche molti comandi un paio di programmi di esempio.
F4	Questo richiama la finestra delle proprietà di un oggetto.
F5	Questo avvierà il tuo programma
F6	Questo pulsante mostra la casella degli strumenti
F7	Compila il tuo programma
F8	Si avvia in modalità a passo singolo
F11	Mostra la finestra immediata

0.61 I controlli più importanti (raccolta strumenti / cassetta degli attrezzi)

I controlli più importanti sono i

- Forma o anche chiamata forma
- Il pulsante di comando, (pulsante di comando, pulsante di comando, pulsante OK nella casella degli strumenti)
- La casella di testo (campo abc nella casella degli strumenti)
- L'area di testo (gambas quasi m..field nella casella degli strumenti)
- DrawingArea (campo con simboli grafici nella casella degli strumenti)
- Il timer (timer, campo con orologio nella cassetta degli attrezzi)



Fig.27 Icona del timer in Gamas

Inizia a lavorare con questi. Puoi quindi usare gli altri abbastanza facilmente.



Fig.28 La collezione di strumenti (cassetta degli attrezzi)

0.61.1 proprietà

Ogni elemento di controllo dalla casella degli strumenti ha proprietà diverse (dimensione, colore, cornice, nome, etichetta, ecc.). Le proprietà possono essere specificate in fase di progettazione, cioè prima dell'avvio del programma.

- Il controllo desiderato viene selezionato nella struttura del modulo.
- La finestra delle proprietà viene richiamata con F6.

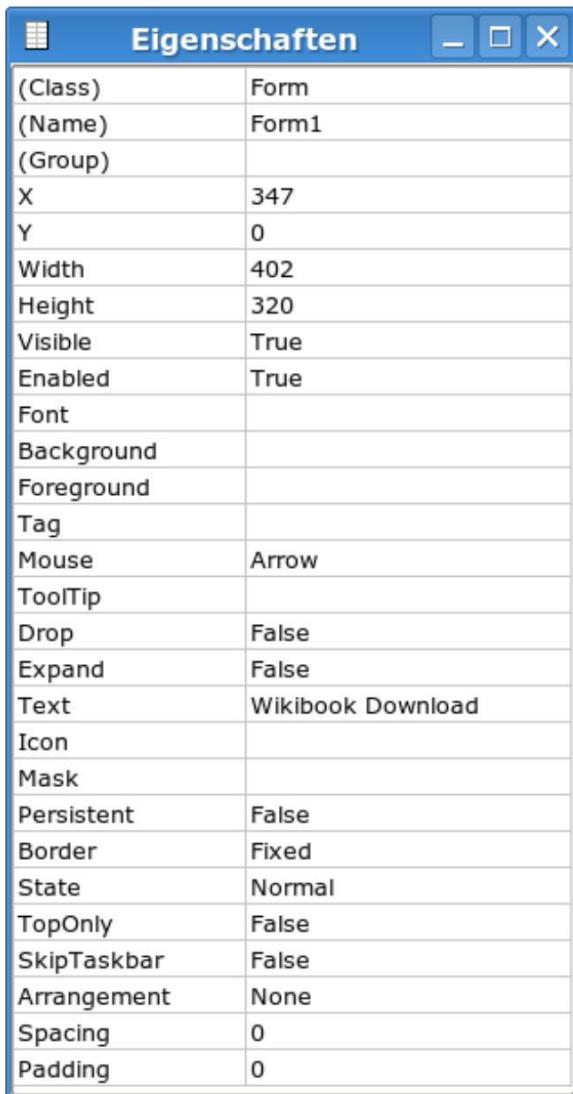


Fig.29 La finestra delle proprietà

- Tutte le proprietà disponibili vengono quindi elencate lì.
- Si sceglie la proprietà desiderata.

Prova questo ad es. B. con la cornice della forma, il nome della forma o la dimensione della forma.

Le proprietà più importanti dei controlli sono descritte di seguito.

Proprietà nome

Questa proprietà si trova in tutti gli oggetti. Determina il nome con il quale l'oggetto è indirizzato nel codice Gambas. Questa proprietà può essere modificata solo in modalità progettazione, non nel programma stesso. L'impostazione predefinita per questa proprietà è, ad esempio, per

primo modulo Form1. Se vuoi fare riferimento a questo modulo da qualche parte nel codice, ad es. se vuoi cambiare una delle sue proprietà, devi usare quel nome.

Proprietà di testo

Questa proprietà si trova nei moduli, nei pulsanti di comando, nelle caselle delle etichette e in molti altri controlli. Per i moduli, specifica il testo che viene visualizzato nella barra del titolo del modulo. Questa proprietà può essere modificata in modalità progettazione così come nel programma stesso (in fase di runtime). Esempio:

```
PUBLIC SUB Form_Open ()  
    Form.Text = "Nuovo titolo"  
FINE
```

In Visual Basic questa proprietà era principalmente chiamata Caption.

Visibile Visibile o nascosto

Questa proprietà è disponibile per i moduli e tutti gli elementi di controllo ad eccezione del timer, perché è sempre invisibile in fase di esecuzione. Questa proprietà viene utilizzata per determinare se un oggetto è visibile o nascosto. Può essere modificato sia in modalità di progettazione che in fase di esecuzione. Questa proprietà può avere solo i valori TRUE (inglese: true: l'oggetto è visibile) o FALSE (inglese: false: false, l'oggetto non è quindi visibile).

Esempio: hai una forma con 2 pulsanti di comando. Se si preme un pulsante, l'altro dovrebbe scomparire.

```
Pulsante PUBLIC SUB1_Click ()  
    Button2.Visible = FALSE  
FINE
```

Il pulsante di comando con il nome Button2 viene cancellato dallo schermo, ovvero reso invisibile.

Abilitato Attivato o disattivato

Questa proprietà è disponibile per i moduli e tutti i controlli. Determina se la forma o il controllo possono rispondere agli eventi generati dall'utente. Con questa proprietà, moduli e controlli possono essere resi disponibili o bloccati in fase di esecuzione. Se gli elementi di controllo sono bloccati, vengono visualizzati in grigio anziché in nero e non possono più essere selezionati. Come la proprietà Visible, Enable può avere solo i valori TRUE (l'oggetto può reagire agli eventi) o FALSE (l'oggetto non può reagire agli eventi).

In questo modo, in modo simile a Visible, puoi bloccare i controlli che attualmente non devono essere utilizzati. Può anche essere utilizzato, ad esempio, per bloccare un campo di testo utilizzato solo per la visualizzazione del testo.

Se si imposta Enabled su FALSE per un timer, viene interrotto e non attiva più alcun evento finché non lo si lascia eseguire di nuovo.

Esempio: il pulsante di comando Button2 è disattivato.

```
Pulsante PUBLIC SUB1_Click ()  
    Button2.Enabled = FALSE  
FINE
```

Colore di sfondo BackColor

BackColor imposta il colore di sfondo per un oggetto e può essere modificato sia in modalità progettazione che in fase di runtime. Esempio:

```
PUBLIC SUB Form_Open ()  
    Form1.BackColor = & 00FF00  
FINE
```

si traduce in uno sfondo giallo nel modulo Form1.

Se si modifica la proprietà BackColor nella finestra delle proprietà, è possibile scegliere uno dei tanti colori predefiniti. Se ciò non è sufficiente, è possibile "mescolare" i propri colori nella finestra della tavolozza dei colori e verranno visualizzati anche questi. Vedi anche la sezione sui comandi grafici.

Il colore dello sfondo può anche essere selezionato tramite una finestra di dialogo standard nel runtime.

Colore di primo piano ForeColor

Nei moduli e in alcuni elementi di controllo, questa proprietà può essere utilizzata per impostare il colore con cui vengono visualizzati lettere e caratteri. Se si modifica questa proprietà in fase di esecuzione, solo le lettere scritte in seguito vengono visualizzate nel nuovo colore.

X, Y corrispondono alle coordinate della posizione

Queste proprietà si trovano nei moduli e in quasi tutti i controlli. Determinano le coordinate dell'angolo superiore sinistro dell'oggetto, rispetto allo schermo per i moduli e rispetto all'angolo superiore sinistro del modulo in cui si trovano per altri controlli.

Entrambe le proprietà possono essere modificate in modalità progettazione: o semplicemente trascinando l'oggetto con il mouse oppure modificandole direttamente nella finestra delle proprietà. D'altra parte, puoi cambiarli in fase di esecuzione e quindi spostare l'oggetto (tranne ovviamente con il timer).

Larghezza, Altezza corrisponde alla larghezza e all'altezza

Queste proprietà determinano la larghezza e l'altezza di un oggetto. Possono anche essere modificati sia in modalità di progettazione che in fase di runtime. La dimensione dell'oggetto cambia quindi immediatamente.

Nel caso di un modulo, l'altezza e la larghezza includono anche i bordi della finestra in alto, in basso e ai lati. Un modo ingegnoso di misurare i controlli è dal centro del bordo di controllo in modo che i controlli con bordi di larghezza diversa possano essere allineati ordinatamente.

0.61.2 Modifica in fase di esecuzione

Le proprietà possono essere modificate anche in fase di esecuzione. Le proprietà vengono quindi modificate nella sequenza del programma solo dopo l'avvio del programma.

Il modulo di comando generale è:

```
Control_name.property_name = property_value
```

Esempio:

Hai inserito un DrawingArea nel modulo. Ora vuoi il file
Cambia colore. Posziona un pulsante di comando per attivare la proprietà durante il
Per modificare il termine:

```
Pulsante PUBLIC SUB1_Click ()  
    DrawingArea1.BackColor = & H00FF00 &  
    FINE
```

Il DrawingArea diventa verde.

Come esercizio, puoi anche modificare le dimensioni di DrawingArea. Come funziona?

```
DrawingArea1.Width = 200  
DrawingArea1.Height = 200
```

Il programma quindi si presenta così:

```
Pulsante PUBLIC SUB1_Click ()  
    DrawingArea1.BackColor = & H00FF00 &  
    DrawingArea1.Width = 200  
    DrawingArea1.Height = 200  
    FINE
```

I nomi dei controlli in inglese vengono assegnati automaticamente da Gambas e resi distinguibili con un numero progressivo (Button1, Button2 ecc.). Questi nomi assegnati automaticamente possono essere modificati con la proprietà Name nella finestra delle proprietà.

0.61.3 Metodi

I metodi vengono utilizzati in modo simile alle proprietà. Sono chiamati dal programma.

I metodi sono piccoli programmi che elaborano le attività del controllo in questione.

Prendi gli appunti come esempio. Ha 3 metodi disponibili: Copia, Incolla e Cancella

Se la programmazione viene eseguita nella finestra del codice, viene visualizzato un elenco delle proprietà e dei metodi disponibili dopo il nome di un componente dopo che il punto è stato scritto.

Una selezione tramite doppio clic semplifica il lavoro qui.

0.61.4 eventi

Gli eventi sono ad es. B. Clic del mouse, movimenti del mouse o una sequenza di tasti.

Il programma Gambas può reagire a tali eventi se lo si desidera. Tutto quello che devi fare è definire un metodo che dica al programma cosa fare.

Esempi di eventi:

L'utente fa clic su un pulsante o su un file no	Fare clic su evento
L'utente modifica un testo in un campo di testo	Cambia evento
Un programma sta terminando perché ce n'è uno da qualche parte Viene raggiunto il comando di fine (utente o professionista sequenza del programma), la finestra viene presa dalla memoria scaricare	Scarica evento
Un timer raggiunge un valore predeterminato	Evento timer
Le dimensioni di una finestra vengono modificate	Ridimensiona evento

Se selezioni un componente nell'editor del modulo con il pulsante destro del mouse e quindi fai clic su

Se si fa clic su Evento, vengono visualizzati tutti gli eventi disponibili per questo componente.

Scegli quello che ti serve.

Gli eventi a cui deve rispondere un controllo vengono impostati in fase di progettazione.

Elenco degli eventi:

- Clic = clic con il mouse (l'evento non viene attivato quando si preme il pulsante del mouse, ma solo quando viene rilasciato!)
- DblClick = doppio clic
- Drag = trascinamento con il mouse
- DragMove
- Disegnare
- Drop = rilascia dopo il trascinamento
- Invio = sposta il mouse in un'area
- GotFocus = Ottieni il marker (o il focus)
- KeyPress = tasto premuto
- KeyRelease

- Lascia = lascia un'area

- LostFocus
- Menù
- MouseDown = premendo qualsiasi pulsante del mouse
- MouseMove
- MouseUP
- Rotellina del mouse
- Aperto

L'evento Click funziona anche in modalità progettazione. Con esso puoi contrassegnare l'elemento di controllo. La finestra del codice associato si apre facendo doppio clic. Il frame per un sottoprogramma viene generato automaticamente con SUB e END e l'evento più frequente di un elemento di controllo è già assegnato.

elemento	Evento standard
forma	Aperto
Pulsante	Clic
Casella di testo	KeyPress
ScrollView	MouseDown
DrawingArea	Disegnare
Timer	Timer
Etichetta	MouseDown
PictureBox	MouseDown
Barra di avanzamento	MouseDown
CheckBox	Clic

Come al solito, inserisci il codice del programma in questo riquadro fornito.

0.62 pulsante di comando

Vedi Gambas: Button⁴⁸

0.63 area di testo

Vedi Gambas: Textarea⁴⁹

0.64 barra di avanzamento = barra di avanzamento

Come puoi programmare la barra di avanzamento? I seguenti esempi dovrebbero mostrarlo:

⁴⁸ Capitolo 0.71 a pagina 70

⁴⁹ Capitolo 0.73 a pagina 76

0.64.1 Da solo

Hai bisogno di una ProgressBar e tre pulsanti di comando per avviare il programma.
Ottieni questi elementi con F6 e un doppio clic sulla tua forma dalla casella degli strumenti.

Il codice sorgente:

```
STATICO PUBBLICO SUB Principale ()  
    hForm AS Fmain  
    hForm = NUOVO Fmain  
    hForm.show  
FINE  
  
PUBLIC SUB _new ()  
    ProgressBar1.value = 0,01  
    'Il valore per value deve essere compreso tra 0 e 1 END  
  
Pulsante PUBLIC SUB1_Click ()  
    ProgressBar1.value = ProgressBar1.value - 0,01  
FINE  
  
PUBLIC SUB Button2_Click ()  
    ProgressBar1.Reset  
FINE  
  
Pulsante PUBLIC SUB 3_Click ()  
    ProgressBar1.value = ProgressBar1.value + 0,01  
FINE
```

0.64.2 Insieme al timer

L'esempio seguente utilizza la barra di avanzamento insieme al timer. A seconda del ritardo del timer, è possibile variare la velocità e quindi z. B. mostrano un processo di ricarica.

Hai bisogno di una barra di avanzamento e di un timer per avviare il programma:

Il codice ha questo aspetto:

```
PUBLIC SUB Form_Open ()  
    «Imposta il timer  
    timer1.Delay = 100  
    timer1.enabled = TRUE  
    'Imposta progressivo  
    ProgressBar1.Value = 0,01  
FINE  
  
PUBLIC SUB Timer1_Timer ()  
    IF ProgressBar1.value <0,99 THEN  
        ProgressBar1.Value = ProgressBar1.Value + 0,01  
    FINISCI SE  
FINE
```

0.65 Il timer (timer, orologio)

Vedi Gambas: Timer⁵⁰

0.66 Il ColumnView

Nella casella degli strumenti c'è un controllo chiamato ColumnView. Il seguente programma mostra come usarlo. È necessario un modulo con un controllo ColumnView sul modulo per farlo funzionare.

```
PUBLIC SUB Form_Open ()
    Creare ()
FINE

PUBLIC SUB Crea ()
    it COME numero intero
    it2 AS intero
    nLines AS Integer
    ColumnView1.Clear
    ColumnView1.Width = 776
    ColumnView1.Height = 464

    "Crea 4 colonne
    ColumnView1.Columns.Count = 4

    'Imposta la larghezza delle colonne
    PER it = da 0 a 3
        ColumnView1.Columns [it] .Width = ColumnView1.Width /
    ColumnView1.Columns.Count
        ColumnView1.Columns [it] .Text = "Titolo" & it
    IL PROSSIMO

    "Crea 5 righe
    nLines = 5
    PER it = da 0 a 4
        ColumnView1.Add (it, it)
        PER it2 = 0 TO ColumnView1.Columns.Count - 1 ColumnView1 [it] [it2]
            = "Col:" & it2 & "Row:" & it
        IL PROSSIMO
    IL PROSSIMO
FINE
```

0.67 Lo slider

Vedi Gambas: barra di scorrimento⁵¹

⁵⁰ Capitolo 0.76 a pagina 80

⁵¹ Capitolo 0.82 a pagina 85

0.68 La forma

Vedi Gambas: Form⁵²

0.69 La vista a griglia

Vedi Gambas: Lattice⁵³

0.70 vista tabella

Vedi Gambas: tabella⁵⁴

0.71 Listview

Vedi Gambas: List⁵⁵

0.72 Il pulsante di comando

Il pulsante di comando è uno dei controlli più importanti di tutti. La maggior parte dei piccoli programmi di esempio in questo Gambas Wikibook vengono avviati tramite un pulsante di comando.

Questo è l'aspetto del pulsante di comando sulla barra degli strumenti. Fai clic su di esso nella barra degli strumenti se ne hai bisogno sulla tua forma. Ottieni la barra degli strumenti con F6.

52 Capitolo 0.84 a pagina 86

53 Capitolo 0.100 a pagina 97

54 Capitolo 0.105 a pagina 101

55 Capitolo 0.111 a pagina 104

Il pulsante di comando



Fig.30

Si prega di attenersi al seguente ordine. Per prima cosa visualizza la forma, quindi attiva la casella degli strumenti, altrimenti non otterrai un pulsante di comando sulla forma.

Questo è l'aspetto del pulsante di comando sulla tua forma in visualizzazione struttura.

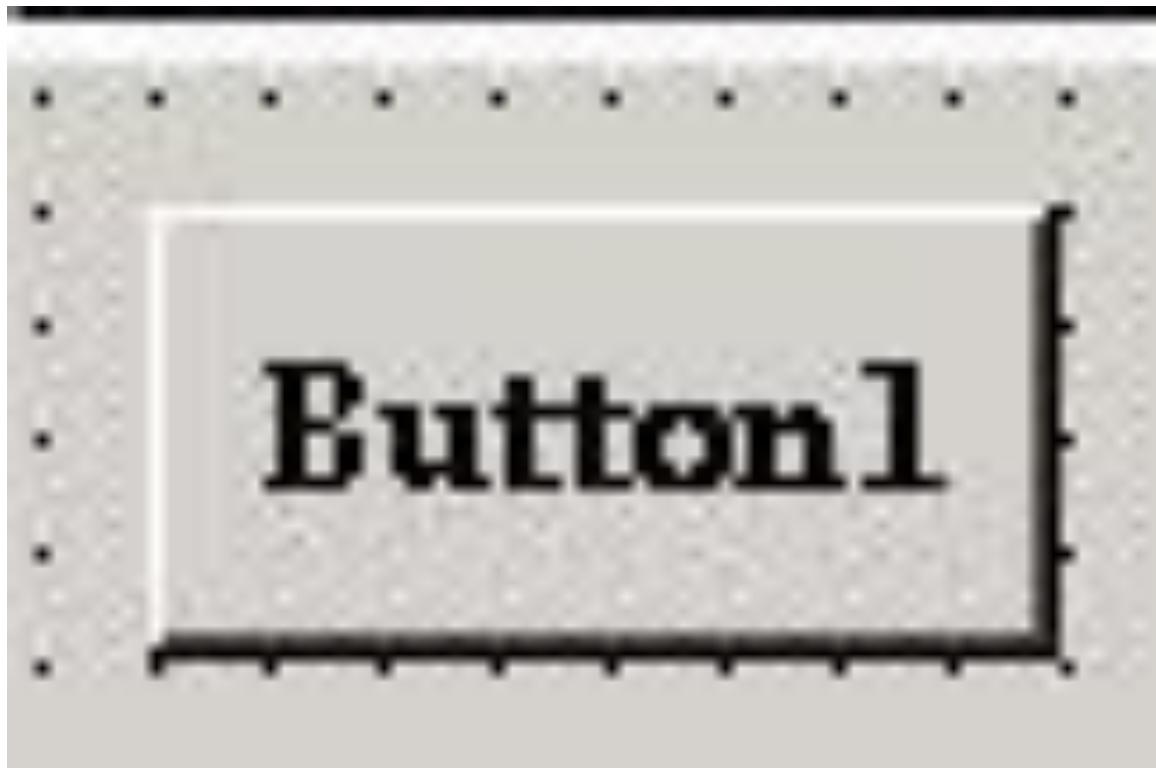


Fig.31

Se fai clic sul pulsante di comando nella vista struttura, verrà contrassegnato con 4 piccoli quadrati bianchi agli angoli. Puoi ingrandirlo o rimpicciolirlo e ovviamente spostarlo ovunque sulla tua forma.

Con F4 puoi definire le proprietà del pulsante. Non lasciare che i grandi numeri ti impressionino. Ad esempio, prova la proprietà Text e assegna un nuovo nome al tuo pulsante.

Se desideri allegare il codice al pulsante, apri la finestra del codice facendo doppio clic sul pulsante di comando.

La gestione del pulsante di comando è fondamentalmente molto semplice. Lo tiri fuori dal Casella degli strumenti sulla forma, rinominala e poi scrivi il codice appropriato dietro di essa:

```
Pulsante PUBLIC SUB1_Click ()  
Stampa 1  
FINE
```

0.72.1 Richiamo del pulsante di comando con la tastiera

Se il pulsante è attivo mentre il programma è in esecuzione, reagisce all'evento "KeyPress".

Il codice per gestire questo evento è quindi, ad esempio:

```
PUBLIC SUB Button1_KeyPress ()  
    IF key.Code = key.Space OR key.Code = key.Return OR key.code =  
key.Enter THEN  
        Stampa 1  
        FINISCI SE  
    FINE
```

I seguenti programmi dovrebbero mostrare alcune caratteristiche speciali del pulsante di comando.

0.72.2 Fare scomparire il pulsante di comando

È possibile rendere invisibile un pulsante di comando con il seguente mini programma:

```
Pulsante PUBLIC SUB1_Click ()  
    Button1.visible = false  
FINE
```

A proposito, la stessa cosa fa:

```
Pulsante PUBLIC SUB1_Click ()  
    Button1.hide  
FINE
```

Se dovesse apparire di nuovo, il codice seguente aiuterà con un Button2:

```
PUBLIC SUB Button2_Click ()  
    Button1.show  
FINE
```

Come si potrebbe visualizzare di nuovo il Button1 senza utilizzare un Button1?

Ad esempio, con un doppio clic sulla forma:

```
PUBLIC SUB Form_DblClick ()  
    Button1.show  
FINE
```

0.72.3 Far scomparire definitivamente il pulsante di comando

Con il seguente mini-programma è possibile far scomparire completamente un pulsante di comando.

```
Pulsante PUBLIC SUB1_Click ()  
    Button1.delete  
FINE
```

0.72.4 Spostare il pulsante di comando

Se si desidera spostare il pulsante di comando durante il runtime, è possibile utilizzare il seguente mini-programma. È necessario un pulsante di comando nel modulo per farlo funzionare.

```
PUBLIC SUB Form_Open ()  
    Button1.Move (160,64,64,32)  
    FINE  
Pulsante PUBLIC SUB1_Click ()  
    Button1.Move (1,64,64,32)  
    FINE
```

Come puoi spostare il pulsante con il mouse?

Ti piacerebbe provare come un pulsante segue il tuo mouse come un cucciolo?

Quindi prova il seguente programma. Hai bisogno di una forma e di un pulsante per farlo funzionare.

```
PUBLIC SUB Form_MouseMove ()  
    Button1.X = Mouse.X  
    Button1.Y = Mouse.Y  
    FINE
```

Una volta avviato il programma, fare clic sulla forma e spostare il mouse attorno alla forma.

0.72.5 Creare un nuovo pulsante di comando in fase di esecuzione

Nella maggior parte dei casi, inserirai tutti i pulsanti di comando necessari nel modulo durante la fase di progettazione, aggiungerai codice e solo successivamente avvierai il programma. È inoltre possibile creare uno o più pulsanti di comando mentre il programma è in esecuzione (runtime). Come farlo è descritto qui.

Hai solo bisogno di una forma vuota per farlo funzionare.

```
b Pulsante AS  
PUBLIC SUB _New ()  
    b = pulsante NUOVO (ME) COME "Fine"  
    b.mostra  
    b.Caption = "end"  
    bx = 100  
    di = 100  
    FINE  
  
PROCEDURA PUBBLICA End_Click ()  
    ME.Chiudi  
    FINE
```

Una variante divertente:

```
b Pulsante AS  
PUBLIC SUB _New ()  
    b = pulsante NUOVO (ME) COME "Fine"
```

```
b. mostra  
b.Caption = "end"  
bx = 100  
di = 100  
FINE  
  
PROCEDURA PUBBLICA End_Click ()  
    b. visibile = FALSO  
FINE  
  
PUBLIC SUB Form_Leave ()  
    b. visibile = VERO  
FINE
```

0.72.6 Ricreare i pulsanti di comando in un ciclo in fase di esecuzione

Se vuoi creare rapidamente molti pulsanti su una forma, un ciclo è più adatto. Ma ci sono alcune catture. Funziona abbastanza bene con il seguente programma di esempio. Tutto ciò di cui hai bisogno è una forma vuota per farlo funzionare.

```
btn AS pulsante  
btnnameAS Stringa  
  
PUBLIC SUB _new ()  
    x AS intero  
    PER x = da 1 a 4  
        btnname = "btn" & Str (x)  
        btn = NUOVO pulsante (ME) COME btnname  
        btn.X = 100 * X  
        btn.Y = 30  
        btn.Larghezza = 50  
        btn.Height = 30  
        btn.didascalia = Str (x)  
        btn.enabled = TRUE  
    IL PROSSIMO  
FINE
```

Se vuoi eliminare tutti i pulsanti in un ciclo, questo è molto più difficile.

0.72.7 Il pulsante di comando errante

Il seguente programma è completamente inutile, ma nel gioco c'è movimento. Hai bisogno di una forma e di un pulsante per farlo funzionare.

```
Pulsante PUBLIC SUB1_Click ()  
    x AS intero  
    PER x = da 1 a 20  
        Pulsante1.X = x * 10  
        Button1.Y = x * 10  
        ATTENDERE 0.5  
        Button1.Text = Str (x)  
    IL PROSSIMO  
FINE
```

0,73 tassi di numero fino a 27

Nel seguente giochino, il computer pensa a un numero compreso tra 1 e 27. L'utente dovrebbe indovinare questo numero con il minor numero di tentativi possibile. Prova a cavartela con un massimo di 5 supposizioni!

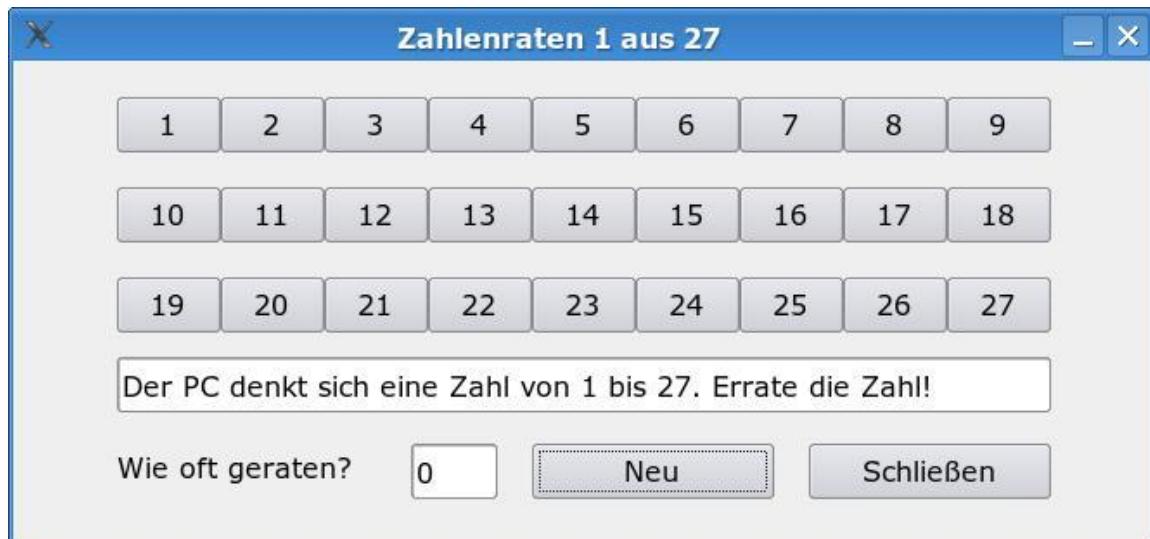


Fig.32

Sono necessari 2 pulsanti di comando, 2 caselle di testo e 1 controllo dell'etichetta sul modulo per avviare il programma.

Codice sorgente: tassi numerici Form1.class⁵⁶

I 27 pulsanti numerici vengono generati in fase di esecuzione. La gamma di pulsanti è sorprendente. In modo da non dover scrivere una funzione Click () separata per tutti i 27 pulsanti, vengono forniti tutti i pulsanti nella riga

b [n] = NUOVO pulsante (ME) COME "btn"

lo stesso nome btn. Di conseguenza, tutti i 27 pulsanti chiamano la stessa funzione btn_Click () quando vengono selezionati. Per determinare all'interno di questa funzione quale pulsante è stato cliccato, utilizziamo LAST. LAST è un riferimento all'ultimo oggetto che ha generato un evento. Possiamo quindi accedere all'oggetto corrispondente tramite LAST. Per identificarlo chiaramente ora, usiamo la sua proprietà tag. La proprietà Tag può essere utilizzata per memorizzare in modo invisibile informazioni in un oggetto, nel nostro caso il numero assegnato al pulsante. All'interno della funzione btn_Click (), LAST.Tag restituisce il valore numerico del pulsante su cui è stato fatto clic.

L'area di testo è un elemento di controllo utilizzato di frequente che, a differenza della casella di testo, offre spazio anche per testi più grandi.

⁵⁶ <http://de.wikibooks.org/wiki/Gambas%3A%20Button%3A%20Zahlenrechner%20Form1>. classe

0.74 Le proprietà, i metodi e gli eventi dell'area di testo

Se vuoi conoscere meglio l'area di testo, puoi dare un'occhiata al file

- le proprietà
- i metodi
- e gli eventi

L'area di testo viene visualizzata al meglio nella funzione di aiuto di Gambas.

Si preme F1 o si seleziona la voce di menu? Dal menu nella finestra del progetto. (Punto interrogativo), lì prendi la prima voce di menu Aiuto Browser F1. (Vedi anche Gambas: Aiuto⁵⁷) Quindi selezionare Componenti nella pagina della documentazione di Gambas. Seleziona il componente gb.qt dai componenti e cerca la classe Textarea. Tutte le proprietà, i metodi e gli eventi dell'area di testo possono essere trovati qui.

Puoi trovare il tutto molto rapidamente su Internet all'indirizzo: gambasdoc.org⁵⁸

0.74.1 Proprietà dell'area di testo

BackColor, Sfondo, Bordo, Colonna, Cursore, Caduta, Abilitato, Espandi, Carattere, ForeColor, Primo piano, H, Maniglia, Altezza, Id, Sinistra, Lunghezza, Linea, Mouse, Avanti, Genitore, Pos, Precedente, Sola lettura, ScreenX, ScreenY, Select, Selection, Tag, Text, ToolTip, Top, Visible, W, Width, Window, Wrap, X, Y

0.74.2 Metodi dell'area di testo

Cancella, Copia, Taglia, Elimina, Trascina, AssicuratiVisibile, Afferra, Nascondi, Inserisci, Abbassa, Sposta, Incolla, Alza, Ripeti, Aggiorna, Ridimensiona, Imposta messa a fuoco, Mostra, ToColumn, ToLine, ToPos, Annulla

0.74.3 Eventi dell'area di testo

Cambia, Cursore, Doppio clic, Trascina, DragMove, Rilascia, Invio, GotFocus, KeyPress, KeyRelease, Esci, LostFocus, Menu, MouseDown, MouseMove, MouseUp, MouseWheel

Per ciascuno di questi punti troverai alcune informazioni di aiuto speciali. Tuttavia, questi sono disponibili solo in inglese.

57 Capitolo 0.17 a pagina 21

58 <http://www.gambasdoc.org/help/comp/gb.qt/textarea>

0.75 Come puoi saltare alla fine del testo nella TextArea con il cursore?

Per questo è necessaria la proprietà Pos di TextArea. È un numero (intero) e puoi usare questo numero per interrogare o impostare la posizione del cursore. Il seguente piccolo esempio mostra come funziona:

Hai bisogno di un `TextArea` e di un pulsante di comando per ottenere l'esempio:

Pulsante PUBLIC SUB1_Click ()
TextArea1.Pos = TextArea1.Length
'Il cursore va alla fine del testo
FINE

TextArea1.Pos conta i caratteri dall'inizio del testo.

0.76 Tipo di testo Gambas

@ @ @ @

@ @ @

* @ @ @ @ @ @ @ * W #

W. * +

@ @ * +++++++ @ @ # W

@ @ +++++++ # @ @ @ @ @ WW @ +

W, WW

@ @ @ # +++++++ @ @ *# +++++++ @ @ *# +W.

@ ++ * @ # @ +++++++ @ @ *# . , W + W:

@ +++++ @ @ @ @ # + @ +++++++ @ *++++ # .. , WWWWW

WWW W

@ *++++ @ @ @ * @ +++ @ +++++++ @ # +++++ *#

WW. * @ WWW W

@ ++ @ @ # + + @ +++ @ +++++++ * @ W + @ + ,

+ ++++++ W, W.

#

@ + # @ +++++ @ +++ @ +++++++ @ *++++++ W +: W + + @ + W

@

@ # +++++ @ # + + @ +++++++ @ +++++++ # W # +++++ @

@ #

@ @ # +++++ @ @ + @ +++++++ @ +++++ :: +++++++ @

@

@ @ +++++ # @ @ +++++++ * @ +++++ +++++++ # @

@ @ #

@ @ @ * + * @ @ +++++++ * @ +++++ +++++++ @ @

@ # + @ @

@ ++ * @ @ + + @ @ # +++++++ @ +++++ +++++++ @ @

+ +++++++ @ @ @ + + + @

@ + + + * @ + + + * # @ @ @ @ @ @ * + @ +++++ . +++++++

@ @ + + + @ + + @


```
@@ @@@@*  
@@ # +++ ##  
@@ @@@*  
@@ @@@@*  
@@ @@@@*
```

0.77 Il programma orario più semplice

Un timer è un elemento di controllo invisibile che rende programmabile una sequenza temporale.

Se il timer viene attivato con la proprietà Enabled, il metodo viene avviato periodicamente ancora e ancora con il ritardo di tempo specificato.

Posiziona un timer sul modulo facendo clic sul campo corrispondente nella casella degli strumenti.

Il timer deve essere acceso se deve funzionare.

Quindi commutare la sua proprietà Enabled su TRUE nella finestra delle proprietà (F4).

Quindi fare clic sul timer e inserire il seguente codice:

```
PUBLIC SUB Timer1_Timer ()  
    STAMPA 1  
    FINE
```

Sequenza del programma:

A seconda di quanto grande si imposta il valore per il ritardo, la stampa sarà più veloce o più lenta. Questo viene ripetuto più e più volte.

Variante: l'uscita viene interrotta con un pulsante.

```
PUBLIC SUB Timer1_Timer ()  
    STAMPA 1  
    FINE  
  
Pulsante PUBLIC SUB1_Click ()  
    Timer1.Enabled = FALSE  
    FINE
```

0.78 Ritardo di modifica in fase di esecuzione

All'inizio, il timer ha sempre un ritardo di 1000 millisecondi ed è Disabilitato (Enabled = FALSE). Quindi deve essere acceso prima.

Queste impostazioni predefinite possono essere modificate o programmate nelle proprietà.

Avvia un nuovo progetto Timer1. Ottieni una nuova forma con il tasto destro del mouse sulla forma nel progetto. Chiama la forma come vuoi. Dichiara la forma come forma iniziale.

È possibile impostare il ritardo nel timer in millisecondi. Se il timer viene attivato con la proprietà Enabled, il metodo Timer1_Timer () viene avviato periodicamente con il ritardo di tempo specificato.

Posiziona un timer e 2 pulsanti di comando nel modulo facendo clic sui campi corrispondenti nella casella degli strumenti.

Il tutto sembra così: sposta la tua forma in modo da poter vedere la finestra diretta!

Il codice del programma per questo:

```
' File di classe Gamas  
  
PUBLIC n AS Intero  
  
PUBLIC SUB Form_Open ()  
    Timer1.Enabled = TRUE  
FINE  
  
PUBLIC SUB Timer1_Timer ()  
    PRINT Timer1.Delay  
    Timer1.Delay = 1000 + n  
FINE  
  
Pulsante PUBLIC SUB1_Click ()  
    n = n + 200  
FINE  
  
PUBLIC SUB Button2_Click ()  
    n = n - 200  
FINE
```

Sequenza del programma: il valore del ritardo può ora essere cambiato più velocemente o più lentamente

Variazione:

Se aggiungi una TextArea al tuo form, verrà visualizzata nella TextArea se utilizzi la seguente variazione di codice.

```
' File di classe Gamas  
  
PUBLIC n AS Intero  
  
PUBLIC SUB Form_Open ()  
    Timer1.Enabled = TRUE  
FINE  
  
PUBLIC SUB Timer1_Timer ()  
    TextArea1.Text = Timer1.Delay & "\n" & TextArea1.Text  
    Timer1.Delay = 1000 + n  
FINE  
  
Pulsante PUBLIC SUB1_Click ()  
    n = n + 200  
FINE  
  
PUBLIC SUB Button2_Click ()  
    n = n - 200  
FINE
```

0.79 Modificare il ritardo con numeri casuali

Nel seguente programmino vengono generati numeri casuali compresi tra 1 e 25. Questi numeri vengono emessi nella finestra diretta e allo stesso tempo il ritardo del timer viene modificato a seconda della dimensione del numero. Per avviare il programma è necessario un modulo e un controllo timer su di esso. Puoi ottenerlo dalla casella degli strumenti con F6.

```
PUBLIC SUB Form_Open ()  
    Timer1.Enabled = TRUE  
FINE  
  
PUBLIC SUB Timer1_Timer ()  
    x AS intero  
    Rendi casuale  
    x = Int (Rnd (1.25))  
    STAMPA x  
    Timer1.Delay = x * 50  
FINE
```

Piuttosto inutile questo programma, ma puoi abituarti al timer e giocare un po' con numeri casuali.

Conto alla rovescia 0,80

Questo programma proviene da Jonás Alvarez. Vedi: linuxfocus.org⁵⁹

È stato tradotto in tedesco, migliorato e commentato:

Hai bisogno di un timer, una casella di testo, un pulsante di attivazione / disattivazione, un campo etichetta e 2 pulsanti di comando per farlo funzionare.

Il codice:

```
CONST fStandart AS Float = 20.0  
fseconds AS float  
'seconds è già definito al di fuori di tutte le subroutine,' quindi può essere utilizzato in  
tutti gli UP.  
'Seconds è il tempo corrente in secondi fino all'inizio' fStandart è il tempo di  
inizio del conto alla rovescia. È fissato come una costante.
```

fStandart può, tuttavia, tramite il pulsante di comando di ripristino Button1_Click ()
'e la casella di testo può essere modificata.

```
PUBLIC SUB Form_Open ()  
    'All'inizio del programma, alcune cose vengono impostate e visualizzate  
    fseconds = fStandart  
    Schermo  
    ToggleButton1.Value = FALSE  
    Acceso spento  
    Textbox1.Text = fseconds  
FINE
```

```
PUBLIC SUB ToggleButton1_Click ()  
    'questo è il pulsante più importante
```

59 <http://linuxfocus.org/Deutsch/March2004/article329.shtml>

```
'avvia il conto alla rovescia o lo interrompe
Timer1.Enabled = Togglebutton1.Value
" Se si preme il pulsante di attivazione / disattivazione, è possibile eseguire
anche il timer. 'ToggleButton1.Value = True >> Timer1.Enabled = True' Se
non viene premuto, il timer si ferma 'ToggleButton1.Value = False >>
Timer1.Enabled = False OnOff
```

FINE

```
Pulsante PUBLIC SUB1_Click ()
fseconds = Val (Textbox1.Text)
Schermo
FINE
```

```
PRIVATO SUB OnOff ()
IF ToggleButton1.Value THEN
    ToggleButton1.Text = "& conto alla rovescia in esecuzione"
ALTRO
    ToggleButton1.Text = "& interruzione"
FINISCI SE
'Quando si fa clic sul pulsante di attivazione / disattivazione, viene visualizzato il testo seguente:
«Countdwon sta correndo.
'Se non viene cliccato, viene visualizzato il testo Interrompi END
```

```
Display PRIVATO SUB ()
DIM fminutes AS Integer
fminutes = Int (Int (fseconds) / 60)
'il comando INT taglia i numeri decimali' che avviene qui due volte
«Vengono tagliati solo i centesimi di secondo
'Quindi viene convertito in minuti e tutto viene tagliato dopo il punto decimale.
```

```
Label1.Caption = fminutes & ":" & format (fseconds - fminutes * 60, "00.0")
'il comando format offre molte possibilità. 'Dai un'occhiata con F1 in
Aiuto
FINE
```

```
PUBLIC SUB Timer1_Timer ()
SE fsecondi <0,1 ALLORA
    Label1.Caption = "0: 00,0"
    Togglebutton1.Value = FALSE
    Togglebutton1_Click
    'Qui il pulsante di commutazione è disattivato dal programma' perché il conto alla
    rovescia è vicino allo zero
    'il programma salta alla subroutine Togglebutton1_Click ELSE
    fseconds = fseconds - 0,1
    Schermo
FINISCI SE
FINE
```

```
PUBLIC SUB Button2_Click ()
ME.Chiudi
FINE
```

Se si preme il pulsante di attivazione / disattivazione, l'orologio inizia a contare all'indietro. Se lo premete di nuovo, il conto alla rovescia viene interrotto. Con Reset puoi modificare l'inizio del conto alla rovescia tramite la casella di testo.

Il programma ha 2 procedure OnOff e Display.

Tempo di risposta del test 0,81

Con questo piccolo programma si misura il tempo di reazione.

La funzione timer viene utilizzata per questo programma. Questo non deve essere confuso con l'elemento di controllo del timer. Inoltre, la visualizzazione della frenata viene emessa solo dopo un tempo casuale iniziale.

Sono necessari 3 pulsanti di comando e una casella di testo per avviare il programma. Il programma necessita ancora di miglioramenti, ma funziona.

Codice del programma:

```
fStart AS Float  
  
Pulsante PUBLIC SUB1_Click ()  
    f Random AS float  
    i AS intero  
  
    Rendi casuale ()  
  
    fZufall = Rnd (0,5, 3)  
    'PRINT f Casuale  
    ATTENDERE f Casuale  
  
    TextBox1.Text = "Frena !!!"  
  
    fStart = Imposta l'ora di inizio del timer.  
    "D'ora in poi, il tempo di risposta viene misurato solo  
FINE  
  
PUBLIC SUB Button2_Click ()  
    f Tempo di reazione AS float  
  
    fTempo di reazione = timer - fStart  
    fReaction time = Int (fReaction time * 1000)  
    TextBox1.Text = fTempo di reazione  
    "TextBox1.Text = " Hai "+ Str $(fReaktionzeit) +"  
    Ci sono voluti millisecondi per frenare! "  
FINE  
  
Pulsante PUBLIC SUB 3_Click ()  
    ME.Chiudi  
FINE
```

0.82 Il comando wait, un'alternativa al timer

In un ciclo non è così facile programmare correttamente il timer. Prova semplicemente il comando WAIT come alternativa.

La sintassi del comando è:

ATTENDERE [Ritardo]

Il comando chiama il loop degli eventi. Se viene specificato un valore per Ritardo, il computer attende fino allo scadere del tempo di ritardo specificato. Il ritardo del numero è uno

Numero in virgola mobile. Quindi, se vuoi aspettare 100 ms, inserisci quanto segue:

```
ATTENDERE 0.1
```

Nessun input del mouse o della tastiera viene elaborato durante la pausa. Solo gli eventi di disegno, timer e descrizione file sono ancora in esecuzione.

Un piccolo mini-programma mostra l'uso del comando. Hai bisogno di un pulsante di comando e di una casella di testo per farlo funzionare. Devi aspettare 5 secondi prima che accada qualcosa.

```
Pulsante PUBLIC SUB1_Click ()
ATTENDI 5
'attende 5 secondi

ATTENDERE 0.1
'attende 100 millisecondi

TextBox1.Text = ""
FINE
```

O cosa dici a questo programma con il pulsante di comando mobile: hai bisogno di una forma e di un pulsante per farlo funzionare.

```
Pulsante PUBLIC SUB1_Click ()
DIM x AS intero
PER x = da 1 a 20
    Pulsante1.X = x * 10
    Button1.Y = x * 10
    ATTENDERE 0.5
    Button1.Text = Str (x)
    IL PROSSIMO
FINE
```

0.83 ScrollBar (cursore)

Con il cursore puoi inserire e produrre valori numerici continui in modo molto piacevole.

Devi prendere il cursore dalla cassetta degli attrezzi QT, poiché non può essere trovato nella normale cassetta degli attrezzi di Gambas.

A tale scopo, caricare il componente `gb.qt.ext`⁶⁰. Puoi farlo tramite la voce di menu Progetto -> Proprietà della finestra del progetto Gambas. Anche il componente `gb.qt.ext` viene selezionato nella visualizzazione del componente.

Un pulsante etichettato QT viene quindi visualizzato nella casella degli strumenti. Fai clic su questo e poi hai una serie di altri controlli tra cui la ScrollBar.

60 Capitolo 12.18 a pagina 424

0.84 programma di esempio

Il programma di esempio è molto semplice. Hai bisogno del cursore e di una casella di testo nel modulo.

Se si aziona lo slider con il mouse come di consueto, nella casella di testo viene visualizzato un valore numerico compreso tra 0 e 99. Se la barra di scorrimento cambia (evento di modifica), il valore della barra di scorrimento (valore) viene trasferito nella casella di testo.

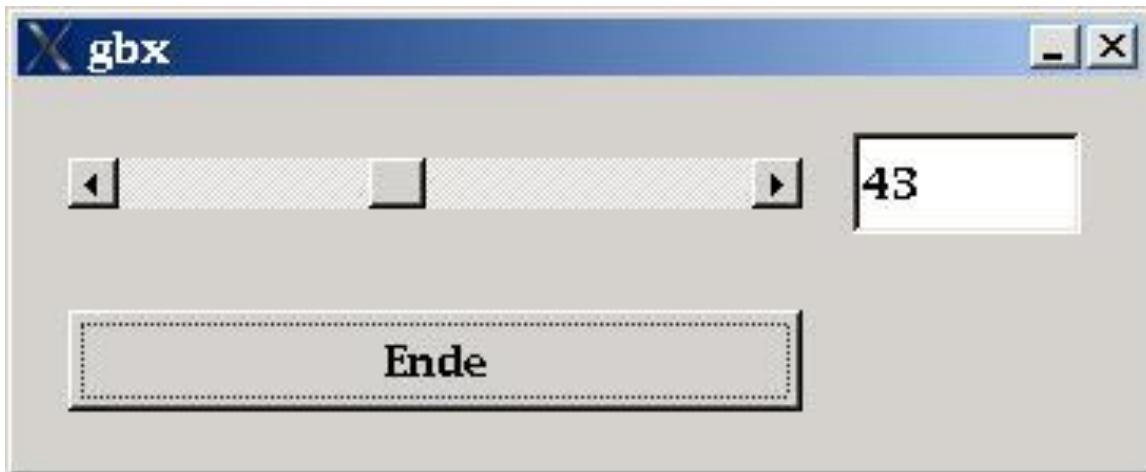


Fig.33 Screenshot barra di scorrimento

Il codice seguente è dietro il cursore:

```
PUBLIC SUB ScrollBar1_Change ()  
    TextBox1.Text = ScrollBar1.Value  
FINE
```

Guarda anche <http://www.madeeasy.de/7/prgmini.htm#form>

0.85 Come si crea un nuovo modulo?

Nell'IDE Gambas, fai clic con il pulsante destro del mouse su Moduli. Quindi scegli Nuovo e Modulo. Di solito si utilizza il modulo selezionato come modulo di partenza e poi lo si deve indicare con la piccola freccia verde. Oppure hai già un modulo di inizio o una classe di inizio, quindi scegli semplicemente un modulo senza la freccia verde.

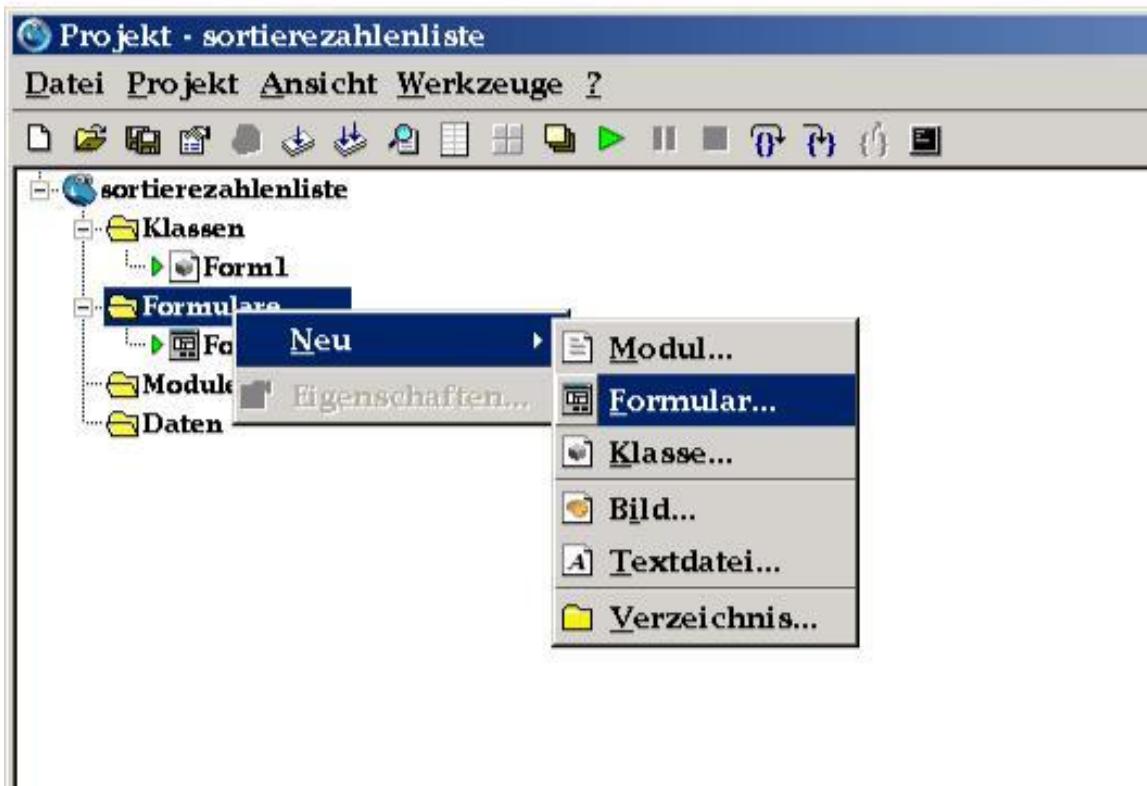


Fig.34

0,86 Semplice in tema di forma

Quanto segue spiega alcune cose sul modulo di cui hai bisogno più e più volte.

Esecuzione delle istruzioni direttamente all'apertura del modulo

```
PUBLIC SUB Form_Open ()
    Istruzione_1
    ...
    Istruzione_N
FINE
```

Esecuzione delle istruzioni quando le incontri con il mouse

```
PUBLIC SUB Form_Enter ()
    Istruzione_1
    ...
    Istruzione_N
FINE
```

Modificare il nome di un modulo con

Contenuti

```
ME.Text = "nuovo titolo"
```

Chiude un modulo

```
ME.Chiudi
```

0,87 domande

Come puoi visualizzare 2 moduli contemporaneamente?

```
STATICO PUBBLICO SUB Principale ()  
    Form1.Show  
    Form2.Show  
FINE
```

o più di 2 contemporaneamente

```
STATICO PUBBLICO SUB Principale ()  
    Form1.Show  
    Form2.Show  
    Form3.Show  
    Form4.Show  
    ....  
FINE
```

Come puoi visualizzare 2 moduli contemporaneamente senza che si sovrappongano?

La Forma 2 è a destra della Forma 1:

```
STATICO PUBBLICO SUB Principale ()  
    Form1.Show  
    Form2.X = Form1.X + Form1.ClientWidth  
    Form2.Y = Form1.Y  
    Form2.Show  
FINE
```

La forma 2 è sotto la forma 1:

```
STATICO PUBBLICO SUB Principale ()  
    Form1.Show  
    Form2.X = Form1.X  
    Form2.Y = Form1.Y + Form1.ClientHeight  
    Form2.Show  
FINE
```

C'è ancora una leggera sovrapposizione; Il motivo è probabilmente che la larghezza e l'altezza del telaio della finestra non vengono conteggiate in ClientWidth o ClientHeight.

0,88 forma iniziale

Gambas assegna automaticamente la classe principale al modulo di partenza. Ciò semplifica la programmazione all'inizio, ma rende più difficile la gestione di diversi moduli.

Forma modale 0,89

Se si richiama un modulo modale, tutte le altre finestre vengono automaticamente rese passive. Quindi puoi solo continuare con il modulo modale.

L'esecuzione del programma del form chiamante viene interrotta nel punto in cui viene chiamato il form modale fino al termine del form modale (se viene chiamato un form non modale, l'esecuzione del programma del form chiamante continua in parallelo).

Viene richiamata una forma modale con:

```
Form2.ShowModal
```

Sfortunatamente, c'è un piccolo difetto qui: il modulo di chiamata diventa passivo, non può più essere utilizzato, ma gli elementi di controllo non sono "disattivati" e suggeriscono l'usabilità all'utente (versione 1.0.15 di Gambas). Puoi rimediare con il seguente codice:

```
ME.Enabled = False
Form2.ShowModal
ME.Enabled = True
```

La 3a riga è necessaria in modo che il modulo chiamante possa essere utilizzato nuovamente dopo che la finestra di dialogo modale è terminata.

Questo vale anche per le finestre di dialogo modali predefinite

```
Messaggio
message.Info
messaggio.Errore
messaggio Elimina
messaggio di avvertimento
messaggio Domanda
```

Questa sezione deve ancora essere elaborata in dettaglio.

Guarda anche: <http://www.binara.com/gambas-wiki/static/Gambas/ModalForm.html> [attualmente non disponibile]

0.90 proprietà di una forma

Disposizione Indietro Colore Sfondo Bordo Didascalia Bambini ClientH ClientHeight ClientW Larghezza client ClientX ClientY Cursore Design Drop abilitato Espandi carattere ForeColor Primo piano H Handle

Altezza Icona Id Sinistra Maschera Menu Modale Mouse Imbottitura successiva Genitore Persistente Schermata precedente X Schermata Y Salta Barra delle attività Spaziatura Stato Tag Testo Titolo ToolTip In alto In altoSolo visibile W Larghezza finestra XY

W o larghezza

fornisce la larghezza del modulo, ma meno gli elementi della finestra (cornice sinistra e destra)

H o Altezza

fornisce l'altezza del modulo, ma meno gli elementi della finestra (cornice sopra inclusa intestazione e cornice sotto)

0.91 metodi di un modulo

centro

Centra il modulo nell'area di lavoro dello schermo

Chiudi Elimina Trascina Trova Afferra Nascondi Inferiore Sposta Sposta Ridimensionato Alza Aggiorna Reparent Ridimensiona RidimensionaScalato SetFocus Mostra MostraDialog MostraModale

0.92 Eventi di un modulo

Attiva Chiudi Doppio clic Disattiva Trascina Trascina Trascina Rilascia Invio GotFocus Nascondi tasto Premi tasto Rilascia lascia perdereMenu messa a fuoco MouseDown MouseMove MouseUp MouseWheel Sposta Apri Ridimensiona Mostra

0.93 Costanti della forma

Risolto Schermo intero Ingrandito Minimizzato Nessuno Normale Ridimensionabile

0,94 ordine di tabulazione del modulo

L'ordine di tabulazione è l'ordine in cui vengono eseguiti gli elementi di controllo grafici quando si preme il tasto tab. Può essere modificato nella finestra della gerarchia dell'IDE.

0,95 Nuovo modulo

Vuoi lavorare con 2 moduli e saltare avanti e indietro da un modulo all'altro:

Esempio: sono necessari 2 moduli e un pulsante di comando su ogni modulo.

Il codice per Form1.class è:

```
Pulsante PUBLIC SUB1_Click ()  
Form2.Show  
Form1.Hide  
FINE
```

Il codice per Form2.class è:

```
Pulsante PUBLIC SUB1_Click ()  
Form1.Show  
Form2.Close  
FINE
```

Notare che le due forme non sono uguali. Form2 è un oggetto di classe iniziale per Form1. Form1 è la cosiddetta classe principale

Se inserisci il codice seguente in Form1:

```
Pulsante PUBLIC SUB1_Click ()  
Form2.Show  
Form1.Close  
FINE
```

il programma non funziona perché termina prematuramente. Anche Form2 non viene visualizzato poiché dipende da Form1.

Questa dipendenza può anche influire sulle proprietà di Form2.

Per favore, riferisci <http://www.madeeasy.de/7/prgnewform.htm>

0.96 Trasferimento dati da un modulo all'altro

Per favore, riferisci <http://www.madeeasy.de/7/prgformtoformplus.htm> L'intera cosa è molto complicata. Forse può essere ancora più facile.

C'è un modo più semplice: vedere il prossimo capitolo con le variabili globali.

0.97 Variabili globali in Gambas

In VB è possibile, ad esempio, indirizzare una casella di testo in Form1 direttamente da Form2. Il comando per questo è: Form1! Textbox1.Text = "zb". Questo non funziona a Gambas. Non ci sono variabili globali cross-project in Gambas. Si può andare in giro creando una classe chiamata Global. Le variabili globali sono dichiarate come variabili pubbliche statiche in questa classe. Quindi puoi accedervi con Global.variablename nell'intero progetto.

Minsini: È ancora una cattiva pratica di programmazione, ma almeno verranno identificate come variabili globali ogni volta che le usi;)

0.97.1 Esempio di programma:

Hai 2 moduli e desideri scambiare testo tra i moduli. La variabile globale per lo scambio di testo dovrebbe essere chiamata semplicemente t.

- In Form1 hai 3 pulsanti di comando e una casella di testo.
- In Form2 hai 2 pulsanti di comando e una casella di testo

Il codice per la classe globale (che è anche la classe iniziale) è:

```
PUBBLICO STATICO t AS String  
STATICO PUBBLICO SUB Principale ()  
form1.Show  
'Se non inserisci questa riga, non verrà visualizzato alcun modulo.  
FINE
```

Il codice per Form1 è:

```
Pulsante PUBLIC SUB1_Click ()  
Textbox1.Text = "Questo è un test"  
Global.t = Textbox1.Text  
FINE  
PUBLIC SUB Button2_Click ()  
Global.t = Textbox1.Text  
Form2.Show  
FINE  
Pulsante PUBLIC SUB 3_Click ()  
Textbox1.Text = Global.t  
FINE
```

Il codice per Form2 è:

```
Pulsante PUBLIC SUB1_Click ()  
Textbox1.Text = Global.t  
FINE  
PUBLIC SUB Button2_Click ()  
Global.t = Textbox1.Text  
Form1.Show  
'Form2.Close  
FINE
```

Se ora modifichi il testo della casella di testo in Form1, verrà trasferito in Form2 e inserito facendo clic su Button1. La stessa cosa accade in Form2. Anche qui è possibile modificare il testo della casella di testo e il testo modificato viene visualizzato nella casella di testo facendo clic su Button3 in Form1.

0.97.2 Programma di esempio semplificato per lo scambio di dati tra i moduli

Il tutto è ancora più semplice come mostra il seguente esempio:

- Hai bisogno di un Global.class che sta iniziando la classe.
- In Form1 sono necessari un'area di testo e un pulsante di comando.
- In Form2 hai bisogno di un'area di testo e di un pulsante di comando.

Codice per Global.class

```
PUBBLICO STATICO t AS String
STATICO PUBBLICO SUB Principale ()
form1.show
FINE
```

Per il nome Global.class puoi anche lasciare il nome predefinito Class1.class, nel qual caso la variabile globale non si chiama più Global.t ma Class1.t.

Codice per Form1

```
PUBLIC SUB Form_Open ()
SE NON Global.t ALLORA
'Se Global.t è ancora vuoto, riempire l'area di testo con un testo

textarea1.text = "Questo è un inizio"
ALTRO
textarea1.Text = global.t
FINISCI SE
FINE
Pulsante PUBLIC SUB1_Click ()
Global.t = textarea1.Text
Form1.Close
Form2.show
FINE
```

Codice per Form2

```
PUBLIC SUB Form_Open ()
Textarea1.Text = global.t
Textarea1.Text = Textarea1.Text & "e qualcosa al riguardo." & Chr (10)
FINE
Pulsante PUBLIC SUB1_Click ()
Global.t = textarea1.Text
form2.close
form1.Show
FINE
```

Ogni volta che si passa da un modulo all'altro, il testo dell'area di testo viene assegnato alla variabile globale te qualcosa viene aggiunto al modulo successivo.

0.98 Copiare il layout del modulo

Si desidera copiare un layout di modulo esteso esattamente da un modulo a un altro. Se selezioni tutti gli elementi del modulo e desideri trasferirli sull'altro modulo con il tasto destro del mouse utilizzando copia e incolla, gli elementi (ad esempio i pulsanti) spesso non sono più al posto giusto.

Con un semplice trucco, tutti gli elementi sono a posto correttamente anche dopo la copia.

Come misura temporanea, metti anche un elemento di controllo esattamente nell'angolo in alto a sinistra (posizione 0,0), qui ad esempio Button3. Questo funge da blocco contro Gambas, gli elementi copiati quando

Inserisci mosse. Ciò consente di risparmiare lavoro di riposizionamento, soprattutto con layout più complessi con molti elementi a volte sovrapposti. Questo piccolo trucco funziona anche con molti altri programmi di grafica.

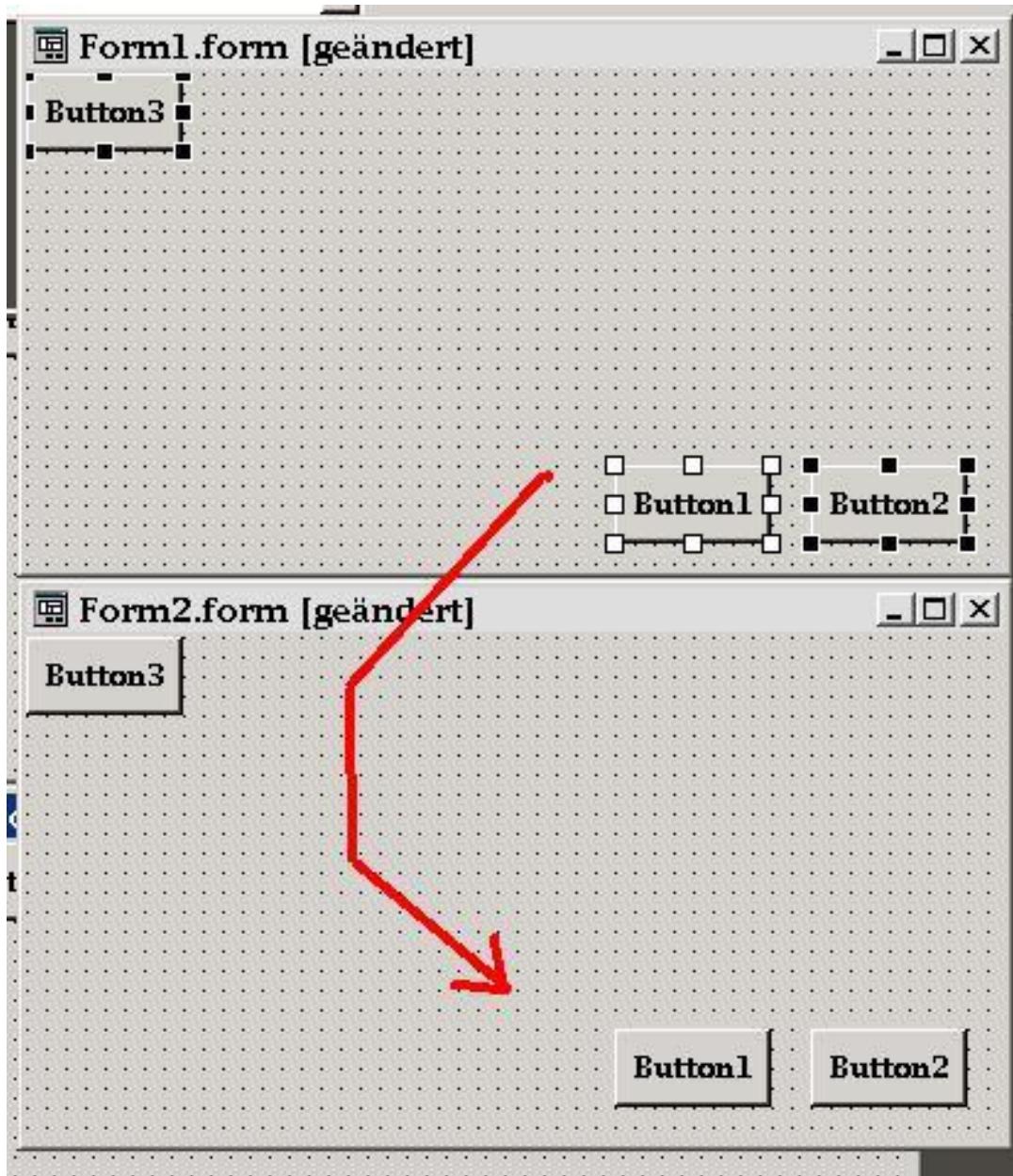


Fig.35

Per favore, riferisci <http://www.madeeasy.de/7/prgformcopy.htm>

0.99 Ridimensiona un'area di testo in una forma

Hai un modulo con un'area di testo. Vuoi sempre adattare la dimensione di quest'area di testo alla dimensione del tuo modulo.

0.99.1 Il programma

Per avviare il programma, sono necessari un modulo e un'area di testo.

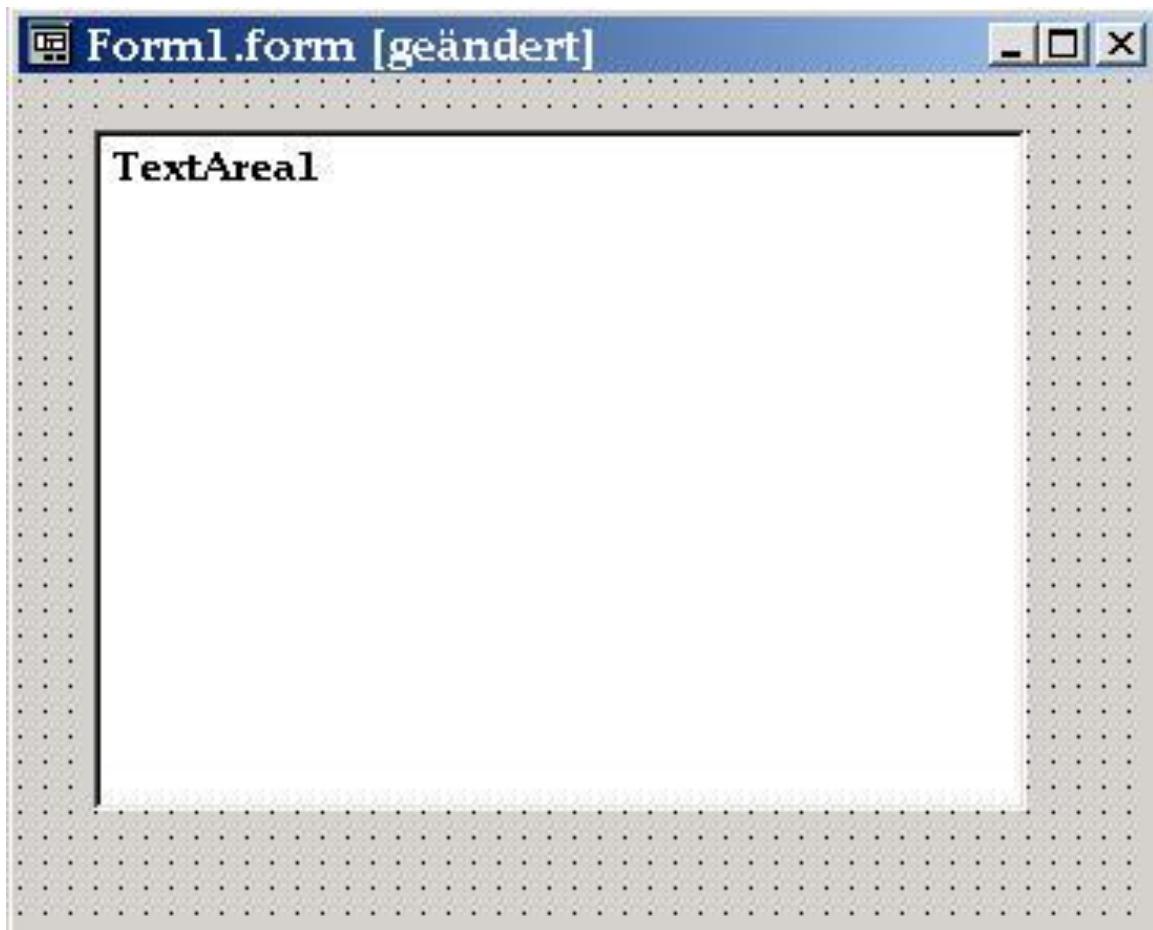


Fig.36

Quando sposti il modulo, l'area di testo si sposta sempre con esso.

Se si imposta la proprietà del bordo su Ridimensionabile nelle proprietà del modulo, l'area di testo viene adattata automaticamente a una modifica delle dimensioni del modulo.

Ecco come appare il programma in modalità di esecuzione:

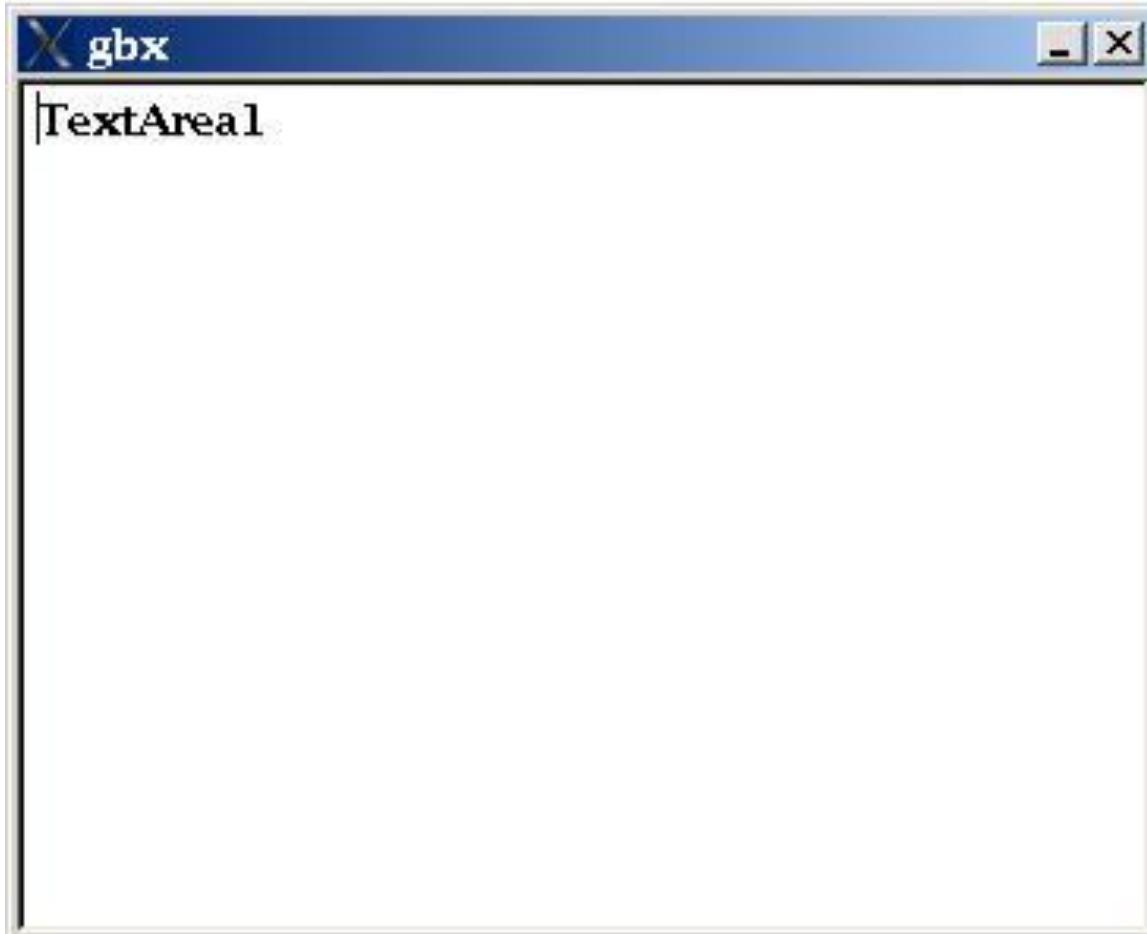


Fig.37

0.99.2 Il codice:

```
PUBLIC SUB Form_Resize ()  
    TextArea1.Move (0, 0, ME.ClientWidth, ME.ClientHeight)  
    FINE
```

Ciò che è stato mostrato qui per un'area di testo, ovviamente, funziona anche con una finestra delle immagini o un'area di disegno.

Per favore, riferisci <http://www.madeeasy.de/7/prgresize.htm>

0.100 Modifica della larghezza di un campo di testo quando si modifica la larghezza del modulo

```
PUBLIC SUB Form_Resize ()  
    TextBox1.Move (TextBox1.X, TextBox1.Y, Form1.ClientW - (TextBox1.X
```

```
* 2), TextBox1.H)
FINE
```

Quando si modifica la larghezza del modulo, la larghezza della casella di testo viene regolata in modo che la distanza dal bordo a destra sia uguale a quella a sinistra.

0.101 Generale

Grid significa griglia in tedesco. Una griglia può essere definita anche meglio una tabella. Le tabelle sono molto adatte per presentare i database in modo chiaro.

TableView è correlato a GridView⁶¹, la vista tabella, poiché viene utilizzata, ad esempio, nel programma di esempio fornito DataReportExample.

Con una griglia puoi definire il numero di colonne, il numero di righe, la larghezza e l'altezza e molto altro ancora. I singoli campi della griglia vengono indirizzati con un array, cioè con un valore per le coordinate di colonna e riga.

Puoi riempire i punti della griglia con testo, numeri, ma anche con un'immagine.

0.102 esempio di programma

Per avviare il programma, è necessario un nuovo modulo. Su questo viene posizionato un GridView dalla casella degli strumenti. Il file immagine x.png dovrebbe essere nella directory del programma, altrimenti l'immagine non verrà visualizzata. Non ci sono messaggi di errore.

```
STATICO PUBBLICO SUB Principale ()
    hForm AS Fmain
    hForm = NUOVO Fmain
    hForm.show
FINE

PUBLIC SUB _new ()
    GridView1.Columns.Count = 4
    GridView1.Rows.Count = 3
    GridView1.Columns.Width = 52
    GridView1.Rows [1] .Height = 52
    GridView1 [0,0] .Text = "0,0"
    GridView1 [0,0] .Alignment = 4
    GridView1 [1,1] .Text = "1,1"
    GridView1 [0,1] .Text = "0,1"
    GridView1 [1,0] .Picture = Picture ["x.png"]
FINE
```

0.103 Teoria di GridView

GridView fornisce un elemento di controllo che visualizza i dati in una griglia.
La classe può essere presa dalla cassetta degli attrezzi. Ma può anche essere generato dal codice.

```
g AS GridView  
PUBLIC SUB _new ()  
    g = NEW GridView (ME) COME "GridView1"  
    g.show  
    g.Columns.Count = 4  
    g.Rows.Count = 3  
    g.Columns.Width = 52  
    g.Rows [1].Altezza = 52  
FINE
```

È necessario un modulo vuoto per avviare il programma di cui sopra. Crea un nuovo GridView.

0.103.1 Proprietà di GridView

BackColor, Background, Border, ClientH, ClientHeight, ClientW, ClientWidth, Column, Columns, Current, Cursor, Design, Drop, Enabled, Expand, Font, ForeColor, Primo piano, Griglia, H, Maniglia, Altezza, Id, Sinistra, Mouse, Principale, Riga, Righe, ScreenX, ScreenY, ScrollBar, Tag, ToolTip, Top, Visible, W, Width, Window, X, Y

0.103.2 metodi

Cancella, Elimina, Trascina, Afferra, Nascondi, Abbassa, Sposta, Alza, Aggiorna, Ridimensiona, Imposta la messa a fuoco, Mostra

0.103.3 eventi

Attiva, Fai clic, Doppio clic, Trascina, DragMove, Rilascia, Invio, GotFocus, KeyPress, KeyRelease, Esci, LostFocus, Menu, MouseDown, MouseMove, MouseUp, MouseWheel, Scroll

0.104 Riempি GridView con i valori

Hai creato un elenco di valori che desideri includere nella griglia. Il seguente programma di esempio mostra come eseguire questa operazione:

Per fare ciò, hai bisogno di una nuova forma con i seguenti elementi:

- 1 TextArea
- 1 GridView
- 2 pulsanti di comando

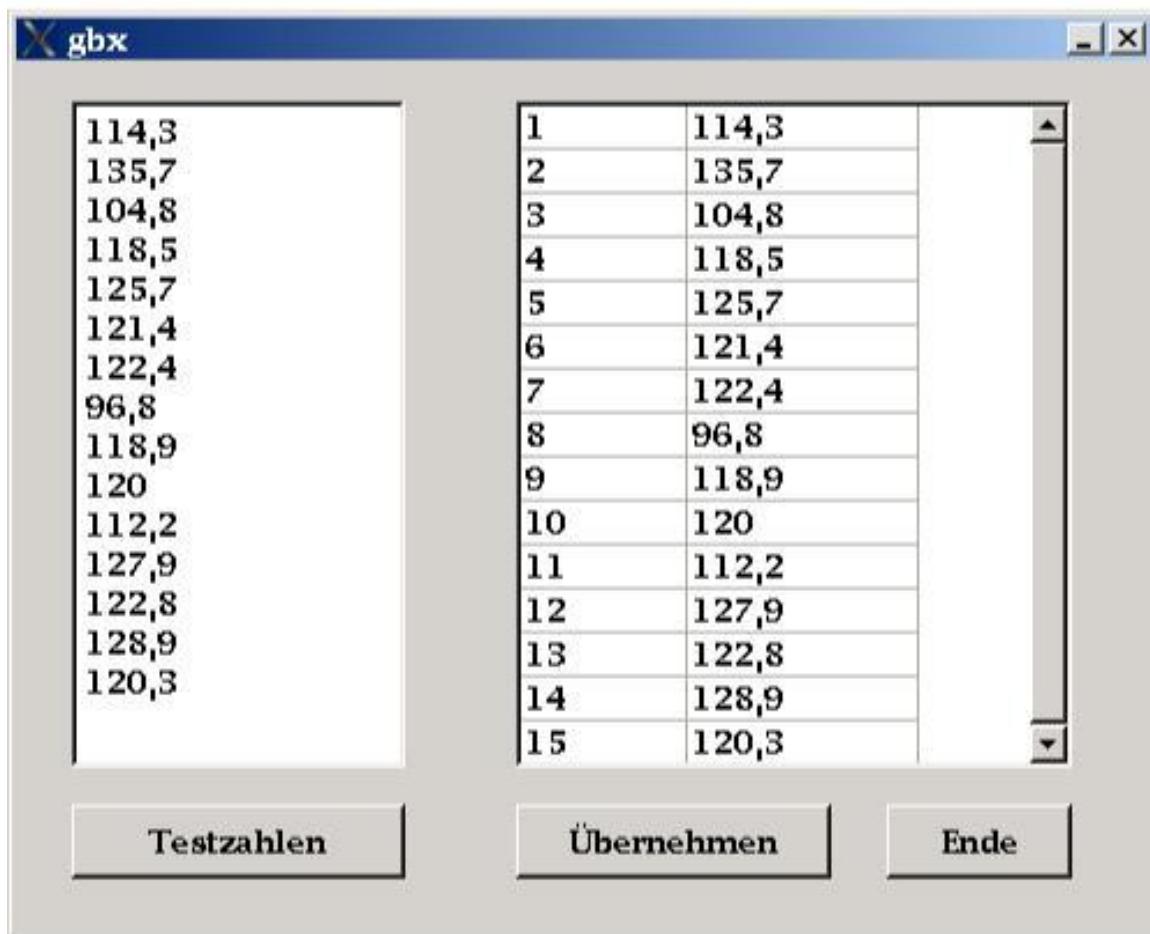


Fig.38 Riempি la griglia con i valori

Il codice per questo:

```

Pulsante PUBLIC SUB1_Click ()
    TextArea1.Text = "114.3 \ n135.7 \ n104.8 \ n118.5 \ n125.7 \ n"
    TextArea1.Text = TextArea1.Text & "121.4 \ n122.4 \ n96.8 \ n118, 9 \
n120 \ n "
    TextArea1.Text = TextArea1.Text &
"112.2 \ n127.9 \ n122.8 \ n" 128.9 \ n120.3 "
FINE

PUBLIC SUB _new ()
    GridView1.Columns.Count = 2
    GridView1.Rows.Count = 15
    GridView1.Columns.Width = 72
FINE

PUBLIC SUB Button2_Click ()
    Testo DIM AS String
    Elenco DIM come stringa []
    DIM valore singolo come stringa
    DIM x AS intero

    x = 0
    text = TextArea1.Text
    List = Split (text, Chr (10))

```

```
PER OGNI singolo valore IN lista
    GridView1 [x, 1] .Text = valore singolo
    GridView1 [x, 0] .Text = x
    x = x + 1
IL PROSSIMO

PRINT liste.Length
FINE
```

Se si preme il pulsante 1, TextArea viene riempita di valori. Se si preme il pulsante 2, questi valori vengono trasferiti alla griglia.

Come si può migliorare il programma?

1. La lunghezza della griglia è fissata con un valore di 15. Potete modificare la lunghezza in modo che sia variabile e corrisponda alla lunghezza della lista (List.Length) in TextArea?

0.105 Imposta il colore di sfondo per qualsiasi cella

A volte vuoi cambiare il colore di sfondo di una o più celle in qualsiasi colore, forse per enfatizzare il contenuto di certe celle. In realtà, questo non è possibile, perché non è previsto nelle proprietà del GridView. Ma c'è una piccola "soluzione alternativa" che descriverò di seguito.

1. ovviamente hai bisogno di un GridView. Chiamiamolo GridViewTest, ad esempio.
2. ora una variabile (es. immagine) si definisce come immagine.
3. quindi viene creato un TextBox (ad esempio TextBoxTemp) nella dimensione di una cella di GridViev.
4. si imposta la proprietà TextBoxTemp.Visible = False (TextBoxTemp è invisibile).
5. nel passaggio successivo si definisce il colore di sfondo del TextBox.
6. il contenuto della cella viene copiato nel TextBox il cui colore di sfondo deve essere modificato.
7. quindi imposta la proprietà di TextBoxTemp.Visible su True (rendilo visibile).
8. ora crea un'immagine con immagine = TextBoxTemp.Grab (),
9. imposta TextBoxTemp.Visible su False e
10. copia l'immagine nella cella GridView.

Se non funziona, un WAIT deve essere messo nel posto giusto. Le parti essenziali qui di nuovo come codice di programma.

```
Immagine DIM COME Immagine
immagine = NUOVA immagine (<Larghezza della casella di testo>, <Altezza della casella di testo>, VERO)
TextBoxTemp.Visible = TRUE
TextBoxTemp.Text = GridViewTest [<Row>, <Column>] .Text
WAIT

immagine = TextBoxTemp.Grab ()
TextBoxTemp.Visible = FALSE
GridViewTest [<Row>, <Column>] .Picture = picture
```

È importante che WAIT sia inserito esattamente in questo modo e prima della riga Grab. WAIT image = TextBox-Temp.Grab () NON funziona.

Forse questo non è programmato "in modo pulito", ma non ho ancora trovato nessun altro modo per risolvere il problema.

0.106 Componente TableView da aggiungere

Devi prendere la tabella dalla cassetta degli attrezzi QT, poiché non può essere trovata nella normale cassetta degli attrezzi di Gambas.

A tale scopo, caricare il componente gb.qt.ext. Puoi farlo tramite la voce di menu Progetto -> Proprietà della finestra del progetto Gambas. Anche il componente gb.qt.ext viene selezionato nella visualizzazione del componente.

Vedi anche: Gambas: componenti⁶².

Un pulsante etichettato QT viene quindi visualizzato nella casella degli strumenti. Fai clic su questo e quindi hai una serie di altri controlli tra cui la vista tabella (TableView) disponibile.

0.107 Usa TableView

Il seguente programma utilizza il controllo TableView da gb.qt.ext. I numeri di 1 * 1 vengono inseriti in una tabella con 10 colonne e 10 righe. Hai bisogno di una forma e del controllo TableView dalla casella degli strumenti per farlo partire.

```

PRIVATO aro COME NUOVO Oggetto []
PUBLIC SUB _New ()
    Variante DIM ari AS []
    DIM i AS intero
    DIM j AS Numero intero

    'Viene generato un array 10 X 10
    PER i = da 0 a 9
        ari = NUOVA variante []
        ari.Resize (10)
        aro.Add (ari)
    IL PROSSIMO

    'i valori da 1 a 100 sono assegnati agli elementi dell'array FOR i = 0 TO
    aro.Count - 1
        FOR j = 0 TO aro [i].Count - 1
            aro [i] [j] = (i + 1) * (j + 1)
        IL PROSSIMO
    IL PROSSIMO

    TableView1.Rows.Count = 10
    TableView1.Columns.Count = 10
FINE

PUBLIC SUB TableView1_Data (Row AS Integer, Column AS Integer)
    TableView1.Data.Text = aro [Riga] [Colonna]
    IF aro [Row] [Column] = 50 THEN TableView1.Data.BackColor =
        Color.Gray

```

62 Capitolo 12.18 a pagina 424

FINE

Modifica 0,108 celle

Il controllo TableView è responsabile solo della visualizzazione dei valori, non è possibile modificare i valori nelle celle. Se vuoi modificare i valori direttamente nelle celle, come lo sai dai fogli di calcolo, devi posizionare dinamicamente un TextBox sopra la cella cliccata, in cui il contenuto della cella può quindi essere modificato. Ma non preoccuparti, sembra più complicato di quello che è.

Avvia un nuovo progetto, crea una nuova forma e posiziona un TableView, un TextBox e un pulsante su questa forma. Quindi è necessario il seguente codice sorgente:
TableView1 Form1.class⁶³

0.109 celle colorate

L'esempio sopra può essere espanso con uno sforzo minimo in modo che diversi colori di sfondo possano essere assegnati alle singole celle. Non è possibile accedere direttamente alle celle di TableView, ma i valori dei colori devono essere mantenuti in una matrice corrispondente (nell'esempio seguente \$ hColor) e resi disponibili a TableView all'interno della routine TableView1_Data (). Per fare ciò, cambiamo il codice sorgente dell'esempio precedente come segue: TableView2 Form1.class⁶⁴

0.110 Navigazione con i tasti cursore

Quello che ci manca ora è la possibilità di cambiare la cella selezionata per la modifica con i tasti cursore. Per far sì che ciò accada, dobbiamo solo fare due cose. Innanzitutto, cambieremo la funzione TextBox1_KeyPress () per rispondere ai tasti appropriati. In secondo luogo, chiamiamo il metodo TableView1.MoveTo () all'inizio della funzione TextBox1Show () per assicurarci che la cella che vogliamo modificare si trovi anche nell'area visibile di TableView. Il codice sorgente corrispondente: TableView3 Form1.class⁶⁵

Inoltre, ora ci asteniamo dall'impostare il colore di sfondo del TextBox su giallo e utilizziamo invece il rispettivo colore di sfondo dall'array \$ hColor.

⁶³ <http://de.wikibooks.org/wiki/Gambas%3A%20Tabelle%3A%20TableView1%20Form1>.

⁶⁴ classe
<http://de.wikibooks.org/wiki/Gambas%3A%20Tabelle%3A%20TableView2%20Form1>.

⁶⁵ classe
<http://de.wikibooks.org/wiki/Gambas%3A%20Tabelle%3A%20TableView3%20Form1>.

Tabella Wiki 0.111

Dopo tutte le prove a secco, ora vogliamo dedicarci a un esempio almeno a metà sensato dell'uso di un TableView.

Ogni ambizioso autore di Wikibook o Wikipedia prima o poi si troverà nell'imbarazzo di voler visualizzare determinate informazioni in una tabella. Per creare la seguente tabella, ad esempio:

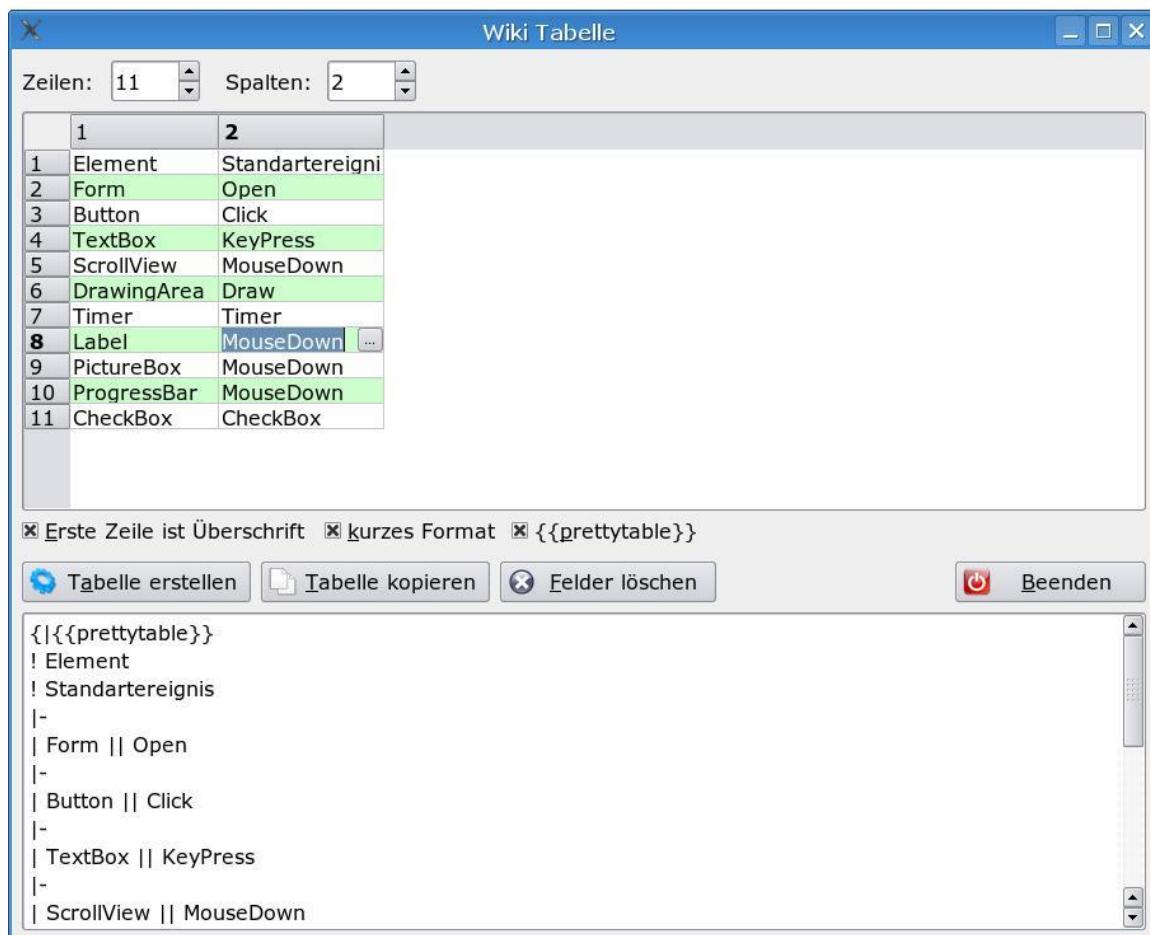
Valore1	Valore2	Valore3
Valore4	Valore5	Valore6
Valore7	Wert8	Wert9

hai bisogno del seguente codice sorgente wiki:

```
{| {{prettytable}}
| Value1 || Value2 || Valore3
|- 
| Value4 || Value5 || Valore6
|- 
| Value7 || Value8 || Wert9
|}
```

Questo diventa rapidamente fonte di confusione con le tabelle più grandi.

Quindi abbiamo bisogno di un programma con il quale possiamo comodamente inserire i dati in una tabella e che ci fornisce il codice sorgente wiki necessario con la semplice pressione di un pulsante.



The screenshot shows the 'Wiki Tabelle' dialog in Gambas. At the top, there are two dropdown menus: 'Zeilen:' set to 11 and 'Spalten:' set to 2. The main area contains a table with 11 rows and 2 columns:

	1	2
1	Element	Standartereignis
2	Form	Open
3	Button	Click
4	TextBox	KeyPress
5	ScrollView	MouseDown
6	DrawingArea	Draw
7	Timer	Timer
8	Label	MouseDown [...]
9	PictureBox	MouseDown
10	ProgressBar	MouseDown
11	CheckBox	CheckBox

Below the table are several checkboxes: 'Erste Zeile ist Überschrift' (checked), 'kurzes Format' (checked), and '{{prettytable}}' (checked). There are also three buttons: 'Tabelle erstellen' (Create Table), 'Tabelle kopieren' (Copy Table), and 'Felder löschen' (Delete Fields). A 'Beenden' (End) button is on the right.

The bottom pane shows the generated source code:

```

{{{{prettytable}}}
! Element
! Standartereignis
|-
| Form || Open
|-
| Button || Click
|-
| TextBox || KeyPress
|-
| ScrollView || MouseDown

```

Fig.39 Tabella Wiki

Codice sorgente:

- Form1.class⁶⁶
- Form1.form⁶⁷
- Form2.class⁶⁸
- Form2.form⁶⁹

Creare un nuovo progetto chiamato Wiki-Table, chiudere di nuovo Gambas immediatamente e copiare il testo sorgente dalle quattro pagine precedenti in (nuovi) file con i nomi appropriati (Form1.class, ...) nella directory del progetto selezionata. Quindi riavvia Gambas e apri il progetto e definisci Form1 come classe iniziale nella finestra del progetto.

Nella cassetta degli attrezzi Gambas è presente un elemento che abilita la visualizzazione di un elenco.

L'esempio seguente mostra come funziona questo elemento:

66 <http://de.wikibooks.org/wiki/Gambas%3A%20Tabelle%3A%20Wiki-Tabelle%20Form1>.
 classe
<http://de.wikibooks.org/wiki/Gambas%3A%20Tabelle%3A%20Wiki-Tabelle%20Form1>.

67 forma
<http://de.wikibooks.org/wiki/Gambas%3A%20Tabelle%3A%20Wiki-Tabelle%20Form2>.

68 classe
<http://de.wikibooks.org/wiki/Gambas%3A%20Tabelle%3A%20Wiki-Tabelle%20Form2>.

69 forma

Hai bisogno di un pulsante di comando e di un elemento di visualizzazione elenco.
Ottieni entrambi dalla casella degli strumenti con F6.

```
Pulsante PUBLIC SUB1_Click ()  
    Listview1.add ("1", "Name1")  
    Listview1.add ("2", "Name2")  
    Listview1.Add ("3", "Name3")  
FINE
```

Se fai clic sul pulsante di comando una seconda volta, verrà visualizzato un messaggio di errore.
Perché?

Nel secondo esempio, le voci del file della sottodirectory corrente vengono inserite nell'elenco.

Hai bisogno di un pulsante di comando e di un elemento di visualizzazione elenco.
Ottieni entrambi dalla casella degli strumenti con F6.

```
Pulsante PUBLIC SUB1_Click ()  
    sDir AS String  
    sPath AS String  
    PER OGNI sDir IN TE (sPath, "*")  
        ListView1.Add (sPath & / sDir, sDir)  
    IL PROSSIMO  
FINE
```

0,112 Messaggio semplice

Un messaggio è molto facile da programmare. Immettere il codice seguente in un modulo vuoto e avviare il programma.

```
PUBLIC SUB Form_Open ()  
    Messaggio ("Questo è un messaggio", "ButtonText")  
FINE
```

In alternativa, puoi utilizzare il seguente codice:

```
PUBLIC SUB Form_Open ()  
    Message.Info ("Ciao, questo è il messaggio informativo", "OK")  
FINE
```

0.113 Messaggio con diverse possibili risposte

Oltre al semplice Message.Info, ci sono altri quattro tipi di messaggi:

- Message.Delete (elimina)
- Message.Error (messaggio di errore)
- Message.Question (domanda)
- Message.Warning

Sono possibili fino a tre risposte per queste finestre di messaggio. C'è anche un'icona Schermo:

- Delete = "bidone della spazzatura",
- Errore = "Punto rosso con X",
- Domanda = "?" Punto interrogativo,
- Avviso = "Triangolo giallo con punto esclamativo"

0.113.1 esempio

L'esempio seguente mostra come funzionano queste finestre di messaggio. Hai bisogno di un pulsante e di un controllo dell'etichetta sul modulo per avviare il programma. Puoi trovarli entrambi nella casella degli strumenti con F6.

0.113.2 Layout

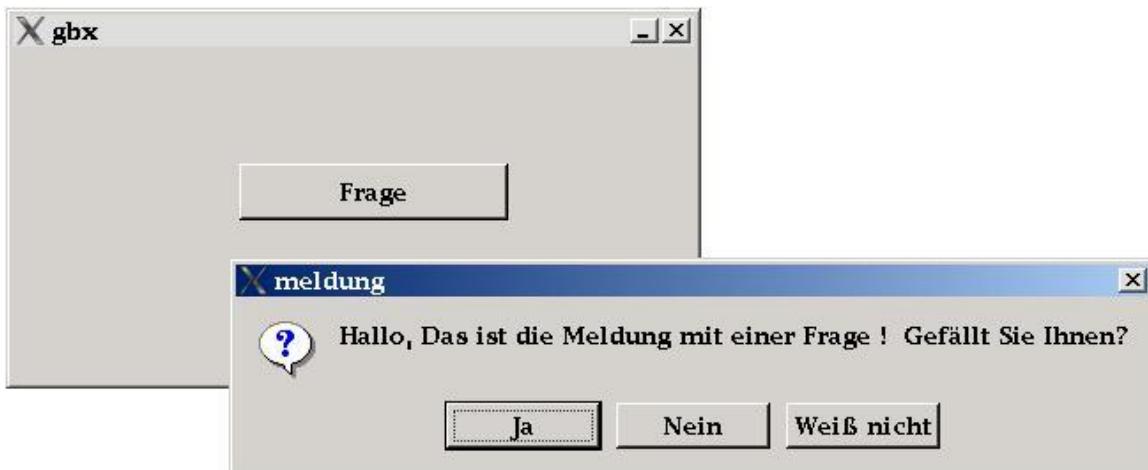


Fig.40

0.113.3 codice

```
Pulsante PUBLIC SUB1_Click ()  
    SELECT Message.Question ("Ciao, questo è il messaggio con estensione  
    Domanda! Ti piace? "," Sì "," No "," Non so ")  
        CASO 1  
            Label1.Text = "Piacere!"  
        CASO 2  
            Label1.Text = "Deplorevole!"  
        CASO 3  
            Label1.Text = "Ti piaceranno presto!"  
    FINE SELEZIONA  
FINE
```

0.114 Introduzione

Se sei un po' interessato alla matematica, un linguaggio di programmazione come Gambas può essere molto utile per capire ancora meglio la matematica.

Al contrario, puoi imparare molto su un linguaggio di programmazione programmando piccoli programmi matematici. Il vantaggio dei programmi matematici è che di solito sono facili da capire e che gli errori possono essere eliminati in breve tempo.

Inoltre, all'inizio puoi fare a meno della grafica e lavorare solo con la finestra diretta, anche se la grafica è ciò che rende i programmi KDE così attraenti.

0.115 espressioni aritmetiche

Per calcolare correttamente le espressioni aritmetiche lunghe in Gambas, è necessario conoscere alcune convenzioni per i simboli matematici che sono valide in quasi tutti i dialetti di base. Per la moltiplicazione, la divisione e l'esponenziazione vengono utilizzati caratteri diversi da quelli usuali in tedesco.

- L'addizione (conteggio in avanti) viene eseguita con +
• ad es. Stampa $8 + 8$
- La sottrazione (detrazione) viene eseguita con:
• ad esempio Stampa $9 - 5$
- La moltiplicazione (prendendo i tempi) si fa con l'asterisco *
• ad es. Stampa $5 * 5$
- La divisione (divide) viene eseguita con la barra /
• ad es. stampa $10/5$
• risultati in 2
• Fai attenzione alla divisione con o senza resto
- L'esponenziazione (esponenti) viene calcolata utilizzando il simbolo ^.
• ad es. Stampa $8 ^ 2$
• risultati in 64
- Le parentesi determinano la priorità nel calcolo di $5 * (5 + 5)$
• Sbagliato è $5 (5 + 5)$, corretto è $5 * (5 + 5)$

Le seguenti regole di priorità si applicano al calcolo:

- Per prima cosa vengono calcolate le parentesi. Prima le parentesi più interne.
- Il seguente ordine di precedenza si applica alle operazioni: prima esponenziale, quindi moltiplica o dividi, quindi somma o sottrai.
- Le espressioni di uguale rango vengono elaborate da sinistra a destra.

C'è un solo simbolo aritmetico per la radice quadrata SQR in Gambas. Le radici superiori devono essere calcolate in modo diverso: qui, la radice x-esima di a è uguale a $a^{(1/x)}$ (a alla potenza da 1 a x) Ad esempio, la terza radice di 27 è calcolata come segue: $27^{(1/3)} = 3$

Invece di

$\frac{36}{236}$

è scritto in Gambas: $36/2/3/6$ o $36 / (2 * 3 * 6) = 1$

Si noti che il segno meno viene utilizzato una volta come segno e una volta come segno di sottrazione. Sono valide le seguenti fatture:

```
5 + -2 = 3
-5 * 3 = -15
5*-3 = -15
5/4 = 1,25 o 1,25 a seconda dell'impostazione del paese
20/2*5 = 50
20/2/5 = 2
20 / (2*5) = 2
5*(2 + 3) = 25
2*(2 + 3) / (4 + 6) = 1
((2 + 3)*2) ^2 = 100
```

0.115.1 Addy il totalizzatore

Se vuoi sommare una colonna di numeri, perdi rapidamente il numero di quale numero hai già inserito e quale no. Ovviamente puoi usare un grande foglio di calcolo per farlo. Tuttavia, questo è un po 'complicato da usare.

Addy, un piccolo programma di sommatoria, dovrebbe svolgere questo compito più facilmente. Dovrebbe essere possibile anche una successiva correzione della voce.

I numeri devono essere inseriti rapidamente, preferibilmente con un ritorno senza l'azione del mouse.

Il programma può anche totalizzare numeri negativi. Verranno quindi detratti correttamente. È anche possibile aggiungere numeri decimali.

Inoltre, è possibile trasferire colonne di numeri tramite il buffer.

Il comando split è molto utile durante la creazione di questo programma. In questo caso, CHR (10) come separatore è l'interruzione di riga.

Di che cosa hai bisogno? Ottieni una nuova forma e chiamala come desideri. Ottieni un TextArea, un TextBox e un Commandbutton dalla casella degli strumenti. Elimina il testo dalle caselle di testo sotto Proprietà Testo.

Ecco come appare:

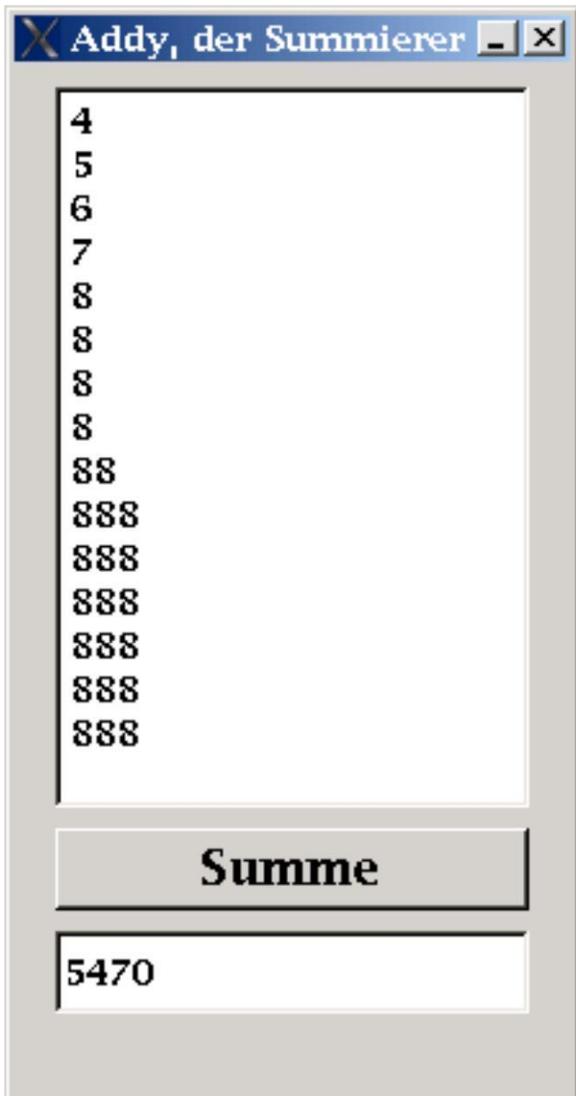


Fig.41

Il codice del programma per esso. Si trova dietro la somma del pulsante di comando:

```
Pulsante PUBLIC SUB1_Click ()
Testo DIM AS String
Somma DIM AS Float
DIM Elt AS String []
DIM Sb AS String

text = TextArea1.Text
Elt = Split (testo, Chr (10))
PER CIASCUNO Sb IN Elt
    somma = somma + Val (sb)
IL PROSSIMO
TextBox1.Text = sum
FINE
```

Il programma è molto pratico, o cosa ne pensi? Nel caso di numeri decimali, i risultati vengono comunque emessi in modo errato con il punto inglese come virgola. Prova a correggere questo errore.

Nel programma viene visualizzato un messaggio di errore Type mismatch: Wanted Number, ottenuto invece Null se provi ad inserire lettere invece di numeri, ad esempio.

Il computer può fare qualcosa solo con i numeri qui, non con le lettere. È possibile evitare l'errore limitando l'immissione a cifre, virgolette e segni meno. È anche possibile rilevare l'errore in modo sicuro con Catch.

Vuoi salvare le voci in Addy. Quindi dai un'occhiata alla seguente estensione:
Gambas: Files # Carica e salva i file con Addyplus⁷⁰

Il programma Addy ti incoraggia a pensare a un piccolo calcolatore di media e deviazione standard.

Potenze 0.115.2

Il seguente programma di esempio mostra come utilizzare i poteri. Mostra anche i possibili errori. Funziona in modalità terminale.

```
STATICO PUBBLICO SUB Principale ()  
STAMPA 4^3  
STAMPA 3^4  
STAMPA 4-1 / 2  
STAMPA 4^ (1/2)  
STAMPA 8^0.333  
STAMPA 8^- (1/3)  
STAMPA 8-1 / 3  
STAMPA 2^ (-2)  
STAMPA 1 / 2^2  
FINE
```

L'output è simile a questo:

```
64  
81  
2  
2  
1.998614185981  
2  
2.666666666667  
0.25  
0.25
```

Domanda: a cosa serve qui il comando Statico? È possibile senza di lui?

Il comando Statico viene assegnato automaticamente dalla GUI in modalità terminale. Se lo elimini, il programma non funziona più e viene visualizzato un messaggio di errore: ERRORE: # 24: Nessun metodo di avvio

⁷⁰ Capitolo 6.61 a pagina 268

0.115.3 Modulo o modulo (resto quando si divide)

Il comando MOD corrisponde - spiegato semplicemente - al resto della condivisione. Se la divisione funziona, non c'è resto (cioè 0).

Alcuni esempi sono utili per la comprensione (soprattutto per i principianti):

- 10 MOD 5 è uguale a 0 (errore comune: non viene fuori 2. MOD non è uguale a /)
- 2 MOD 3 dà 2
- 6 MOD 4 dà anche 2
- 8 MOD 3 dà 2

Un altro esempio mostra il seguente programma nel funzionamento da terminale:⁷¹

```
STATICO PUBBLICO SUB Principale ()  
DIM un numero intero AS  
DIM b AS Interio  
  
b = 2  
PER a = 10 TO 2 STEP - 1  
    STAMPA a MOD b  
    IL PROSSIMO  
FINE
```

Sostituisci valori diversi per b e guarda l'output. Se si imposta b = 0, viene visualizzato un messaggio di errore: Divisione per zero.

Modalità tutorial

Per favore, riferisci <http://www.madeeasy.de/7/prgmodus.htm>

Calcola il checksum

Nel seguente programma, la modalità Comando viene utilizzata per calcolare il checksum di un numero a più cifre: Ottieni una nuova forma e chiamala come preferisci. Ottieni due caselle di testo dalla casella degli strumenti.

Elimina il testo dalle caselle di testo sotto Proprietà Testo.

Ottieni un pulsante di comando e rinominalo.

Per favore, riferisci <http://www.madeeasy.de/7/prgquer.htm>

Il codice seguente si trova dietro il campo del comando checksum: Notare che il simbolo di divisione / non viene utilizzato qui, ma il simbolo parziale senza resto \.

```
Pulsante PUBLIC SUB1_Click ()  
    Numero DIM AS Integer  
    Somma di controllo DIM AS intero
```

71 Capitolo 12.2.1 a pagina 419

Contenuti

```
numero = Val (TextBox1.Text)
DO WHILE numero <> 0
    Checksum = checksum + numero MOD 10
    numero = numero \ 10
CICLO CONTINUO
TextBox2.Text = Str $ (checksum)
FINE
```

Pari o dispari

Il comando modulo può essere utilizzato per verificare se un intero è pari (divisibile per 2) o dispari (non divisibile per 2).

Programma: sono necessari 2 pulsanti di comando e 2 caselle di testo sul modulo per avviare il programma.

Utilizzare il codice seguente per impostare automaticamente le proprietà dei controlli:

```
PUBLIC SUB Form_Open ()
    Button1.Text = "Genera numero casuale"
    Button1.Move (16, 8, 176, 32)
    Button2.Text = "Test pari-dispari"
    Button2.Move (16, 48, 176, 32)
    Textbox1.Text = ""
    Textbox1.Move (192, 8, 88, 32)
    Textbox2.Text = ""
    Textbox2.Move (192, 48, 88, 32)
FINE
```

Ovviamente puoi anche impostare le proprietà manualmente usando la finestra delle proprietà.

Il vero codice inizia qui:

```
Pulsante PUBLIC SUB1_Click ()
    Rendi casuale ()
    textbox1.Text = Str (Int (Rnd (1,101)))
FINE

PUBLIC SUB Button2_Click ()
    DIM n AS intero

    n = Val (textbox1.text)

    IF textbox1.Text = "" THEN textbox1.Text = Str (Int (Rnd (1,101)))

    SE n MOD 2 ALLORA
        textbox2.Text = "Dispari"
    ALTRO
        textbox2.Text = "Linea retta"
    FINISCI SE
FINE
```

Il primo comando genera un numero intero casuale compreso tra 1 e 100. Il secondo comando verifica se può essere diviso per 2.

Se n è un numero dispari, allora n MOD 2 è uguale a 1 perché è rimasto 1 resto. Il valore di verità è vero. Se non è rimasto alcun resto, il valore di verità è falso.

Puoi anche scrivere in modo più dettagliato:

```
SE n MOD 2 = vero Allora
```

Cerca di capirlo e prova il programma.

0.116 Panoramica dei simboli matematici in Gambas

vedi Segni matematici in Gambas⁷²

0.117 Rappresentazione numerica

0.117.1 Calcoli semplici con numeri interi

I numeri vengono inizialmente gestiti come al solito a Gambas. L'espressione PRINT 8 + 8 restituisce l'output 16.

```
PUBLIC SUB Form_Open ()  
    STAMPA 8 + 8  
FINE
```

Se vuoi lavorare più a lungo con i numeri nel programma, devi definirli come variabili. La forma più semplice per questo è la variabile intera. Una variabile numerica viene quindi dichiarata, ad esempio, con Dim number AS Integer.

Nel seguente semplice programma di esempio per la somma di tutte le cifre da 1 a 10, sono definite 2 variabili numeriche i e k. Entrambi fanno un ciclo da 1 a 10 e si sommano. Copia il programma in Gambas e lascialo funzionare. Sostituisci il segno più con altri simboli matematici e guarda cosa succede.

```
PUBLIC SUB Form_Open ()  
    DIM i AS intero  
    DIM k AS Integer  
  
    PER i = da 1 a 10  
        PER k = da 1 a 10  
            STAMPA i + k & "";  
            IL PROSSIMO  
            STAMPA  
            IL PROSSIMO  
    FINE
```

Con la prima riga con il comando Form_Open (), il programma si avvia immediatamente quando viene chiamato il form. I comandi DIM vengono utilizzati per definire le variabili iek come numeri interi (= numeri interi). Questo è seguito da 2 cicli FOR - TO annidati. Nel primo ciclo, la variabile

72 Capitolo 10.2 a pagina 382

ho contato da 1 a 10. Ma prima di passare a un valore più alto, il ciclo interno di k deve essere elaborato ancora e ancora da 1 a 10. Nell'output con il comando print, i ek vengono quindi collegati matematicamente e visualizzati nella finestra diretta con PRINT. La parte & "" inserisce sempre uno spazio tra i numeri calcolati i + k. Il punto e virgola alla fine della riga impedisce un'interruzione di riga. Quindi l'output viene scritto su una riga.

Solo nel ciclo esterno per i è presente un comando di stampa solitario. Ciò causa un avanzamento riga ogni volta che viene annunciato un nuovo valore per i. Il cambio di riga avviene sempre dopo 10 numeri nell'output orizzontale. Si noti che i loop vengono chiusi con il comando successivo. Il comando Next non include la specifica Next i o Next k in Gambas. Il computer lo sa da solo.

Se non capisci i 2 cicli, sostituisci il segno più con una virgola o un punto e virgola. Quindi avvia il programma. Ora puoi vedere come vengono incrementati i valori di i e k. Se ancora lo capisci, ti consiglio la modalità a passo singolo nel flusso del programma.

Prova a semplificare il programma e mantieni uno dei valori, ad esempio i, costante. Per fare ciò, aggiungi, ad esempio, una riga i = 5 dopo i comandi DIM e cancella il ciclo esterno per il valore i incluso il comando Next.

Il tuo programma dovrebbe quindi assomigliare a questo:

```
PUBLIC SUB Form_Open ()  
    DIM i AS intero  
    DIM k AS Integer  
  
    io = 5  
    PER k = da 1 a 10  
        STAMPA i + k & "",  
        IL PROSSIMO  
        STAMPA  
    FINE
```

La variabile i è effettivamente superflua qui e può anche essere eliminata. Risultato:

```
PUBLIC SUB Form_Open ()  
    DIM k AS Integer  
  
    PER k = da 1 a 10  
        STAMPA 5 + k & "",  
        IL PROSSIMO  
    FINE
```

Anche la moltiplicazione è semplice e come al solito. Tutto quello che devi fare è sostituire il segno più nel programma di addizione di cui sopra con il segno di moltiplicazione inglese, l'asterisco **.

La moltiplicazione mostra, ad esempio, un semplice programma una tantum: Vedi anche <http://www.madeeasy.de/7/prgschleife2.htm>

```
PUBLIC SUB Form_Open ()  
    DIM i AS intero  
    DIM k AS Integer  
  
    PER i = da 1 a 10  
        PER k = da 1 a 10
```

```
STAMPA i * K & "";
IL PROSSIMO
STAMPA
IL PROSSIMO
FINE
```

0.117.2 Rappresentazione esponenziale con E.

Se si esegue il seguente programma, l'output non viene visualizzato come previsto, 10 100 1000 ecc., Ma dal numero un milione in poi, i numeri grandi vengono abbreviati con 1.E + 8 e 1.E + 10. La lettera E in questo contesto significa 10 alla potenza di 10. Il numero successivo con un segno più o meno davanti è il numero con cui si aumenta il dieci, cioè la potenza.

1.E + 8 significa più precisamente 1.0×10^8 o 1.0 * 10 alla potenza di 8 o 1.0×10^8

```
PUBLIC SUB Form_Open ()
DIM i AS intero
DIM k AS Integer

io = 10
PER k = da 1 a 10
    STAMPA k ^ i & "";
    IL PROSSIMO
FINE
```

Per ripetere: la "E" maiuscola significa "Times 10 alla potenza di". Il numero intero che segue indica la potenza di dieci per la quale il numero deve essere moltiplicato. Un altro modo per vedere il numero intero che segue la "E" è che indica la direzione e il numero di posizioni in cui il punto decimale deve essere spostato. I numeri interi positivi indicano uno spostamento a destra; interi negativi indicano uno spostamento a sinistra.

0.117.3 Errore di calcolo

Per inciso, il seguente programma restituisce un errore di calcolo.

```
PUBLIC SUB Form_Open ()
    STAMPA 1000000 * 1000000
FINE
```

Ricalcola più volte tutti i tuoi programmi Gambas, poiché spesso si verificano risultati imprevisti a causa della conversione delle variabili e del diverso utilizzo del punto e della virgola in frazioni decimali in inglese e tedesco.

Esempio: tedesco $1/2 = 0,5$ inglese $1/2 = 0,5$

Tuttavia, questo problema si verifica anche nella maggior parte degli altri linguaggi di programmazione e non è specifico di Gambas.

I matematici tedeschi dovrebbero adattarsi alla notazione inglese il prima possibile e anche cambiare i simboli per la moltiplicazione, la divisione e l'elevamento a potenza nella notazione del computer.

Un elenco di paesi con i rispettivi separatori decimali può essere trovato qui:
<http://www.fit.net/Separatore decimali/>

**0.118 Conteggio in avanti, conto alla rovescia, troncatura, arrotondamento ecc. INC,
DEC, INT, FRAC**

0.118.1 INC Count up, aggiungi 1

Il comando INC aggiunge uno a ciascuna variabile numerica. INC sta per Increment, cioè contare.

Esempio in modalità terminale (cioè senza grafica)

```
STATICO PUBBLICO SUB Principale ()  
DIM X AS Integer  
X = 7  
INC X  
STAMPA X  
FINE
```

L'output avviene nella finestra diretta e si legge: 8

Secondo esempio con un array

```
STATICO PUBBLICO SUB Principale ()  
DIM A [3, 3] AS float  
DIM X AS Integer  
DIM Y AS Integer  
  
X = 2  
Y = 1  
A [X, Y] = Pi greco  
INC A [X, Y]  
STAMPA A [X, Y]  
FINE
```

Edizione: 4.14159265359

Ovviamente puoi semplicemente aggiungerne uno al posto del comando INC

```
INC X
```

Equivale a

```
X = X + 1
```

I seguenti sono sbagliati e non funzionano:

```
X = INC X
```

0.118.2 Conto alla rovescia DEC

L'opposto di INC è fatto dal comando DEC. Sottrae 1 dal numero. DEC significa decremento in inglese, cioè conto alla rovescia.

Esempio in modalità terminale (cioè senza grafica)

```
STATICO PUBBLICO SUB Principale ()
  DIM X AS Integer
  X = 7
  DEC X
  STAMPA X
FINE
```

L'output avviene nella finestra diretta e si legge: 6

```
STATICO PUBBLICO SUB Principale ()
  DIM A [3, 3] AS float
  DIM X AS Integer
  DIM Y AS Integer

  X = 2
  Y = 1
  A [X, Y] = Pi greco
  DIC A [X, Y]
  STAMPA A [X, Y]
FINE
```

Edizione: 2.14159265359

0,119 Segno e valore assoluto Sgn, Abs

0.119.1 Sgn (Signum) Segno di un numero

Il comando Sgn fornisce un valore che rappresenta il segno di un numero. Il numero può essere qualsiasi espressione numerica legale. Il suo segno determina il valore emesso dalla funzione Sgn:

- Se il numero > 0, Sgn (numero) restituisce il valore 1.
- Se il numero = 0, Sgn (numero) restituisce il valore 0.
- Se il numero è <0, Sgn (numero) restituisce il valore -1.

Esempi semplici:

Contenuti

```
PRINT Sgn (Pi)
```

Risultato: 1

```
PRINT Sgn (-Pi)
```

Risultato: -1

```
PRINT Sgn (0)
```

Risultato: 0

Esempio:

```
PUBLIC SUB Form_Open ()  
    DIM x AS intero  
    DIM y AS Float  
  
    PER x = da 1 a 100  
        y = Rnd (-5, 5)  
        Vengono generati 100 numeri casuali compresi tra -5 e +5.  
  
        SELECT CASE Sgn (y) '... assess.  
            CASO 0 'Se numero zero.  
                PRINT y & "" & Sgn (y) & "Signum = zero."  
            CASO 1 'Se il numero è positivo.  
                PRINT y & "" & Sgn (y) & "Signum = numero positivo."  
            CASO -1 'Se il numero è negativo.  
                PRINT y & "" & Sgn (y) & "Signum = numero negativo."  
        FINE SELEZIONA  
    IL PROSSIMO  
FINE
```

O qualcos'altro:

```
PUBLIC SUB Form_Open ()  
    DIM x AS intero  
    DIM y AS Float  
    DIM pos AS intero  
    DIM neg AS intero  
    DIM nul AS intero  
  
    pos = 0  
    neg = 0  
    zero = 0  
  
    PER x = da 1 a 100  
        y = Rnd (-5, 5)  
        Vengono generati 100 numeri casuali compresi tra -5 e +5.  
  
        SELECT CASE Sgn (y) '... assess.  
            CASO 0 'Se il numero è zero.  
                INC nul  
            CASO 1 'Se il numero è positivo.  
                INC pos  
            CASO -1 'Se il numero è negativo.  
                INC neg  
        FINE SELEZIONA
```

IL PROSSIMO
STAMPA zero, pos, neg
FINE

Ovviamente puoi farlo anche senza il comando INC:

```
STATICO PUBBLICO SUB Principale ()
  DIM x AS intero
  DIM y AS Float
  DIM pos AS intero
  DIM neg AS intero
  DIM nul AS intero

  pos = 0
  neg = 0
  zero = 0

  PER x = da 1 a 100
    y = Rnd (-5, 5)
    Vengono generati 100 numeri casuali compresi tra -5 e +5.

  SELECT CASE Sgn (y) '... assess.
    CASO 0 'Se il numero è zero.
      zero = zero + 1
    CASO 1 'Se il numero è positivo.
      pos = pos + 1
    CASO -1 'Se il numero è negativo.
      neg = neg + 1
  FINE SELEZIONA
  IL PROSSIMO
  STAMPA zero, pos, neg
FINE
```

Come previsto, non ci sono quasi risultati sullo zero, aumentare il numero x e vedere se lo zero viene raggiunto ad un certo punto.

0.119.2 Valore assoluto di un numero ABS

Il comando ABS restituisce il valore assoluto di un numero, ovvero il segno viene cancellato se negativo. In matematica, per questo vengono utilizzate 2 barre verticali $| -2 | = 2$.

Piccoli esempi:

Stampa abs (-2)

Risultato: 2

Stampa abs (2)

Risultato: 2

Stampa abs (0)

Risultato: 0

Guarda anche: <http://www.madeeasy.de/7/prgabs.htm>

Un esempio leggermente più lungo in modalità terminale:

```
STATICO PUBBLICO SUB Principale ()  
DIM i AS intero  
PER i = da -10 a 10  
    PRINT i & "=> Abs (i) =>" & Abs (i)  
    IL PROSSIMO  
FINE
```

0.120 Il comando Val: crea qualcosa di significativo da un testo

Val () converte una stringa (= sequenza di caratteri ascetici) in un tipo di dati che corrisponde al contenuto della stringa. Se la stringa sembra un numero in virgola mobile, Val () la trasforma in un numero in virgola mobile. Val () usa la lingua locale della tua macchina (qui tedesco), che è stata specificata in "locales".

Puoi controllare le tue impostazioni con il seguente comando della console: "locales" o "echo \$ LANG".

Su una macchina tedesca ottieni: de_DE @ euro

Attenzione: con i numeri decimali e l'impostazione del paese! Il comando Val è soggetto a errori!

Se un Textbox1 contiene 6.6 con un'impostazione del paese tedesco, Val (textbox1.text) restituisce una variabile float con il valore 6.6.

Se la casella di testo dice 6.6, Val (textbox1.text) restituisce una data 06.06.04

Se vuoi riconvertire il numero in un testo tedesco, usa Str (numero) per la conversione.

Prova semplicemente le impostazioni del paese del tuo PC con il seguente programma. L'output avviene nella finestra diretta:

```
PUBLIC SUB Form_Open ()  
PRINT Val ("6.6.99")  
STAMPA Val ("09/06/72 01:00")  
PRINT Val ("3.1415")  
PRINT Val ("3.1415")  
STAMPA Val ("- 25")  
PRINT Val ("True")  
PRINT Val ("False")  
PRINT IsNull (Val ("Gambas"))  
FINE
```

Se l'impostazione del paese è tedesca, il risultato è:

```

06/06/99
31415
3.1415
-25
Vero
Falso
Vero

```

Un altro programma di esempio

Guarda anche: <http://www.madeeasy.de/7/prgval.htm>

Se si immette una stringa di testo, il programma converte la stringa con Val e viene verificato il tipo di variabile convertita.

Il risultato è l'output.

```

PUBLIC SUB _new ()
    TextLabel1.Text = "Digita qualcosa come una data, a
Sembra un numero, un punto decimale o una variabile booleana "
    FINE

Pulsante PUBLIC SUB1_Click ()
    x variante AS
    x = Val (TextBox1.Text)

    IF IsDate (x) THEN Label1.Text = "Questa è una data!"
    IF IsBoolean (x) THEN Label1.Text = "Questo è un booleano
Variabile!"
    IF IsInteger (x) THEN Label1.Text = "Questo è un numero!"
    IF IsFloat (x) THEN Label1.Text = "Questo è un numero punto!"
    IF IsString (x) THEN Label1.Text = "Questa è una stringa!"
    FINE

```

Attenzione: se si immette True, viene riconosciuta una variabile booleana, ma il valore non viene emesso in una seconda casella di testo. Se si immette False, viene riconosciuta una variabile booleana, ma il valore non viene emesso in una seconda casella di testo.

0.120.1 Teoria

Espressione = Val (stringa)

Converte una stringa in un valore booleano, un numero o una data, a seconda del contenuto della stringa. Le impostazioni locali correnti vengono utilizzate durante la conversione di numeri e date.

La conversione avviene come segue:

- Se la stringa può essere interpretata come data e ora (con separatori di data e ora), vengono restituite data e ora.
- Altrimenti, se la stringa può essere interpretata come un numero in virgola mobile, viene restituito un numero in virgola mobile.
- Altrimenti, se la stringa può essere interpretata come un numero intero, viene restituito un numero intero.

- Altrimenti, se la stringa è TRUE o FALSE, viene restituito il valore booleano corrispondente.
- Altrimenti viene restituito NULL.

0.121 Calcola il numero di Eulero

Guarda anche <http://www.madeeasy.de/7/prge.htm> e <http://www.madeeasy.de/2/e.htm>

La seguente formula può essere utilizzata per calcolare il numero di Eulero.

$$e = 1 + 1/1! + 1/2! + 1/3! + 1/4! + \dots + 1 / n!$$

La formula è incorporata in un semplice programma. Nell'esempio, la sequenza infinita viene annullata dopo 15 passaggi di loop. Ma puoi anche calcolare il numero in modo più preciso.

Il programma è codificato come un programma terminale senza grafica.

```
STATICO PUBBLICO SUB Principale ()  
E AS float  
QF AS galleggiante  
F AS intero  
K AS intero  
  
E = 1  
F = 1  
PER K = DA 1 A 15  
    F = F * K  
    QF = CFloat (1) / F  
    E = E + QF  
    STAMPA E.  
IL PROSSIMO  
FINE
```

0.122 Calcola la sequenza di Fibonacci

Guarda anche: <http://www.madeeasy.de/7/prgfibo.htm>

Il seguente programma può essere utilizzato per calcolare la sequenza di Fibonacci.

È codificato come un programma terminale senza grafica.

```
STATICO PUBBLICO SUB Principale ()  
un numero intero AS  
b AS intero  
x AS intero  
  
a = 1  
b = 1  
PER x = da 1 a 23  
    STAMPA 2*x-1, a  
    STAMPA 2*x, b  
    a = a + b
```

b = a + b
IL PROSSIMO
FINE

Produzione:

```
1   1
2   1
3   2
4   3
5   5
6   8 °
7   13 °
8   21
934
1055
1189
12144
13233
14377
15610
16987
171597
182584
194181
206765
2110946
```

Prova il codice sopra come funzione⁷³ programmare:

L'input della funzione dovrebbe essere fibo (n). Il risultato è il numero di Fibonacci che appartiene al valore N corrispondente. Tuttavia, dovrà introdurre un limite di dimensione molto presto, poiché i numeri diventano molto grandi molto rapidamente.

0.123 Calcola la sequenza di Lucas

La sequenza di Lucas ha la stessa regola di formazione della sequenza di Fibonacci. Tuttavia, differisce nei due termini iniziali. Invece di 0 e 1, i due termini iniziali sono 2 e 1.

Il seguente programma può essere utilizzato per calcolare la sequenza di Lucas.

È codificato come un programma terminale senza grafica.

```
STATICO PUBBLICO SUB Principale ()
  un numero intero AS
  b AS intero
  x AS intero

  a = 2
  b = 1
  PER x = da 1 a 23
    STAMPA 2*x-1, a
    STAMPA 2*x, b
    a = a + b
    b = a + b
```

73 Capitolo 10.11 a pagina 395

IL PROSSIMO
FINE

produzione

```
1    2
2    1
3    3
4°   4°
5    7°
6    11
7    18°
8    29
947
1076
11123
12199
13322
14521
15843
161364
172207
183571
195778
209349
2115127
2224476
2339603
```

0.124 Programmare il simbolo somma å

La matematica causa notevoli difficoltà al principiante semplicemente perché tende a usare caratteri nuovi e complicati. Un esempio è il simbolo della somma. Se vuoi capirlo, dovresti semplicemente programmarlo. Dal punto di vista della programmazione, è un ciclo For Next con sommatoria dei risultati di ogni passaggio del ciclo.

Come puoi programmare il segno?

Usiamo 2 esempi per questo:

0.124.1 Esempio 1

$$\text{un} \begin{matrix} 3 \\ \text{io} \\ \text{io} = & 2 \end{matrix}$$

Il programma per questo ha questo aspetto: hai bisogno di una casella di testo e di un pulsante di comando per farlo funzionare.

Codice:

```
PUBLIC SUB Form_Open ()
    Button1.Text = "Calcola la somma"
```

```

Textbox1.Text = ""
FINE

Pulsante PUBLIC SUB1_Click ()
    i AS intero
    'i è la variabile di esecuzione da -2 a 3
    s AS Variant
    È la somma
    s = 0
    PER i = da -2 a 3
        s = s + i
        'STAMPA i, s
    IL PROSSIMO
    textbox1.Text = Str (s)
FINE

```

Modicare il programma in modo da poter inserire il valore iniziale e il valore finale della variabile di esecuzione i in una casella di testo. Per fare ciò, inserire 2 nuove variabili nel programma

```

a come numero intero
' a = valore iniziale e
come numero intero
' e = valore finale

```

Cambia il loop in

```
Per i = a fino a e
```

Prima di fare ciò, devi assegnare valori a a ed e:

```

a = 0
e = 6

```

O:

```

a = val (Textbox2.Text)
e = val (Textbox3.Text)

```

0.124.2 Esempio 1a

```

10
å ii
io = 1

```

Il programma per questo ha questo aspetto: hai bisogno di una casella di testo e di un pulsante di comando per farlo funzionare.

Codice:

```

PUBLIC SUB Form_Open ()
    Button1.Text = "Calcola la somma"

```

Contenuti

```
Textbox1.Text = ""  
FINE  
  
Pulsante PUBLIC SUB1_Click ()  
    i AS intero  
    'i è la variabile di esecuzione da 1 a 10  
    s AS Variant  
    s = 0  
    PER i = da 1 a 10  
        s = s + i*io  
        'STAMPA i, s  
    IL PROSSIMO  
    textbox1.Text = Str (s)  
FINE
```

Modificare il programma in modo da poter inserire il valore iniziale e il valore finale della variabile di esecuzione i in una casella di testo. Per fare ciò, inserire 2 nuove variabili nel programma

```
a come numero intero  
' a = valore iniziale e  
come numero intero  
' e = valore finale
```

Cambia il loop in

```
Per i = a fino a e
```

Prima di fare ciò, devi assegnare valori a a ed e:

```
a = 1  
e = 10
```

O:

```
a = val (Textbox2.Text)  
e = val (Textbox3.Text)
```

0.124.3 Esempio 2

Dovrebbe essere calcolato

$$\begin{array}{c} 6^{\circ} \text{ io} \\ \text{un} \frac{i+1}{\overline{}} \\ \text{io} = 0 \end{array}$$

Il programma per questo assomiglia a questo:

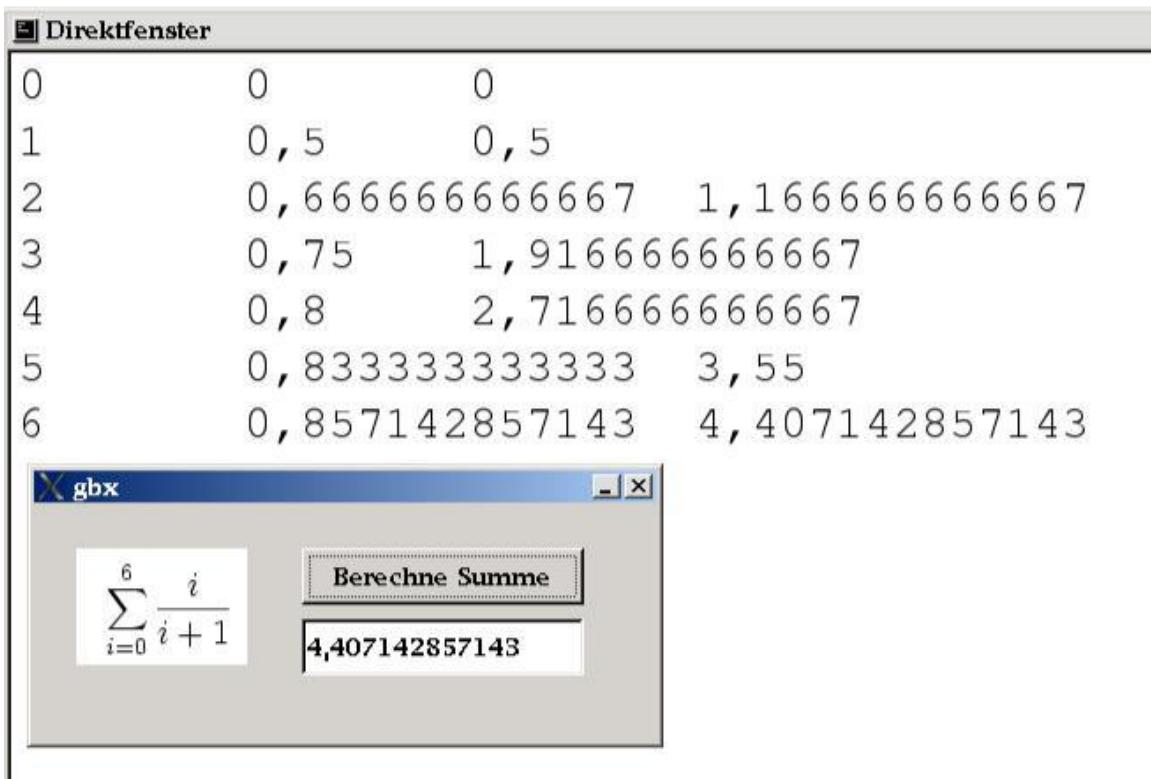


Fig.42

I valori intermedi sono stati emessi nella finestra diretta con la stampa. Hai bisogno di una casella di testo e di un pulsante di comando per farlo funzionare.

```

PUBLIC SUB Form_Open ()
    Button1.Text = "Calcola somma"
    Textbox1.Text = ""
FINE

Pulsante PUBLIC SUB1_Click ()
    i AS intero
    'i è la variabile di esecuzione da 0 a 6
    s AS Variant
    s = 0
    PER i = da 0 a 6
        s = s + i / (i + 1)
        STAMPA i, (i / (i + 1)), s
        '0 + 1/2 + 2/3 + 3/4 + 4/5 + 5/6 + 6/7 = 210/420 + 280/420 + 315/420 + 336/420 +
    350/420 + 360/420
    IL PROSSIMO
    textbox1.Text = Str (s)
FINE

```

Puoi vedere la differenza tra il calcolo del computer e il tuo calcolo su carta. $0 + 1/2 + 2/3 + 3/4 + 4/5 + 5/6 + 6/7 = 210/420 + 280/420 + 315/420 + 336/420 + 350/420 + 360/420$ Si cerca il minimo comune denominatore sulla carta. Quindi espandi le frazioni di conseguenza. I contatori possono quindi essere aggiunti. Risultato $1851/420 = 617/140 = 4,407142857142857$

Il vantaggio decisivo del programma per computer è che può essere facilmente ampliato. Si può lasciare il valore iniziale 0 per i come variabile generale. Si può lasciare il valore finale per i = 6 generale. Quindi il programma non è adatto solo per una somma speciale, ma anche per eventuali somme dello stesso tipo.

```
PUBLIC SUB Form_Open ()  
    Button1.Text = "Calcola somma"  
    Textbox1.Text = ""  
FINE  
  
Pulsante PUBLIC SUB1_Click ()  
    i AS intero  
    'i è la variabile di esecuzione da 0 a 6  
    a come numero intero  
    'a è il valore iniziale di i  
    z come numero intero  
    'z è il valore finale per i  
    s AS Variant  
  
    s = 0  
    a = 0  
    z = 6  
    PER i = a TO z  
        s = s + i / (i + 1)  
        'STAMPA i, (i / (i + 1)), s  
    IL PROSSIMO  
    textbox1.Text = Str (s)  
FINE
```

Contare 0,125 numeri

Questo programma conta la frequenza di 1,2,3,4,5,6,7,8,9,0 da una lunga riga di numeri decimali. È programmato in modo tipico per principianti. Non c'è da vergognarsi perché serve allo scopo. Un array potrebbe probabilmente abbreviare notevolmente il programma.

I risultati vengono visualizzati come al solito nella finestra diretta.

Per avviare il programma è necessaria un'area di testo e 2 pulsanti di comando sul modulo.

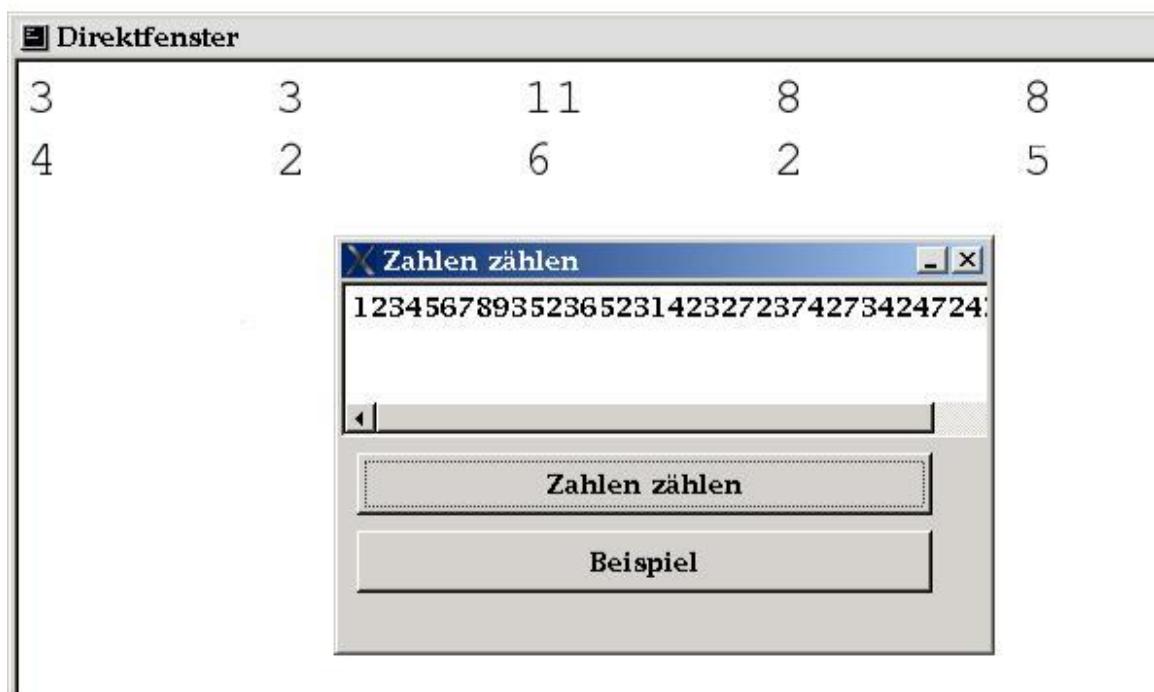


Fig.43 Immagine dello schermo

```

PUBLIC SUB Form_Open ()
    Button1.Text = "Conta numeri"
    Button2.Text = "Esempio"
    Textarea1.Text = ""
    ME.Text = "Conta numeri"
FINE

Pulsante PUBLIC SUB1_Click ()
    s come stringa
    n AS intero
    Tempo AS stringa
    char AS stringa
    z0 AS intero
    z1 AS intero
    z2 AS intero
    z3 AS intero
    z4 AS intero
    z5 AS intero
    z6 AS intero
    z7 AS intero
    z8 AS intero
    z9 AS intero

    s = textarea1.Text
    z0 = 0
    z1 = 0
    z2 = 0
    z3 = 0
    z4 = 0
    z5 = 0
    z6 = 0
    z7 = 0
    z8 = 0
    z9 = 0

    Tempo = RTrim $(s)

```

```
Tempo = LTrim $ (Tempo)
FOR n = 1 TO Len (Tempo)
    char = Mid $ (Tempo, n, 1)
    SELEZIONA CASE char
        CASO "0"
            z0 = z0 + 1
        CASO "1"
            z1 = z1 + 1
        CASO "2"
            z2 = z2 + 1
        CASO "3"
            z3 = z3 + 1
        CASO "4"
            z4 = z4 + 1
        CASO "5"
            z5 = z5 + 1
        CASO "6"
            z6 = z6 + 1
        CASO "7"
            z7 = z7 + 1
        CASO "8"
            z8 = z8 + 1
        CASO "9"
            z9 = z9 + 1
    FINE SELEZIONA
    IL PROSSIMO
    STAMPA z0, z1, z2, z3, z4
    STAMPA z5, z6, z7, z8, z9
FINE

PUBLIC SUB Button2_Click ()
    textarea1.Text =
    "1234567893523652314232723742734247247247899990001223445"
FINE
```

0.126 Rappresentazione funzionale

Sfortunatamente il comando VB Scale non funziona ancora in Gambas. Ce ne sono ancora alcuni utili rappresentazioni funzionali: <http://www.madeeasy.de/7/prgkoord.htm> <http://www.madeeasy.de/7/prgfunkt.htm>

0.127 Funzioni trigonometriche nei gambas

Acsh / ACosh, Acs / ACos, Asn / ASin, Asnh / ASinh, Atn / ATan, Athn / ATanh, Cos, Cosh, Deg, Rad, Pi, Sin, Sinh

0.127.1 Programma di esempio per seno e cos senza grafici

```
STATICO PUBBLICO SUB Principale ()
    DIM un galleggiante AS
    PER a = da - 10 a 10 PASSO 0.1
        STAMPA a, "Peccato", Peccato (a), "      Cos ", Cos (a)
    IL PROSSIMO
FINE
```

Ci sono ancora alcuni errori nell'output. Ad esempio, il valore $a = 0$ non viene emesso correttamente. Anche i valori $a = 0,1$ e $a = 10$ causano difficoltà.

0.127.2 Uscita grafica della funzione seno

Per favore, riferisci <http://www.madeeasy.de/7/prgsinohne.htm>

Questo programma grafico mostra la programmazione della funzione seno: è necessario un pulsante di comando e un campo di disegno. Impostare l'altezza del campo di disegno nelle proprietà su 100 (Altezza).

Il codice seguente è dietro il comando Sine curve: The following code is behind the Sine curve command:

```
Pulsante PUBLIC SUB1_Click ()
    DIM dymax AS Integer
    ' valore y massimo disegnato = Altezza drawymax DIM dymin AS
    Integer
    ' valore y minimo disegnato = 0 drawymin
    DIM ymax AS intero
    DIM ymin AS Integer
    'Sistema di coordinate desiderato con ymax e ymin DIM y AS
    Float
    'cambiando il valore originale di y
    DIM dy AS float
    'valore tracciato di y
    DIM dyi AS Integer
    'Valore intero di dy
    DIM x AS float
    DIM dxi AS Integer
    'Valore intero di dx

    ' Esempio dymax = 100 e dymin = 0 dymax =
    100
    dymin = 0
    ymax = 1
    ymin = -1

    PER x = da 0 a 12 PASSO 0.1
        y = peccato (x)
        'Formula
        'Il rapporto distanza parziale / distanza totale in entrambe le scale è lo stesso
        '(dy - dymin) / (dymax - dymin) = (y - ymin) / (ymax - ymin)' ==> dy - dymin = (y -
        ymin) / (ymax - ymin) * (dymax - dymin) ==> dy = (y - ymin) / (ymax - ymin) * (dymax
        - dymin) + dymin
        dy = CFloat (y - ymin) / (ymax - ymin) * (dymax - dymin) + dymin
        'PRINT x, y, dy
        dyi = fix (dy)
        dxi = Fix (x *30)
        Draw.Begin (DrawingArea1)
        Draw.Point (dxi, dyi)
        Disegna Fine
    IL PROSSIMO
    FINE
```

Gioca con le dimensioni individuali e guarda come cambia la grafica. Modificare il programma in modo che il seno inizi correttamente con 0 e diventi solo positivo. Cosa c'è che non va qui? (I valori dy sono ancora tracciati dall'alto verso il basso e non viceversa.) Prova a ottenere le conversioni per dyi in un'unica formula.

Logica 0.128

Vedere Gambas: operatori logici⁷⁴

Calcola 0,129 numeri primi

Il seguente programma mostra la ricerca dei numeri primi utilizzando il setaccio di Eratostene⁷⁵. I risultati vengono visualizzati nella finestra diretta.

Per farlo, hai bisogno di 2 pulsanti di comando e 1 casella di testo nel modulo.

```
PUBLIC SUB Form_Open ()  
    ME.Text = "Calcolo dei numeri primi"  
    FINE  
  
Pulsante PUBLIC SUB1_Click ()  
    m AS intero  
    f AS intero  
    n AS intero  
    Una stringa $ AS  
  
    if textbox1.Text = "" Allora textbox1.text = "1"  
    m = Val (textbox1.Text)  
    'ottiene il primo numero dal campo di testo1 per il test  
  
    SE m / 2 = Int (m / 2) ALLORA m = m - 1  
    'Se questo numero è divisibile per 2 senza resto, cioè se è un numero pari  
  
    'il programma torna indietro di un numero per ottenere un numero dispari  
  
    IF textbox1.Text = "" THEN m = 6  
    'Se non è stato inserito alcun numero di partenza, viene assegnato il numero di  
    partenza m = 6.  
  
    PER m = m TO m + 1000 FASE 2  
        'Anello principale  
        'Fai scorrere i prossimi 1000 numeri dispari  
        'm è la variabile per i numeri dispari  
  
        f = 1  
        'f è la variabile per la verifica dei fattori,  
        sono divisibile per f oppure no  
  
        n = m  
        'n = è il numero del test per il quale non è ancora chiaro se si tratta di un numero  
        primo  
  
        FARE MENTRE f < Sqr (n)  
            'fintanto che il divisore f è minore della radice  
            'è fuori dalla prova il numero n deve essere testato  
            f = f + 2  
            'Aumenta il divisore di f = 1 di 2 ogni volta, 3,5,7,9 ecc. Prima prova quindi con f  
            = 3  
            DO WHILE n / f = Int (n / f)  
                'Verifica il divisore f finché n / f è divisibile senza resto
```

74 Capitolo 10.2 a pagina 383

75 <http://de.wikipedia.org/wiki/Sieb%20des%20Eratosthenes>

```

'PRINT "m =" ; m; "n =" ; n; "f =" ; f
n = Int (n / f)
'Ridurre il numero di prova n al numero n / f
CICLO CONTINUO
CICLO CONTINUO

A $ = "1" & Chr (13) & "2" & Chr (13) & "3"
SE m <7 ALLORA STAMPA A $
'Chr (13) = interruzione di riga
'All'inizio tra 1 e 5 ci sono problemi con l'output' quindi le prime tre righe sono
sostituite da A $

SE n = m ALLORA STAMPA n
'Se il numero di prova n non era divisibile per f' è un numero
primo e può essere stampato
IL PROSSIMO
CATTURARE
    message.Info ("Immettere un numero")
FINE

```

Logaritmo 0,130

Il logaritmo può essere utilizzato per rendere chiari e piccoli numeri molto grandi. Ad esempio, 1.000.000 diventa 10^6 (diciamo 10 alla potenza di 6 = $10 * 10 * 10 * 10 * 10 * 10$). L'esponente 6 è il logaritmo di 1.000.000 al numero di base selezionato qui 10. I termini esponente ed esponente sono identici.

La funzione logaritmo $y = \log_b(x)$ restituisce un valore per qualsiasi numero positivo x . Quindi il logaritmo del numero 1 234 567 in base 10 è $\log_{10}(1234567) = 6.09151466408626$. In particolare per valori molto piccoli o molto grandi o che possono occupare un intervallo di valori molto ampio, i logaritmi si sono dimostrati molto utili. Poiché la funzione logaritmo cresce costantemente con valori x maggiori, le relazioni logiche che si applicano al valore x rimangono per lo più anche per il logaritmo di x .

0.130.1 Uso dei logaritmi:

- pH = valore acido delle soluzioni chimiche
- dB = decibel = misura del volume
- bit = unità di informazione = misura della quantità di informazioni
- Carta logaritmica in statistica

Guarda anche:

- <http://www.madeeasy.de/2/log.htm>
- Matematica: _Matematica scolastica: _ Logaritmo⁷⁶
- w: logaritmo⁷⁷

⁷⁶ http://de.wikibooks.org/wiki/Mathematik%3A_Schulmathematik%3A_Logaritme

⁷⁷ <http://de.wikipedia.org/wiki/Logarithmus>

0.130.2 Esempio di comando di registro del programma

Per comprendere il comando Log, puoi semplicemente calcolarlo per diversi valori: $L = \log_e(x)$ ⁷⁸ L è il logaritmo naturale di x alla base e, il numero di Eulero

e = 2,718281828459

Scrivi un mini programma con un modulo e un pulsante di comando:

Il codice per questo è:

```
PUBLIC SUB Form_Open ()  
    Button1.Text = "log"  
FINE  
  
Pulsante PUBLIC SUB1_Click ()  
    x AS intero  
    PER x = da 1 a 10  
        STAMPA x, Registro (x)  
        IL PROSSIMO  
FINE
```

L'output avviene nella finestra diretta. È:

```
1     0  
2     0.69314718056  
3     1.098612288668  
4     1.38629436112  
5     1.609437912434  
6     1.791759469228  
7     1.945910149055  
8     2.07944154168  
92.197224577336  
102.302585092994
```

Se avvii il tuo ciclo per x con un numero negativo, riceverai un messaggio di errore:

Errore matematico

Perché non esiste un logaritmo di un numero negativo.

Versione più accurata

Se desideri eseguire il ciclo in passaggi più precisi, utilizza la seguente variante di codice:

```
PUBLIC SUB Form_Open ()  
    Button1.Text = "log"  
FINE
```

78 <http://www.madeeasy.de/2/e.htm>

```
Pulsante PUBLIC SUB1_Click ()
    x AS float
    PER x = da 0,1 a 10 PASSO 0,1
        STAMPA x, Registro (x)
    IL PROSSIMO
FINE
```

Output nella finestra diretta:

```
0.1      -2.302585092994
0.2      -1.609437912434
0.3      -1.203972804326
0.4      -0.916290731874
0.5      -0.69314718056
0.6      -0.510825623766
0.7      -0.356674943939
0.8      -0.223143551314
0.9      -0.105360515658
1       -1.110223024625E-16
1,10,095310179804
1,20,182321556794
1,30,262364264467
1,40,336472236621
```

Eccetera...

0.130.3 Esempio di comando di registro del programma con qualsiasi input

Se vuoi testare il comando log con qualsiasi numero positivo, puoi usare il seguente programma.

Hai bisogno di 2 caselle di testo e 1 pulsante di comando nel modulo. Rendi la Textbox2 abbastanza grande in modo che ci sia spazio per tutte le cifre decimali.

Codice:

```
PUBLIC SUB Form_Open ()
    ME.Text = "Cosa fa il comando log"
    Textbox1.Text = ""
    Textbox2.Text = ""
    Button1.Text = "Log"
FINE

Pulsante PUBLIC SUB1_Click ()
    x AS float
    L AS galleggiante

    IF Textbox1.Text = "" THEN textbox1.text = "1"
    IF Val (textbox1.Text) <= 0 THEN
        message.info = "Input non valido"
        RITORNO
    FINISCI SE

    'Converti il testo immesso 1 in un numero esatto
    x = Val (textbox1.Text)
    L = log (x)
    'Emissione del risultato come stringa nel campo di testo 2
    textbox2.Text = Str (L)
```

FINE

C'è ancora un bug nel codice di questo programma. La casella dei messaggi (messaggio) è stata programmata in modo errato:

Invece di:

```
IF Val (textbox1.Text) <= 0 THEN message.info = "Voce non valida"
```

Dovresti scrivere:

```
IF Val (textbox1.Text) <= 0 THEN Message.info ("Numero di voce non valido inferiore a zero", "OK")
```

Come si può impedire l'inserimento di lettere errate?

0.130.4 Programma di esempio doppio logaritmo

Il logaritmo duale utilizza come base il numero 2. Per ottenerlo con il log dei comandi, devi convertire il naturale in duale. per questo viene utilizzata la seguente formula:

```
L = log2x = logex / loge2
```

Codice:

```
PUBLIC SUB Form_Open ()  
    ME.Text = "Doppio logaritmo"  
    Textbox1.Text = ""  
    Textbox2.Text = ""  
    Button1.Text = "Log2"  
FINE  
  
Pulsante PUBLIC SUB1_Click ()  
    x AS float  
    L AS galleggiante  
  
    x = Val (textbox1.Text)  
    L = Log (x) / Log (2)  
    'Emissione del risultato come stringa nel campo di testo 2  
    textbox2.Text = Str (L)  
FINE
```

0.130.5 Logaritmo decadico del programma di esempio

Il logaritmo decadico utilizza come base il numero 10. Per ottenerlo con il log dei comandi, devi convertire il naturale in decadico. per questo viene utilizzata la seguente formula:

L = log10x = logex / loge10

Codice:

```
PUBLIC SUB Form_Open ()  
    ME.Text = "Doppio logaritmo"  
    Textbox1.Text = ""  
    Textbox2.Text = ""  
    Button1.Text = "Log2"  
FINE  
  
Pulsante PUBLIC SUB1_Click ()  
    x AS float  
    L AS galleggiante  
  
    x = Val (textbox1.Text)  
    L = Log (x) / Log (10)  
    'Emissione del risultato come stringa nel campo di testo 2  
    textbox2.Text = Str (L)  
FINE
```

0.130.6 Calcolatore del logaritmo del programma di esempio

Il seguente programmino può essere utilizzato per calcolare i logaritmi. Per farlo funzionare hai bisogno di:

- 5 pulsanti di comando
- 4 caselle di testo
- 3 campi di etichette di testo

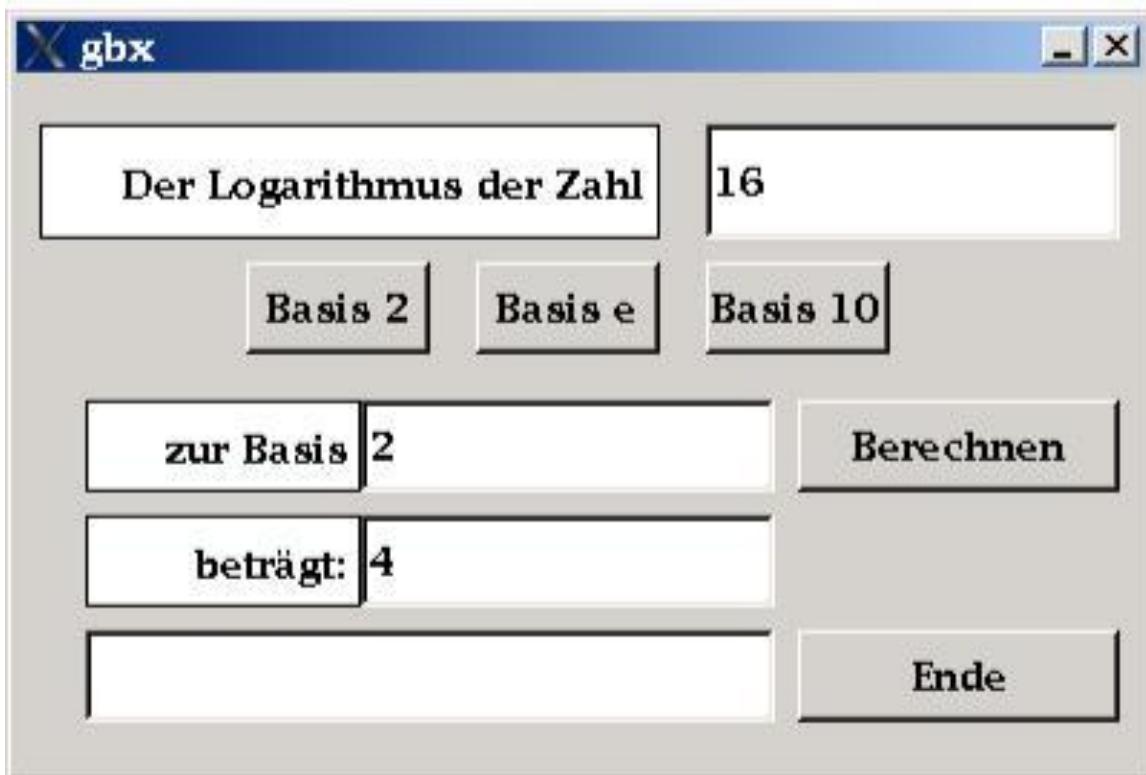


Fig.44

Il codice:

```

PUBLIC SUB Form_Open ()
    'Denominazione automatica dei controlli all'avvio del programma
    Textbox1.Text = ""
    Textbox2.Text = ""
    Textbox3.Text = ""
    textbox4.Text = ""
    Button1.Text = "Base 2"
    Button2.Text = "Base e"
    Button3.Text = "Base 10"
    Button4.Text = "Calcola"
    Button5.Text = "Fine"
    Textlabel1.Text = "Il logaritmo del numero"
    Textlabel2.Text = "alla base"
    Textlabel3.Text = "è:"
    Textlabel1.Alignment = 66
    Textlabel2.Alignment = 66
    Textlabel3.Alignment = 66
    Textlabel1.BackColor = & Hfffff &
    Textlabel2.BackColor = & HFFFFFF &
    Textlabel3.BackColor = & HFFFFFF &
    Textlabel1.Border = 1
    Textlabel2.Border = 1
    Textlabel3.Border = 1
    Form1.Move (174,455,393,242)
    Form1.Border = Window.Fixed
    Button1.Move (80,64,64,32)
    Button2.Move (160,64,64,32)
    Button3.Move (240,64,64,32)
    Button4.Move (272,112,112,32)
    Button5.Move (272,192,112,32)

```

```

TextBox1.Move (240,16,144,40)
TextBox2.Move (120,112,144,32)
TextBox3.Move (120,152,144,32)
TextBox4.Move (24,192,240,32)
TextLabel1.Move (8,16,216,40)
TextLabel2.Move (24,112,96,32)
TextLabel3.Move (24,152,96,32)
FINE

Pulsante PUBLIC SUB 4_Click ()
    x AS float
    y AS float
    L AS galleggiante

    textbox4.Text = ""
    'Intercetta le voci errate
    IF Textbox1.Text = "" OR textbox2.Text = "" THEN textbox4.text = "Input mancante!"

    IF Textbox1.Text = "" OR textbox2.Text = "" THEN RETURN
    IF Val (textbox1.Text) <= 0 THEN textbox4.Text = "Voce non valida"

    IF Val (textbox1.Text) <= 0 THEN RETURN
    IF Val (textbox2.Text) <= 1 THEN textbox4.Text = "Voce non valida"

    IF Val (textbox2.Text) <= 1 THEN RETURN

    " Converte il testo immesso 1 e 2 in un numero esatto
    x = Val (textbox1.Text)
    y = Val (textbox2.Text)

    'formula logaritmica generale
    L = Log (x) / Log (y)

    'Emissione del risultato nel campo di testo 3
    textbox3.Text = Str (L)
FINE

Pulsante PUBLIC SUB1_Click ()
    textbox2.Text = 2
FINE

Pulsante PUBLIC SUB 3_Click ()
    textbox2.Text = 10
FINE

PUBLIC SUB Button2_Click ()
    textbox2.Text = "2,718282"
FINE

Pulsante PUBLIC SUB 5_Click ()
    ME.Chiudi
FINE

```

Calcolo interessi 0,131

Le formule di calcolo degli interessi di seguito utilizzano i simboli come segue:

- Capitale iniziale: K_0 (Capitale dopo 0 anni)
- Capitale finale: K_n (Capitale dopo n anni)
- Durata (anni interi): n Ingresso in anni
- Durata (giorni): t Inserimento in giorni

- Tasso di interesse in decimale: $i = 100\% P$ (per periodo di interesse)

0.131.1 Esempio di programma di interesse semplice senza interesse composto (interesse lineare)

Anfangskapital: 1000 Euro

Laufzeit: 10 Jahre

Zinssatz: 6 %

Berechnen

Endkapital: 1600 Euro

Fig.45 miniatura

Per calcolare l'interesse semplice, usiamo la seguente formula:

$$K_n = K_0 \cdot (1 + n \cdot i)$$

Per il programma di esempio hai bisogno di:

- 4 caselle di valore
- 1 pulsante di comando

oppure utilizza il seguente codice .form

```
# Gambas Form File 2.0
{Form Form
    MoveScaled (0,0,53,31)
    Text = ("")
    {Label1 Label MoveScaled
        (45,8,6,3)
        Testo = ("anni")
    }
    {Label2 label MoveScaled
        (45,13,4,3)
        Testo = ("%")
    }
    {Label3 Label MoveScaled
        (1,3,14,3)
        Testo = ("capitale iniziale")
    }
```

```

{Label4 Label MoveScaled
 (4,8,11,3)
 Text = ("runtime")
}
{Label5 Label MoveScaled
 (4,13,11,3)
 Testo = ("tasso di interesse")
}
{Label6 Label MoveScaled
 (1,25,14,4)
 Testo = ("capitale finale")
}
{Button1 Button MoveScaled
 (7,18,38,5)
 Testo = ("Calcola")
}
{ValueBox1 ValueBox
 MoveScaled (16,3,28,3)
}
{Label7 Label MoveScaled
 (45,3,6,3) Text = ("Euro")
}
{ValueBox2 ValueBox
 MoveScaled (16.8,28,3)
}
{ValueBox3 ValueBox
 MoveScaled (16,13,28,3)
}
{ValueBox4 ValueBox
 MoveScaled (16,25,28,4)
}
{Label8 Label MoveScaled
 (45,25,6,3) Text = ("Euro")
}
}

```

Il codice .class

"File di classe Gambas

```

'K (capitale iniziale)      = ValueBox1.Value
'n  (Durata in anni)        = ValueBox2.Value
'io  (Tasso d'interesse)    = ValueBox3.Value / 100
'Kn (capitale finale)       = ValueBox4.Value

```

Pulsante PUBLIC SUB1_Click ()

'Calcola il capitale finale con la formula:

$$\text{Kn} = K * (\text{io} * n + 1)$$

```
ValueBox4.Value = (ValueBox1.Value * (ValueBox3.Value / 100 * ValueBox2.Value + 1))
```

FINE

0.131.2 Esempio di programma per il calcolo dell'interesse composto (interesse esponenziale)



Anfangskapital: 1000 Euro

Laufzeit: 10 Jahre

Zinssatz: 6 %

Mit Zinseszins

Berechnen

Endkapital: 1790,85 Euro

Zinsen: 790,85 Euro

Fig.46 miniatura

Per calcolare l'interesse composto, utilizziamo la seguente formula:

$$K_n = K_0 \cdot (1 + i)^n$$

Per il programma di esempio hai bisogno di:

- 5 caselle di valore
- 1 casella di controllo
- 1 pulsante di comando

oppure utilizza il seguente codice .form

File di modulo Gambas 2.0

```
{Form Form MoveScaled  
    (0,0,53,39) Text = ("")  
    {Label1 Label MoveScaled  
        (45,8,6,3) Text = ("Years")  
    }  
    {Label2 Label MoveScaled  
        (45,13,4,3) Text = ("%")  
    }
```

```

}
Etichetta {Label3
    MoveScaled (1,3,14,3)
    Testo = ("capitale iniziale")
}
{Label4 label
    MoveScaled (4,8,11,3)
    Text = ("runtime")
}
{Label5 label
    MoveScaled (4,13,11,3)
    Testo = ("tasso di interesse")
}
Etichetta {Label6
    MoveScaled (1,28,14,4)
    Testo = ("capitale finale")
}
Pulsante {Button1
    MoveScaled (7,22,38,5)
    Testo = ("Calcola")
}
{ValueBox1 ValueBox
    MoveScaled (16,3,28,3)
}
{Label7 Label MoveScaled
    (45,3,6,3) Text = ("Euro")
}
{ValueBox2 ValueBox
    MoveScaled (16,8,28,3)
}
{ValueBox3 ValueBox
    MoveScaled (16,13,28,3)
}
{ValueBox4 ValueBox
    MoveScaled (16,28,28,4)
}
{Label8 Label MoveScaled
    (45,28,6,3)
    Testo = ("Euro")
}
{Label9 Label MoveScaled
    (1,34,14,3)
    Testo = ("interesse")
}
{ValueBox5 ValueBox
    MoveScaled (16,34,28,3)
}
Etichetta {Label10
    MoveScaled (45,34,6,3)
    Testo = ("Euro")
}
{CheckBox1 CheckBox MoveScaled
    (16,18,27,2) Text = ("Con
    interesse composto")
}
}

```

Il codice .class

"File di classe Gambas

'K (capitale iniziale)	= ValueBox1.Value
'n (termine in anni)	= ValueBox2.Value
'io (Tasso d'interesse)	= ValueBox3.Value / 100
'Kn (capitale finale)	= ValueBox4.Value
(Interesse / valore degli	
'Z interessi)	= ValueBox5.Value

Contenuti

Pulsante PUBLIC SUB1_Click ()

```
IF CheckBox1.Value = FALSE THEN
'Calcola l'interesse normale

'Calcola il capitale finale utilizzando la formula:
'Kn = K*(io*n + 1)
ValueBox4.Value = Round (ValueBox1.Value * (ValueBox3.Value / 100 * ValueBox2.Value
+ 1), -2)
'Calcola interesse / valore di interesse utilizzando la formula:
'Z = K*(io*n + 1) -K = Kn-K
ValueBox5.Value = Round (ValueBox4.Value - ValueBox1.Value, -2)
```

ALTRO

'Calcola l'interesse composto

```
'Calcola il capitale finale utilizzando la formula:
'Kn = K*(i + 1) ^n
ValueBox4.Value = Round (ValueBox1.Value * (ValueBox3.Value / 100 + 1) ^ ValueBox2.Value, -
2)
'Calcola interesse / valore di interesse utilizzando la formula:
'Z = K*(i + 1) ^n-K = Kn-K
ValueBox5.Value = Round (ValueBox4.Value - ValueBox1.Value, -2)
```

FINISCI SE

FINE

0.132 Statistiche descrittive

0.132.1 media

La media aritmetica (anche media) è la media utilizzata più frequentemente ed è quindi indicata anche come media standard.

Puoi calcolare la media aritmetica di una serie di numeri sommando tutti gli elementi della serie e dividendo la somma ottenuta per il numero di elementi.

Valore medio = somma di valori / numero di valori

Esempio per la media aritmetica di 3, 4 e 5:

Media = $(3 + 4 + 5) / 3 = 4$

Ciò è implementato nel seguente esempio di programma.

Ottieni un'area di testo, una casella di testo e un pulsante di comando dalla casella degli strumenti. Disporli sul modulo come mostrato di seguito: I numeri sono intesi solo come esempio.

```
Textarea1
*****
* 5 *
```

```

* 5
* 4 °
* 5
*
*
*****
Pulsante di comando 1:
*****
*Media*
*****
Casella di testo 1
*****
*4.75
*****

```

Il codice dietro il pulsante di comando Valore medio ha questo aspetto e ricorda fortemente ADDY, il Totalizzatore:

```

Pulsante PUBLIC SUB1_Click ()
Testo DIM AS String
Somma DIM AS Float
DIM mw AS Float
    DIM Elt AS String []
DIM Sb AS String
text = textarea1.Text
Elt = Split (testo, Chr (10))
PER CIASCUNO Sb IN Elt
    somma = somma + Val (sb)
IL PROSSIMO
mw = sum / elt.Length
textbox1.Text = mw
FINE

```

I trucchi principali del programma sono un array di stringhe, il comando Split e il carattere CHR (10) per separare i singoli numeri.

0.132.2 Varianza e deviazione standard

Il programma per il calcolo del valore medio può essere ampliato in modo da poter calcolare anche la varianza e la deviazione standard. È incluso anche un set di dati di prova.

La distribuzione normale gaussiana mostra che la probabilità è circa 2/3 che i valori misurati siano entro +/- una deviazione standard dal valore atteso e quindi circa 1/6 che siano al di sotto o al di sopra di questo intervallo.

Hai bisogno di 3 pulsanti di comando, un'area di testo e 2 caselle di testo per farlo funzionare.



Fig.47 Media screenshot, varianza e
Calcola la deviazione standard

Il codice ha questo aspetto:

```
Pulsante PUBLIC SUB1_Click ()  
Testo DIM AS String  
Somma DIM AS Float  
DIM mw AS Float  
Varianza DIM AS Float  
DIM sigma AS Float  
Elenco DIM come stringa []  
DIM valore singolo AS stringa
```

```

text = textarea1.Text
List = Split (text, Chr (10))
PER OGNI singolo valore IN lista
    sum = sum + val (valore singolo)
IL PROSSIMO
mw = sum / list.Length
textbox1.Text = mw
varianza = 0
PER OGNI singolo valore IN lista
    varianza = varianza + ((Val (valore singolo) - mw) ^2)
    'STAMPA valore singolo, mw, ((Val (valore singolo) - mw) ^2)
    'PRINT varianza
IL PROSSIMO
varianza = varianza / (List.Length - 1)
'PRINT varianza
sigma = Sqr (varianza)
textbox2.Text = sigma
FINE
-----
PUBLIC SUB Button2_Click ()
    'Compila il record di dati del test nell'area di testo.
    textarea1.Text = "114,3"
    textarea1.Text = textarea1.Text & Chr (10) & "135.7" textarea1.Text
    = textarea1.Text & Chr (10) & "104.8" textarea1.Text = textarea1.Text
    & Chr (10) & "118.5" textarea1. Text = textarea1.Text & Chr (10) &
    "125.7" textarea1.Text = textarea1.Text & Chr (10) & "121.4"
    textarea1.Text = textarea1.Text & Chr (10) & "122.4" textarea1.Text =
    textarea1.Text & Chr (10) & "96.8" textarea1.Text = textarea1.Text &
    Chr (10) & "118.9" textarea1.Text = textarea1. Text & Chr (10) & "120"
    textarea1.Text = textarea1. Text & Chr (10) & "112,2" textarea1.Text =
    textarea1.Text & Chr (10) & "127,9" textarea1.Text = textarea1.Text &
    Chr (10) & "122.8" textarea1.Text = textarea1.Text & Chr (10) &
    "128.9" textarea1.Text = textarea1.Text & Chr (10) & "120.3" 'mw =
    119.373

    'varianza = 92,6016
    'sigma = 9,62297
    FINE
-----
Pulsante PUBLIC SUB 3_Click ()
    ME.Chiudi
    FINE

```

Per un programmatore professionista, rivoltare lo stomaco per vedere il set di dati di prova in fase di caricamento. Ma funziona. E come una piccola sfida di programmazione tecnica, il caricamento del set di dati di test può essere notevolmente ridotto con il comando array.

Vedi Gambas: Arrays⁷⁹

0.132.3 La mediana

Il valore mediano o centrale di un elenco di valori si trova tra due metà dell'elenco di uguale dimensione. Il 50% dei valori è maggiore / uguale alla mediana, il 50% è minore / uguale alla mediana. A differenza della media aritmetica, difficilmente può essere influenzata da valori estremi (valori anomali) nell'elenco dei valori.

79 Capitolo 6.157 a pagina 360

Se vuoi programmare la mediana devi:

- 1. Ordinare l'elenco dei valori
- 2. Verificare se si tratta di un numero pari o di un numero dispari di valori.
- 3. Utilizzare le formule appropriate per calcolare la mediana.
- Con un numero dispari di valori

$$x_{\text{median}} = x[(n + 1) / 2]$$

- • Se il numero di valori è pari

$$x_{\text{median}} = 1/2 * (x[n / 2] + x[n / 2 + 1])$$

Esempio 1:

elenco ordinato di valori:

```
11  
12 °  
13 °  
14th  
15 °
```

Il numero di valori è dispari, $n = 5$

$$(n + 1) / 2 = 6/2 = 3$$

$$x_{\text{median}} = x[3] = 13$$

Esempio 2.

elenco ordinato di valori:

```
11  
12 °  
13 °  
14th  
15 °  
16
```

Il numero di valori è pari, $n = 6$

$$\begin{aligned} n / 2 &= 6/2 = 3 \\ x[3] &= 13 \\ x[4] &= 14 \\ x_{\text{median}} &= 1/2 * (13 + 14) = 13,5 \end{aligned}$$

È possibile trasferire l'elenco di valori sopra riportato direttamente nel programma di esempio:
utilizzare il file

Appunti: evidenzia, premi Ctrl + C. Scarica di nuovo nel tuo programma con Ctrl + V.

Ordina e calcola la mediana. Funziona ?

Programma di esempio: sono necessarie 2 aree di testo, 3 pulsanti di comando e una casella di testo per impostare il file

Per avviare il programma:

```
' Elenco file CLASS
Gambas AS String [] PUBLIC
SUB Form_Open ()
    ME.Text = "Calcola valore medio"
FINE
PUBLIC SUB Button1_Click ()
'riordina
c AS Intero j AS Intero
n AS Intero y AS
Elenco varianti AS
String [] elemento AS
String txt AS String
text AS String

text = Textarea1.Text list = Split
(text, Chr (10)) y = 0

n = liste.length
RIPETI c
= 0
PER j = 0 TO n - 2
'PRINT j, y, list [0], ar [1], ar [2], ar [3], ar [4] IF Val (list
[j])> Val (list [j + 1]) THEN
    y = Val (lista [j]) lista [j] = lista [j
+ 1] lista [j + 1] = Str (y) c = 1

ENDIF
NEXT UNTIL c
= 0
PER OGNI elemento IN lista
txt = txt & Str (elemento) & Chr (10)
NEXT PRINT txt
textarea2.Text = "" txt =
Mid $ (txt, 1, -1)

'L'ultima interruzione di riga chr (10) deve scomparire
textarea2.Text = txt
FINE
PUBLIC SUB Button2_Click ()
'Calcola la mediana, riorganizza in anticipo il necessario !! testo
come stringa

median AS Float list AS
String [] post AS String text
= Textarea2.Text

list = Split (text, Chr (10))
"Determina la lunghezza e utilizza la formula corretta", a seconda
che la lunghezza sia pari o dispari. IF liste.Lunghezza MOD 2 POI

'PRINT liste.Length MOD 2 & "Odd" PRINT
(liste.length + 1) / 2
'PRINT list [(liste.length + 1) / 2 - 1] median = Val (list
[(liste.length + 1) / 2 - 1])
'L'array inizia con l'elemento 0 non con 1!
ALTRO
'PRINT liste.Length MOD 2 & "Even" median = (Val (liste
[liste.length / 2 - 1]) +
Val (liste [liste.length / 2])) / 2
'L'array inizia con l'elemento 0 non con 1!
FINISCI SE
```

```
textbox1.Text = Str (mediana)
FINE
Pulsante PUBLIC SUB 3_Click ()
'Elenco dispari di valori come esempio
textarea1.Text = "114,3"
textarea1.Text = textarea1.Text & Chr (10) & "135.7" textarea1.Text
= textarea1.Text & Chr (10) & "104.8" textarea1.Text = textarea1.Text
& Chr (10) & "118.5" textarea1.Text = textarea1.Text & Chr (10) &
"125.7" textarea1.Text = textarea1.Text & Chr (10) & "121.4"
textarea1.Text = textarea1.Text & Chr (10) & "122.4" textarea1.Text =
textarea1.Text & Chr (10) & "96.8" textarea1.Text = textarea1.Text &
Chr (10) & "118.9" textarea1.Text = textarea1.Text & Chr (10) &
"120" textarea1.Text = textarea1.Text & Chr (10) & "112,2"
textarea1.Text = textarea1.Text & Chr (10) & "127,9" textarea1.Text =
textarea1.Text & Chr (10) & "122.8" textarea1.Text = textarea1.Text &
Chr (10) & "128.9" textarea1.Text = textarea1.Text & Chr (10) &
"120,3" 'mediana = 120,3

FINE
```

0.133 Distribuzione normale gaussiana

Il seguente programma calcola i valori della distribuzione gaussiana tra -5 e 5. Vedi anche: <http://www.madeeasy.de/2/gauss.htm>

```
PUBLIC SUB Form_Open ()
DIM x AS float
DIM y AS Float
DIM t AS stringa
PER x = da -5 a 5 PASSO 0.1
y = 2,718 ^ (-x * X)
t = t & Str (x) & "" & Str (y) & Chr (13) & Chr (10)
IL PROSSIMO
STAMPA t
FINE
```

0,134 test

0.134.1 test a quattro campi

Il test in quattro campi è uno strumento semplice ma molto importante per la valutazione statistica delle caratteristiche in 2 gruppi. Guarda anche <http://www.madeeasy.de/7/4feld.htm>

Per eseguire il programma è necessario:

- 4 campi etichetta
- 4 caselle di testo
- 1 area di testo
- 3 pulsanti di comando.

Il codice è:

```

PUBLIC SUB Form_Open ()
    ME.Text = "test in quattro campi"
FINE
Pulsante PUBLIC SUB1_Click ()
    ME.Chiudi
FINE
PUBLIC SUB Button2_Click ()
    textbox1.Text = ""
    textbox2.Text = ""
    textbox3.Text = ""
    textbox4.Text = ""
    textarea1.Text = ""
FINE
Pulsante PUBLIC SUB 3_Click ()
    ' Rem incorpora routine di errore per i campi vuoti DIM N1 AS
    Float
    DIM N2 AS Float DIM M1
    AS Float DIM M2 AS
    Float DIM R1 AS Float
    DIM NN AS Float DIM N
    AS Float DIM E1 AS
    AS Float DIM E2 AS
    Float DIM MM AS Float
    DIM E2 AS Float DIM R2
    AS Float DIM E3 AS
    Float DIM E4 AS Float
    DIM D1 AS Float DIM C1
    AS Float DIM D2 AS
    Float DIM C2 AS Float
    DIM D3 AS Float DIM C3
    AS Float DIM D4 AS
    Float DIM C4 AS Float
    DIM C AS Float DIM S
    AS Float DIM P AS Float
    textarea1.Text = ""

N1 = Val (textbox1.Text)
N2 = Val (textbox2.Text)
M1 = Val (textbox3.Text)
M2 = Val (textbox4.Text)
'VALORE ATTESO EN1
R1 = N1 + M1
NN = N1 + N2
N = N1 + N2 + M1 + M2 E1 =
R1 * NN / N
textarea1.Text = textarea1.Text & "Valore atteso E1 =" & Str (E1) & Chr (13) & Chr (10)

'Rem VALORE ATTESO EN2
MM = M1 + M2
E2 = R1 * MM / N
textarea1.Text = textarea1.Text & "Valore atteso E2 =" & Str (E2) & Chr (10) & Chr (13)

'Rem VALORE ATTESO EN3
R2 = N2 + M2
E3 = R2 * NN / N
textarea1.Text = textarea1.Text & "Valore atteso E3 =" & Str (E3) & Chr (13) & Chr (10)

'Rem VALORE ATTESO EN4 E4
= R2 * MM / N

```

```
textarea1.Text = textarea1.Text & "Valore atteso E4 =" & Str (E4) &
Chr (13) e Chr (10)
'Rem PROBE = 0?
S = E1 + E2 + E3 + E4
P = S - N
textarea1.Text = textarea1.Text & "Probe" & Str (S) & "-" & Str (N) & "=" & Str (P) & "= 0?"
& Chr (13) e Chr (10)'Rem CHI'2

D1 = (N1 - E1) ^ 2
C1 = D1 / E1
D2 = (M1 - E2) ^ 2
C2 = D2 / E2
D3 = (N2 - E3) ^ 2
C3 = D3 / E3
D4 = (M2 - E4) ^ 2
C4 = D4 / E4
C = C1 + C2 + C3 + C4
textarea1.Text = textarea1.Text & "CHI-QUADRAT =" & Str (C) & Chr (13)
& Chr (10)
SE C - 3.841 > 0 ALLORA textarea1.Text = textarea1.Text & "A P = 0,05 c'è una differenza
significativa"
IF C - 3.841 <= 0 THEN textarea1.Text = textarea1.Text & "Con P = 0.05 non c'è differenza
significativa" END
```

0.134.2 Esegui test

Se una misurazione viene eseguita più volte per motivi di sicurezza, il runtest può essere utilizzato per verificare se i singoli valori misurati ottenuti sono sparsi in modo casuale attorno al valore reale o se la misurazione fornisce valori non casuali. In generale, il run test può essere utilizzato anche per verificare la reale casualità di generatori casuali come un dado o una moneta, ripetendoli più volte.

Guarda anche: <http://de.wikipedia.org/wiki/Run-Test>

Controllo della casualità di una moneta

La seguente serie di risultati è stata ottenuta lanciando più volte le monete.

```
0100100101110111010001001101100101101
```

Lo 0 sta per lo stemma e 1 per il numero di monete.

Una corsa è quindi una serie di lanci con lo stesso risultato.

Le seguenti esecuzioni sono nella sequenza

```
0
1
00
1
00
1
```

```

0
111
0
1
000
1
00
11
0
11
00
1
0
11
0
1

```

Ci sono un totale di 24 corse, ovvero sequenze con lo stesso carattere 0 o 1.

Come puoi ora controllare questa serie per la sua casualità?

Per questo sono necessarie alcune formule e parametri per il test di esecuzione.

- Hai bisogno del numero totale di cucciolate n.
- È necessario il numero totale di corse R.
- Hai bisogno del numero di 1 lanci n1.
- Hai bisogno del numero di tiri 0 n2.
- È necessario il valore atteso ER per il numero di esecuzioni statisticamente previsto.

$$ER = 2 * n1 * n2 / n + 1$$

- Hai bisogno della varianza varR e della radice della varianza.

$$varR = 2*n1*n2*(2*n1*n2-n) / (n*n*(n1 + n2-1))$$

- È necessario un livello di significatività p = 0,05
- Hai bisogno di una variabile di test z. Questo deve essere compreso tra -1,96 e +1,96, quindi si può assumere una sequenza casuale.

$$z = (R - ER) / sqrt(varR)$$

Come puoi programmare tutto questo?

Esempio: hai bisogno di

- 4 aree di testo,
- 13 pulsanti di comando
- 10 caselle di testo

sul modulo per avviare il programma.

Il funzionamento del programma richiede per abituarsi. È praticamente in senso orario. La guida dell'operatore può essere migliorata aggiungendo alcune routine di errore con CATCH e indicando il corretto funzionamento. Inoltre, il comando SETFOCUS è utile per il prossimo

Contenuti

Pulsante di comando contrassegnato che devi premere. Con poche subroutine sensibili e l'avvio del programma con la semplice pressione di un pulsante, tutto diventa più facile e meno soggetto a errori. Questo poi arriva nella versione migliorata 2.

disposizione

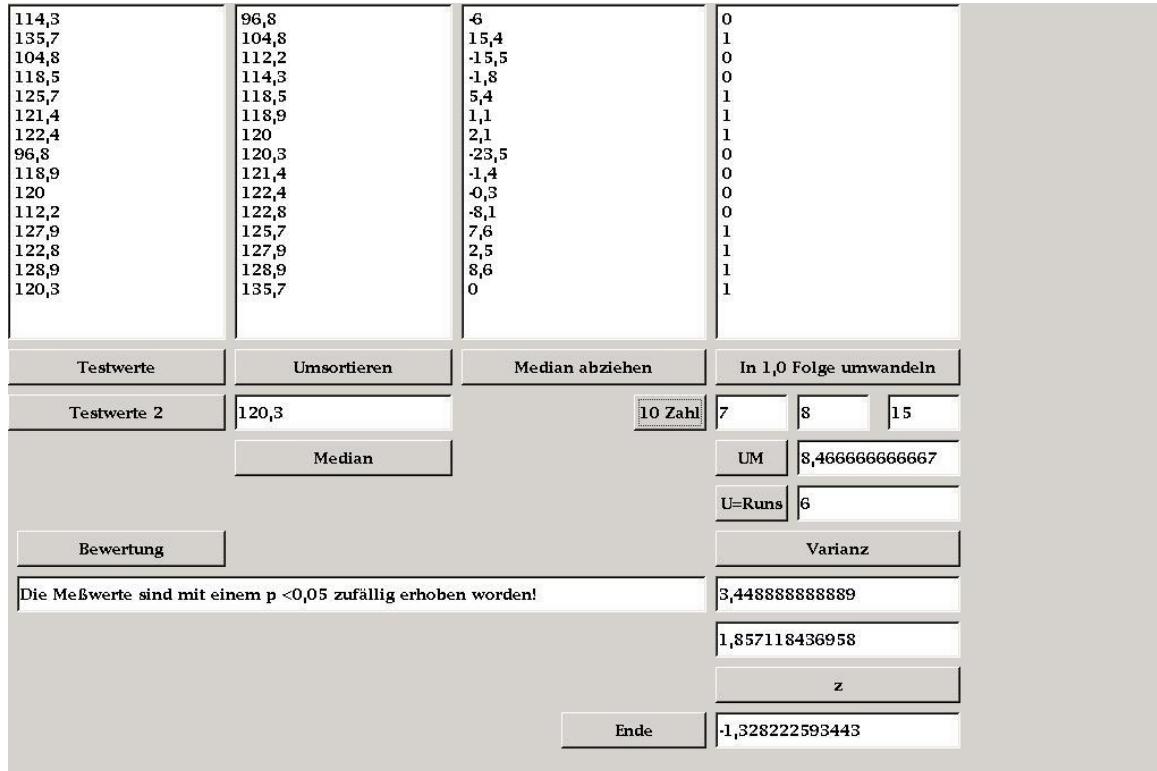


Fig.48

Il codice

```

PUBLIC SUB Form_Open ()
    ME.Text = "Esegui runtest"
    'Il test di esecuzione è un test non parametrico
    'sulla stazionarietà o non correlazione
    'IN una serie temporale o altra sequenza.
    'L'ipotesi nulla da scartare è qui,
    'che i valori consecutivi non sono correlati. "Una sequenza" è
    'definita come una sequenza di "simboli identici IN una sequenza di
    'simboli" con un alfabeto finito.

    "Prima una serie numerica" deve essere generata da una sequenza di simboli
    «Per prima cosa viene calcolata la mediana della serie
    'e dedotto da tutti i valori.
    "NELLA riga risultante, i numeri negativi" sono sostituiti da un "-" e i
    numeri positivi da un "+". 'Per le dimensioni del campione n1, n2>
    10
    '(con n1, n2 uguale al numero di "più" e "meno")

```

'la distribuzione di probabilità delle corse approssimativamente normale' con il valore
 medio UM = $2 * n1*n2 / (n1 + n2) + 1$
 'e varianza

'varianza = $2*n1*n2*(2*n1*n2 - n1 - n2) / (n1 + n2) / (n1 + n2) / (n1 + n2 - 1)$

'La variabile di test z è ora calcolata in base a:

'z = $(U - UM) / \text{radice della varianza}$

'dove U è il numero misurato di "esecuzioni" da "-" "e" + "nella serie.

'z è più o meno normale normale.

'Esempio

'13 3 14 14 1 14 3 8 14 17 9 14 13 2 16 1 3 12 13 14

'13

'3

'14

'14

«1

'14

'3

'8th

'14

'17

'9

'14

'13

'2

'16

«1

'3

'12

'13

'14

'Mediana 13 Stampa dalla serie originale

'0-10 1 1-12 1-10-5 1 4-4 1 0-11 3-12-10-1 0 1

'Conversione in + e -

'+ - + - + - + - + + - + - + + + + +

«1 2 34 5 6 7 ° 8 9101112

13 °

'U è il numero di run

'U = 13 run o U: 13

'n1 = segno più' + ': 11

'n2 = segno meno' - ': 9

'UM = $(2*11*9) / (11 + 9) + 1 = 10,9$

'varianza' = ... = 4,6 ?????

'U = 2,1

'z = $(13-10,9) / 2,1 = 1,0$

Decisione: l'ipotesi non viene respinta.

"Gli elementi del campione sono stati probabilmente presi a caso.

'A un livello di significatività di 0,05, H0 è rifiutato se $|z| > 1,96$. Questo non è il caso.

Decisione: l'ipotesi non viene respinta. Gli elementi del campione sono stati probabilmente presi a caso.

'che la maggior parte degli z-score per le lunghezze 1, 2 e 3' hanno un
 valore assoluto maggiore di 1,96. 'Questa è una forte prova che questi
 dati lo siano

'IN realtà NON casuale.

'Guarda anche:

<http://www.itl.nist.gov/div898/handbook/eda/section3/eda35d.htm>

<http://www.wu-wien.ac.at/usr/stat2/melichar/spsspu1/spssk3-9.htm>

Bradley, (1968). Test statistici senza distribuzione, capitolo 12.

'

FINE

list AS String []

Pulsante PUBLIC SUB1_Click ()

Testo DIM AS String

```
DIM mediana AS float
Elenco DIM come stringa []
DIM post AS String
text = Textarea2.Text
list = Split (text, Chr (10))
'riordinare
"Determina la lunghezza e utilizza la formula corretta", a seconda
che la lunghezza sia pari o dispari. IF liste.Lunghezza MOD 2 POI

    PRINT liste.Length MOD 2 & "Odd"
    PRINT (lista.length + 1) / 2
    STAMPA elenco [(liste.length + 1) / 2 - 1]
    median = Val (liste [(liste.length + 1) / 2 - 1]) 'L'array inizia con
    l'elemento 0 non con 1! ALTRO

    PRINT liste.Length MOD 2 & "Even"
    median = (Val (liste [liste.length / 2 - 1]) +
    Val (liste [liste.length / 2])) / 2
    'L'array inizia con l'elemento 0 non con 1!
    FINISCI SE
    textbox1.Text = Str (mediana)
    Button4.setfocus
    CATTURARE
    IF textarea1.Text = "" THEN message.info ("Devi essere nel campo di testo
1 inserisci un elenco di valori ")
    IF textarea2.Text = "" THEN message.info ("Devi prima premere riordina")
    RITORNO
    FINE
    PUBLIC SUB Button2_Click ()
'riordinare
c AS intero
j AS intero
n AS intero
y AS variante
Elenco DIM come stringa []
elemento AS String
txt come stringa
testo come stringa
text = Textarea1.Text
list = Split (text, Chr (10))
y = 0
n = liste.length
RIPETERE
c = 0
PER j = 0 TO n - 2
    'PRINT j, y, list [0], ar [1], ar [2], ar [3], ar [4]
    IF Val (list [j]) > Val (list [j + 1]) THEN
        y = Val (elenco [j])
        list [j] = list [j + 1]
        elenco [j + 1] = Str (y)
        c = 1
    FINISCI SE
    IL PROSSIMO
FINO A c = 0
PER OGNI elemento IN lista
txt = txt & Str (elemento) & Chr (10)
IL PROSSIMO
PRINT txt
textarea2.Text = ""
txt = Mid $ (txt, 1, -1)
«L'ultima interruzione di riga deve sparire
textarea2.Text = txt
Button1.setfocus
FINE
Pulsante PUBLIC SUB 3_Click ()
```

```

textarea1.Text = "114,3"
textarea1.Text = textarea1.Text & Chr (10) & "135.7" textarea1.Text
= textarea1.Text & Chr (10) & "104.8" textarea1.Text = textarea1.Text
& Chr (10) & "118.5" textarea1.Text = textarea1.Text & Chr (10) &
"125.7" textarea1.Text = textarea1.Text & Chr (10) & "121.4"
textarea1.Text = textarea1.Text & Chr (10) & "122.4" textarea1.Text =
textarea1.Text & Chr (10) & "96.8" textarea1.Text = textarea1.Text &
Chr (10) & "118.9" textarea1.Text = textarea1.Text & Chr (10) & "120"
textarea1.Text = textarea1.Text & Chr (10) & "112,2" textarea1.Text =
textarea1.Text & Chr (10) & "127,9" textarea1.Text = textarea1.Text &
Chr (10) & "122.8" textarea1.Text = textarea1.Text & Chr (10) &
"128.9" textarea1.Text = textarea1.Text & Chr (10) & "120.3" 'mw =
119.373

'varianza = 92,6016
'sigma = 9,62297
'mediano
Button2.setfocus
FINE
Pulsante PUBLIC SUB 4_Click ()
'Sotrai la mediana
m COME galleggiante
d AS float
n AS intero
y AS variante
Elenco DIM come stringa []
elemento AS String
txt come stringa
testo come stringa
text = Textarea1.Text
m = Val (textbox1.Text)
list = Split (text, Chr (10))
y = 0
n = liste.length
PER OGNI elemento IN lista
d = Val (elemento) - m
'PRINT elemento, m, d
txt = txt & Str (d) & Chr (10)
IL PROSSIMO
PRINT txt
textarea3.Text = ""
txt = Mid $ (txt, 1, -1)
«L'ultima interruzione di riga deve sparire
textarea3.Text = txt
    Button6.setfocus
FINE
Pulsante PUBLIC SUB 5_Click ()
'converti in sequenza 1.0
e AS intero
Elenco DIM come stringa []
elemento AS String
txt come stringa
testo come stringa
text = Textarea3.Text
list = Split (text, Chr (10))
'n = liste.length
PER OGNI elemento IN lista
SE Val (elemento) <0 ALLORA
e = 0
ALTRO
e = 1
FINISCI SE
txt = txt & Str (e) & Chr (10)
IL PROSSIMO

```

Contenuti

```
PRINT txt
textarea4.Text = ""
txt = Mid $(txt, 1, -1)
«L'ultima interruzione di riga deve sparire
textarea4.Text = txt
FINE
Pulsante PUBLIC SUB 6_Click ()
DIM s AS String
DIM n AS intero
DIM T AS stringa
Carattere DIM come stringa
Contatore DIM AS intero
DIM izaehler AS Integer
Elenco DIM come stringa []
elemento AS String
txt come stringa
testo come stringa
text = Textarea4.Text
list = Split (text, Chr (10))
n = liste.length
count = 0
izaehler = 0
PER OGNI elemento IN lista
IF element = "0" THEN count = count + 1
IF element = "1" THEN izaehler = izaehler + 1
IL PROSSIMO
textbox2.Text = ozaehler
textbox3.Text = izaehler
textbox4.Text = ozaehler + izaehler
FINE
Pulsante PUBLIC SUB 11_Click ()
    ME.Chiudi
FINE
Pulsante PUBLIC SUB 7_Click ()
UM AS galleggiante
n1 AS intero
n2 AS intero
n1 = Val (textbox2.Text)
n2 = Val (textbox3.Text)
UM = 2 * n1 * n2 / (n1 + n2) + 1
textbox5.Text = Str (UM)
FINE
Pulsante PUBLIC SUB 8_Click ()
DIM s AS String
DIM n AS intero
DIM T AS stringa
Carattere DIM come stringa
charnext AS stringa
Contatore di esecuzione DIM AS intero
Elenco DIM come stringa []
    elemento AS String
txt come stringa
testo come stringa
text = Textarea4.Text
list = Split (text, Chr (10))
n = liste.length
PER OGNI elemento IN lista
s = s & elemento
IL PROSSIMO
errore di rughe = 1
    T = Trim (s)
    PER n = 1 TO Len (T) - 1
        char = Mid $(T, n, 1)
        charnext = Mid $(T, n + 1, 1)
            IF char <> charnext THEN runzehler = runzehler + 1
IL PROSSIMO
```

```

textbox6.Text = errore di rughe
FINE
Tasto PUBLIC SUB 9_Click ()
'varianza = 2*n1*n2*(2*n1*n2 - n1 - n2) / (n1 + n2) / (n1 + n2)
varianza AS Float
n1 AS intero
n2 AS intero
due AS interi
somma AS intero
esegui AS intero
n1 = Val (textbox2.Text)
n2 = Val (textbox3.Text)
run = Val (textbox6.text)
due = 2 * n1 * n2
somma = n1 + n2
STAMPA due, somma, due * (due - somma),
varianza = due *(due - somma) / (somma * totale * (somma - 1))
'varianza = (run * (corsa - 2)) / (4 * (corsa - 1))
textbox7.Text = Str (varianza)
SE varianza <0 ALLORA varianza = -varianza
textbox8.Text = Str (Sqr (varianza))
FINE
Pulsante PUBLIC SUB 10_Click ()
z AS float
eseguire AS float
UM AS galleggiante
svar AS float
run = Val (textbox6.text)
UM = Val (textbox5.text)
svar = Val (textbox8.text)
z = (esegui - UM) / svar
textbox9.Text = Str (z)
FINE
Pulsante PUBLIC SUB 12_Click ()
z AS float
t1 come stringa
t2 come stringa
t1 = "I valori misurati sono stati presi a caso con un p <0,05!" t2 = "I valori misurati NON
sono stati presi in modo casuale con un p <0,05!"

z = Val (textbox9.Text)
SE z > 1,96 XOR z <-1,96 ALLORA
textbox10.text = t2
ALTRO
textbox10.text = t1
FINISCI SE
FINE
Tasto PUBLIC SUB 13_Click ()
'qualche altro esempio di valori, certamente non a caso
textarea1.Text = "1"
textarea1.Text = textarea1.Text & Chr (10) & "2"
textarea1.Text = textarea1.Text & Chr (10) & "3"
textarea1.Text = textarea1.Text & Chr (10) & "4"
textarea1.Text = textarea1.Text & Chr (10) & "5"
textarea1.Text = textarea1.Text & Chr (10) & "6"
textarea1.Text = textarea1.Text & Chr (10) & "7"
textarea1.Text = textarea1.Text & Chr (10) & "8"
textarea1.Text = textarea1.Text & Chr (10) & "9"
textarea1.Text = textarea1.Text & Chr (10) & "10"
textarea1.Text = textarea1.Text & Chr (10) & "11"
textarea1.Text = textarea1.Text & Chr (10) & "12"
textarea1.Text = textarea1.Text & Chr (10) & "13"
textarea1.Text = textarea1.Text & Chr (10) & "14"
textarea1.Text = textarea1.Text & Chr (10) & "15"
FINE

```

Semplificazioni

- Gli esempi possono essere inseriti molto più velocemente con il comando array.
- L'ordinamento può essere notevolmente semplificato con il comando array Array.sort.
- Il tutto dovrebbe funzionare con la semplice pressione di un pulsante.
- I comandi set focus dovrebbero essere eseguiti nell'ordine corretto
- Le subroutine per le singole sezioni sarebbero più chiare.

Il tutto funziona meno soggetto a errori quando tutto accade con la semplice pressione di un pulsante. Tuttavia si dovrebbe lasciare in piedi la versione di cui sopra, che procede passo dopo passo.

0.134.3 Esegui test 2

Il run test è stato nuovamente programmato in una forma un po 'più semplice:

Devi

- 9 pulsanti di comando
- 9 caselle di testo
- 1 area di testo

Sono state integrate 2 semplici sequenze di esempio.

Inserzione:

```
PUBLIC SUB Form_Open ()  
'Il test di esecuzione è un test non parametrico  
'sulla stazionarietà o non correlazione  
'IN una serie temporale o altra sequenza.  
'L'ipotesi nulla da scartare è qui,  
'che i valori consecutivi non sono correlati. "Una sequenza" è  
definita come una sequenza di "simboli identici IN una sequenza di  
simboli" con un alfabeto finito.  
  
"Prima una serie numerica" deve essere generata da una sequenza di simboli  
«Per prima cosa viene calcolata la mediana della serie  
'e dedotto da tutti i valori.  
"NELLA riga risultante, i numeri negativi" sono sostituiti da un "-" e i  
numeri positivi da un "+". 'Per le dimensioni del campione n1, n2>  
10  
'(con n1, n2 uguale al numero di "più" e "meno")' la distribuzione di probabilità  
delle corse approssimativamente normale 'con il valore medio UM = 2 * n1*n2 /  
(n1 + n2) + 1  
'e varianza  
  
'varianza = 2*n1*n2*(2*n1*n2 - n1 - n2) / (n1 + n2) / (n1 + n2) / (n1 + n2 - 1)  
'La variabile di test z è ora calcolata in base a:  
'z = (U - UM) / radice della varianza  
'dove U è il numero misurato di "esecuzioni" da "-" "e" + "nella  
serie.  
'z è più o meno normale normale.  
'Esempio  
'13 3 14 14 1 14 3 8 14 17 9 14 13 2 16 1 3 12 13 14  
'Mediana 13      Stampa dalla serie originale  
'0-10 1 1-12 1-10-5 1 4-4 1 0-11 3-12-10-1 0 1  
'Conversione in + e -  
'+-+-+-+---++-+--++-
```

```

«1 2 3      4 5 6  7 °  8 °  9      101112   13 °
'U è il numero di run
'U = 13 run o U: 13
'n1      = Segno più "+": 11
'n2      = Segno meno   "-": 9
'UM = (2*11*9) / (11 + 9) + 1 = 10,9
'varianza' = ... = 4,6 ?????
'U = 2,1
'z = (13-10,9) / 2,1 = 1,0
Decisione: l'ipotesi non viene respinta.
"Gli elementi del campione sono stati probabilmente presi a caso.

'che la maggior parte degli z-score per le lunghezze 1, 2 e 3' hanno un
valore assoluto maggiore di 1,96. 'Questa è una forte prova che questi
dati lo siano

'IN realtà NON casuale.
'Guarda anche:
http://www.itl.nist.gov/div898/handbook/eda/section3/eda35d.htm
http://www.wu-wien.ac.at/usr/stat2/melichar/spsspu1/spssk3-9.htm
Bradley, (1968). Test statistici senza distribuzione, capitolo 12.
'

FINE

```

```

Pulsante PUBLIC SUB1_Click ()
ME.CLOSE
FINE

```

```

PUBLIC SUB Button2_Click ()
DIM s AS String
DIM n AS intero
DIM T AS stringa
Carattere DIM come stringa
Contatore DIM AS intero
DIM izaehler AS Integer
s = textarea1.Text
count = 0
izaehler = 0
T = Trim (s)
PER n = 1 TO Len (T)
    char = Mid $ (T, n, 1)
    SELEZIONA CASE TRUE
        CASE char = "0"
            count = count + 1
        CASE char = "1"
            izaehler = izaehler + 1
    FINE SELEZIONA
    IL PROSSIMO
textbox1.Text = ocount
textbox2.Text = izaehler
textbox3.Text = ozaehler + izaehler
FINE

```

```

Pulsante PUBLIC SUB 3_Click ()
" lanci multipli di monete per generare una sequenza casuale di 0 e 1
DIM un numero intero AS
DIM t AS stringa
DIM M AS Inter
Rendi casuale ()
PER a = DA 1 A 200
    M = Int (2 * Rnd)
    'Crea un tiro in mezzo
    t = t & LTrim $ (Str $ (M))
    0 e 1
IL PROSSIMO

```

Contenuti

```
Textarea1.Text = t  
FINE
```

```
Pulsante PUBLIC SUB 4_Click ()  
UM AS galleggiante  
n1 AS intero  
n2 AS intero  
n1 = Val (textbox1.Text)  
n2 = Val (textbox2.Text)  
UM = 2 * n1 * n2 / (n1 + n2) + 1  
'UM = 2 * n1*n2 / (n1 + n2) + 1  
'Media della distribuzione di probabilità delle corse  
textbox4.Text = Str (UM)  
FINE
```

```
Pulsante PUBLIC SUB 6_Click ()  
'varianza = 2*n1*n2*(2*n1*n2 - n1 - n2) / (n1 + n2) / (n1 + n2)  
'Varianza della distribuzione di probabilità delle serie  
varianza AS Float  
n1 AS intero  
n2 AS intero  
due AS interi  
somma AS intero  
esegui AS intero  
n1 = Val (textbox1.Text)  
n2 = Val (textbox2.Text)  
run = Val (textbox5.text)  
due = 2 * n1 * n2  
somma = n1 + n2  
STAMPA due, somma, due * (due - somma),  
varianza = due *(due - somma) / (somma * totale * (somma - 1))  
'varianza = (run * (corsa - 2)) / (4 * (corsa - 1))  
textbox6.Text = Str (varianza)  
SE varianza <0 ALLORA varianza = -varianza  
textbox7.Text = Str (Sqr (varianza))  
FINE
```

```
Pulsante PUBLIC SUB 7_Click ()  
'+ - + + - + - + + - + + - + - + +  
Textarea1.Text = "10110100110110100011"  
FINE
```

```
Pulsante PUBLIC SUB 5_Click ()  
DIM s AS String  
DIM n AS intero  
DIM T AS stringa  
Carattere DIM come stringa  
charnext AS stringa  
Contatore di esecuzione DIM AS intero  
s = textarea1.Text  
errore di rughe = 1  
T = Trim (s)  
PER n = 1 TO Len (T) - 1  
    char = Mid $ (T, n, 1)  
    charnext = Mid $ (T, n + 1, 1)  
    IF char <> charnext THEN runzehler = runzehler + 1  
IL PROSSIMO  
textbox5.Text = errore di rughe  
FINE
```

```
Pulsante PUBLIC SUB 8_Click ()  
z AS float  
eseguire AS float  
UM AS galleggiante  
svar AS float  
run = Val (textbox5.text)  
UM = Val (textbox4.text)  
svar = Val (textbox7.text)  
z = (esegui - UM) / svar  
'z è la variabile di test  
textbox8.Text = Str (z)  
FINE
```

```
Tasto PUBLIC SUB 9_Click ()  
z AS float  
t1 come stringa  
t2 come stringa  
t1 = "I valori misurati sono stati presi a caso con un p <0,05!" t2 = "I valori misurati NON  
sono stati presi in modo casuale con un p <0,05!"  
  
z = Val (textbox8.Text)  
SE z> 1,96 XOR z <-1,96 ALLORA  
textbox9.text = t2  
ALTRO  
textbox9.text = t1  
FINISCI SE  
FINE
```

Illustrazione della sequenza del programma per l'episodio 01 Chaitin A

011011001101111000010		
1,0,Zahl	U = Zahl der Runs	Muenze
9	11	wikizb
11	Varianz	
20	4,637368421053	
UM	2,153454996291	z
10,9	0,0464370048	
Bewertung		Ende
Die Meßwerte sind mit einem p <0,05 zufällig erhoben worden!		

Fig.49

0.134.4 Esegui test 3

Rispetto a Runtest 2, il programma è stato programmato in modo ancora più semplice. Il test ora viene eseguito con un clic. 3 esempi sono stati integrati.

Devi

- 5 pulsanti di comando
- 9 caselle di testo
- 1 area di testo

Sono state integrate 3 semplici sequenze di esempio.

Codice:

```
PUBLIC SUB Form_Open ()
ME.Text = "Runtest of coin flips"
FINE
```

```

Pulsante PUBLIC SUB1_Click ()
s come stringa
Episodio '10 dall'area di testo
n AS intero
'Esegui variabile
T AS stringa
'ritagliato 10 episodi da p
char AS stringa
'posizione unica della 10a serie
oz AS Integer
'Numero di zeri
iz AS intero
'Numero di uno
UM AS galleggiante
'Media della distribuzione di probabilità delle corse
esegue AS Integer
'Numero di esecuzioni
charnext AS stringa
'posizione successiva in sequenza
varianza AS Float
'Varianza della distribuzione di probabilità delle serie
due AS interi
somma AS intero
'Variabili ausiliarie per il calcolo della varianza
svar AS float
'Radice della varianza
z AS float
'Cifra di controllo
t1 come stringa
t2 come stringa
'Testi di valutazione
'----- 0 e 1, calcolo UM -----
s = textarea1.Text
oz = 0
iz = 0
T = Trim (s)
    PER n = 1 TO Len (T)
        char = Mid $ (T, n, 1)
        SELEZIONA CASE TRUE
            CASE char = "0"
                oz = oz + 1
            CASE char = "1"
                iz = iz + 1
        FINE SELEZIONA
    IL PROSSIMO
    textbox1.Text = oz
    textbox2.Text = iz
    textbox3.Text = oz + iz
    UM = 2 * oz * iz / (oz + iz) + 1
    'Media della distribuzione di probabilità delle corse
    textbox4.Text = Str (UM)
    '--- Esegue -----
    corre = 1
        PER n = 1 TO Len (T) - 1
            char = Mid $ (T, n, 1)
            charnext = Mid $ (T, n + 1, 1)
            IF char <> charnext THEN run = run + 1
    IL PROSSIMO
    textbox5.Text = viene eseguito
    '----- Varianza -----
    'varianza = 2*n1*n2*(2*n1*n2 - n1 - n2) / (n1 + n2) / (n1 + n2)
    'Varianza della distribuzione di probabilità delle serie
    due = 2 * oz * iz
    somma = oz + iz
    'STAMPA due, somma, due * (due - somma),
    varianza = due *(due - somma) / (somma * totale * (somma - 1))

```

```
textbox6.Text = Str (varianza)
SE varianza <0 ALLORA varianza = -varianza
textbox7.Text = Str (Sqr (varianza))
svar = Sqr (varianza)
z = (corre - UM) / svar
'z è la variabile di test
textbox8.Text = Str (z)
'----- Valutazione-----
t1 = "Il I valori misurati sono stati presi a caso con un p <0,05! "
t2 = "Il I valori misurati con un p <0,05 NON vengono presi in modo casuale
stato! "
SE z> 1,96 XOR z <-1,96 ALLORA
textbox9.text = t2
ALTRO
textbox9.text = t1
FINISCI SE
FINE
PUBLIC SUB Button2_Click ()
'+ - + + - + - + + - + - + -
Textarea1.Text = "10110100110110100011"
FINE
Pulsante PUBLIC SUB 3_Click ()
" lanci multipli di monete per generare una sequenza casuale di 0 e 1
DIM un numero intero AS
DIM t AS stringa
DIM M AS Intero
Rendi casuale ()
PER a = DA 1 A 200
    M = Int (2 * Rnd)
    'Crea un tiro in mezzo          0 e 1
    t = t & LTrim $ (Str $ (M))
IL PROSSIMO
Textarea1.Text = t
FINE
Pulsante PUBLIC SUB 4_Click ()
ME.CLOSE
FINE
Pulsante PUBLIC SUB 5_Click ()
Textarea1.Text = "10101010101010101010"
FINE
```

0,135 entropia per sequenze 01 semplici

L'entropia è un termine molto usato ma poco compreso. Il termine è ancora poco utilizzato nelle statistiche, anche se potrebbe effettivamente essere un termine di base.

- Per favore, riferisci [http://de.wikipedia.org/wiki/Entropie_\(Informationstheorie\)](http://de.wikipedia.org/wiki/Entropie_(Informationstheorie))
- Per favore, riferisci <http://www.madeeasy.de/2/entropie.htm>
- Per favore, riferisci <http://www.madeeasy.de/2/zufallgz.htm>

Di seguito sono elencati e descritti alcuni semplici programmi Gambas, che dovrebbero elaborare passo dopo passo il calcolo dell'entropia di semplici sequenze 01.

0.135.1 Esercizio preliminare 1

Nel seguente programma sono elencate sistematicamente tutte le possibilità di una sequenza 01 sempre più lunga. Tutto ciò di cui hai bisogno è un modulo vuoto per avviare il programma. Il

L'output avviene nella finestra diretta.

```
PUBLIC SUB Form_Open ()
'Questo programma conta da 1 a 2^8 nel sistema binario
'Gli spazi iniziali sono riempiti con zero.
s AS intero
z AS intero
t AS stringa
PER s = da 1 a 8
    PER z = 0 TO (2^s - 1)
        t = Bin $(z, s)
        STAMPA t
    IL PROSSIMO
IL PROSSIMO
FINE
```

Produzione:

```
0
1
00
01
10
11
000
001
010
011
100
101
110
111
0000
0001
0010
0011
0100
```

0.135.2 Esercizio preliminare 2

In un secondo esercizio preliminare, i valori di entropia vengono assegnati a tutti i numeri binari, che sono approssimativamente semplificati (e quindi ancora calcolati in modo abbastanza errato). Una sequenza 01 completamente ordinata ha sempre il valore di entropia 0. Una sequenza 01 completamente casuale ha un valore di entropia che corrisponde alla sua lunghezza.

Esempio:

- 000000000000000000000000 entropia = 0
- 01101100110111100010 entropia = 20

Attenzione, i valori di entropia calcolati sono ancora abbastanza sbagliati !!

```
' File di classe Gambas
t AS String
PUBLIC SUB Form_Open ()
s AS Integer
```

```
z AS intero
n AS intero
e AS intero
n = 0
PER s = da 1 a 10
    PER z = 0 TO (2*s - 1)
        t = Bin $(z, s)
        STAMPA "n, t" & n, t;
        e = ent (t)
        STAMPA "E =" ed e
        n = n + 1
    IL PROSSIMO
IL PROSSIMO
FINE
FUNZIONE PUBBLICA ent (e AS Integer) AS Integer
oz AS Integer
iz AS intero
gzAS intero
char AS stringa
te AS stringa
n AS intero
oz = 0
iz = 0
gz = 0
te = Trim (t)
    PER n = 1 TO Len (te)
        char = Mid $(te, n, 1)
        SELEZIONA CASE TRUE
            CASE char = "0"
                oz = oz + 1
            CASE char = "1"
                iz = iz + 1
        FINE SELEZIONA
        gz = oz + iz
    IL PROSSIMO
    STAMPA "gz, oz, iz" & gz, oz, iz;
    SE oz> iz ALLORA e = gz - oz
    «SE oz> 0,6*iz THEN e = gz - oz Else runstest
    SE oz <iz ALLORA e = oz
    «SE oz <0,4*iz THEN e = oz Else runstest
    SE oz = iz ALLORA e = gz
    'IF oz = iz THEN runstest
RITORNO e
FINE
```

0.135.3 10 Statistiche

Il problema di distinguere tra sequenze 01 casuali e non casuali può anche essere affrontato in modo diverso. Il seguente programma 01Statistik mostra una possibilità.

Il programma può essere facilmente testato utilizzando gli esempi che contiene.

Qualsiasi sequenza 01 può essere copiata nell'area di testo e valutata. Sarà solo

Vengono prese in considerazione le corse fino a una lunghezza massima di 10.

Hai bisogno

- molti pulsanti di comando
- molte caselle di testo
- un'area di testo

per avviare il programma.

Guarda anche: <http://www.madeeasy.de/2/prg01st.htm> (Sempre in VB, il layout è rimasto lo stesso)

```
' File di classe Gambas
t AS String
PUBLIC SUB Button9_Click ()
    ME.Close
FINE
PUBLIC SUB Button2_Click ()
'questa funzione dovrebbe esaminare un file 01' dovrebbe
restituire il numero di 00 coppie '0, 0, oo numero di altre
coppie
DIM s AS String DIM n
AS Intero DIM T AS
String DIM char AS
String
DIM oozaehler AS Integer DIM
oizaehler AS Integer DIM
iozaehler AS Integer DIM
iiizaehler AS Integer s =
textarea1.Text oozaehler = 0
oizaehler = 0 iozaehler = 0
iiizaehler = 0

T = Trim (s)
FOR n = 1 TO Len (T) PASSO 2
    char = Mid $ (T, n, 2)
    SELEZIONA CASE TRUE
        CASE char = "00"
            oo contatore = oo contatore + 1
        CASE char = "01"
            oizaehler = oizaehler + 1
        CASE char = "10"
            contatore io = contatore io + 1
        CASE char = "11"
            iizaehler = iizaehler + 1
    FINE SELEZIONA
    IL PROSSIMO
textbox4.Text = oozaehler
textbox5.Text = oizaehler
textbox6.Text = io counter
textbox7.Text = iiizaehler
FINE
Pulsante PUBLIC SUB1_Click ()
'Quanti 1 e 0 ci sono nella sequenza?
DIM s AS String
DIM n AS intero
DIM T AS stringa
Carattere DIM come stringa
Contatore DIM AS intero
DIM izaehler AS Integer
s = textarea1.Text
count = 0
izaehler = 0
T = Trim (s)
PER n = 1 TO Len (T)
    char = Mid $ (T, n, 1)
    SELEZIONA CASE TRUE
        CASE char = "0"
            count = count + 1
        CASE char = "1"
            izaehler = izaehler + 1
    FINE SELEZIONA
```

```
IL PROSSIMO
textbox1.Text = ocount
textbox2.Text = izaehter
textbox3.Text = ozaehler + izaehter
FINE
Pulsante PUBLIC SUB 6_Click ()
" lanci multipli di monete per generare una sequenza casuale di 0 e 1
DIM un numero intero AS
DIM t AS stringa
DIM M AS Inter
Rendi casuale ()
PER a = 1 FINO A 20000
    M = Int (2 * Rnd)
    'Crea un tiro in mezzo          0 e 1
    t = t & LTrim $ (Str $ (M))
IL PROSSIMO
Textarea1.Text = t
FINE
Pulsante PUBLIC SUB 8_Click ()
M AS stringa
t AS stringa
un numero intero AS
PER a = 1 A 10000
    M = "10"
    t = t & LTrim $ (Str $ (M))
IL PROSSIMO
Textarea1.Text = t
FINE
Pulsante PUBLIC SUB 4_Click ()
DIM s AS String
DIM n AS intero
DIM T AS stringa
DIM Tempo AS String
Carattere DIM come stringa
un numero intero AS
DIM o1 AS Integer
DIM o2 AS intero
DIM o3 AS intero
DIM o4 AS intero
DIM o5 AS intero
DIM o6 AS Integer
DIM o7 AS Integer
DIM o8 AS intero
DIM o9 AS intero
DIM oa AS Integer
s = textarea1.Text
    T = Trim (s)
' 'Rem questa funzione dovrebbe esaminare un file 01
' 'Rem Dovresti restituire il numero di 000,0000,0000 ecc. (Le corse)

o1 = 0
o2 = 0
o3 = 0
o4 = 0
o5 = 0
o6 = 0
o7 = 0
o8 = 0
o9 = 0
oa = 0
' 'Rem 0000000001000000000010000000000
' Cls
Tempo = RTrim $ (LTrim $ (s))
' 'Rem a = 10 a
= 10
' 'Rem PRINT Mid $ (Temp, 1, 11)
```

```

IF Mid $ (Tempo, 1, a + 1) = "0000000001" THEN oa = oa + 1
' 'Rem PRINT Mid $ (Temp, Len (Temp) - 10, 11) IF Len
(Tempo) <a THEN GOTO a9
IF Len (Tempo) = a AND s = "0000000000" ALLORA oa = oa + 1 IF Len
(Tempo) = a THEN GOTO a9
IF Mid $ (Tempo, Len (Tempo) - 10, 11) = "1000000000" THEN oa = oa + 1 FOR n = 1 TO
Len (Tempo)
IF Mid $ (Tempo, n, a + 2) = "100000000001" THEN oa = oa + 1 'Rem PRINT
Mid $ (Tempo, n, a)
IL PROSSIMO

a9:
' 'Rem a = 9
a = 9
'Rem 0000000010000000001000000000
" Rem PRINT Mid $ (Temp, 1, 10)
IF Mid $ (Tempo, 1, 10) = "0000000001" THEN o9 = o9 + 1 'Rem
PRINT Mid $ (Temp, Len (Temp) - 9, 10) IF Len (Tempo) <a THEN
GOTO a8
IF Len (Tempo) = a AND s = "000000000" ALLORA o9 = o9 + 1 IF Len
(Tempo) = a THEN GOTO a8
IF Mid $ (Tempo, Len (Tempo) - 9, 10) = "1000000000" THEN o9 = o9 + 1 FOR n = 1 TO
Len (Tempo)
IF Mid $ (Tempo, n, a + 2) = "100000000001" ALLORA o9 = o9 + 1
' 'Rem PRINT Mid $ (Temp, n, a)
IL PROSSIMO

a8:
' 'Rem a = 8
a = 8
'Rem 0000000010000000001000000000
' 'Rem PRINT Mid $ (Temp, 1, a + 1)
IF Mid $ (Tempo, 1, a + 1) = "0000000001" THEN o8 = o8 + 1 'Rem PRINT
Mid $ (Temp, Len (Temp) - a, a + 1)
IF Len (Tempo) <a THEN GOTO a7
IF Len (Tempo) = a AND s = "000000000" THEN o8 = o8 + 1 IF Len
(Tempo) = a THEN GOTO a7 FOR n = 1 TO Len (Tempo)

IF Mid $ (Tempo, n, a + 2) = "100000000001" ALLORA o8 = o8 + 1
' 'Rem PRINT Mid $ (Temp, n, a)
IL PROSSIMO

a7:
' 'Rem a = 7
a = 7
" Rem 000000001000000000100000000
IF Mid $ (Tempo, 1, a + 1) = "000000001" THEN o7 = o7 + 1 IF Len
(Tempo) <a THEN GOTO a6
IF Len (Tempo) = a AND s = "000000000" THEN o7 = o7 + 1 IF Len
(Tempo) = a THEN GOTO a6
IF Mid $ (Tempo, Len (Tempo) - a, a + 1) = "100000000" THEN o7 = o7 + 1 FOR n = 1 TO
Len (Tempo)
IF Mid $ (Tempo, n, a + 2) = "100000000001" ALLORA o7 = o7 + 1
IL PROSSIMO

a6:
' 'Rem a = 6 a
= 6
'Rem 000000100000001000000
IF Mid $ (Tempo, 1, a + 1) = "00000001" THEN o6 = o6 + 1 IF Len
(Tempo) <a THEN GOTO a5
IF Len (Tempo) = a AND s = "00000000" THEN o6 = o6 + 1 IF Len
(Tempo) = a THEN GOTO a5
IF Mid $ (Tempo, Len (Tempo) - a, a + 1) = "100000000" THEN o6 = o6 + 1 FOR n = 1 TO
Len (Tempo)
IF Mid $ (Tempo, n, a + 2) = "100000000001" ALLORA o6 = o6 + 1

```

```
IL PROSSIMO
'
a5:
' 'Rem a = 5 a
= 5
' 'Rem 00000100000100000
IF Mid $ (Tempo, 1, a + 1) = "000001" THEN o5 = o5 + 1 IF Len
(Tempo) <a THEN GOTO a4
IF Len (Tempo) = a AND s = "00000" THEN o5 = o5 + 1 IF Len
(Tempo) = a THEN GOTO a4
IF Mid $ (Tempo, Len (Tempo) - a, a + 1) = "100000" THEN o5 = o5 + 1 FOR n = 1 TO
Len (Tempo)
IF Mid $ (Tempo, n, a + 2) = "1000001" ALLORA o5 = o5 + 1
IL PROSSIMO
'

a4:
' 'Rem a = 4 a
= 4
' 'Rem 00001000010000
IF Mid $ (Tempo, 1, a + 1) = "00001" THEN o4 = o4 + 1 IF Len
(Tempo) <a THEN GOTO a3
IF Len (Tempo) = a AND s = "0000" THEN o4 = o4 + 1 IF Len
(Tempo) = a THEN GOTO a3
IF Mid $ (Tempo, Len (Tempo) - a, a + 1) = "10000" THEN o4 = o4 + 1 FOR n = 1 TO
Len (Tempo)
IF Mid $ (Tempo, n, a + 2) = "100001" ALLORA o4 = o4 + 1
IL PROSSIMO
a3:
a = 3
" Rem 00010001000
IF Mid $ (Tempo, 1, a + 1) = "0001" THEN o3 = o3 + 1 IF Len
(Tempo) <a THEN GOTO a2
IF Len (Tempo) = a AND s = "000" THEN o3 = o3 + 1 IF Len
(Tempo) = a THEN GOTO a2
IF Mid $ (Tempo, Len (Tempo) - a, a + 1) = "1000" THEN o3 = o3 + 1 FOR n = 1 TO
Len (Tempo)
IF Mid $ (Tempo, n, a + 2) = "10001" ALLORA o3 = o3 + 1
IL PROSSIMO
'

a2:
a = 2
" Rem 00100100
IF Mid $ (Tempo, 1, a + 1) = "001" THEN o2 = o2 + 1 IF Len
(Tempo) <a THEN GOTO aeins
IF Len (Tempo) = a AND s = "00" THEN o2 = o2 + 1
IF Len (Tempo) = a THEN GOTO aeins
IF Mid $ (Tempo, Len (Tempo) - a, a + 1) = "100" THEN o2 = o2 + 1 FOR n = 1 TO
Len (Tempo)
IF Mid $ (Tempo, n, a + 2) = "1001" THEN o2 = o2 + 1
IL PROSSIMO
aeins:
a = 1
" Rem 01010
IF Mid $ (Tempo, 1, a + 1) = "01" THEN o1 = o1 + 1 IF Len
(Tempo) <a THEN GOTO a0
IF Mid $ (Tempo, Len (Tempo) - a, a + 1) = "10" THEN o1 = o1 + 1 FOR n = 1 TO
Len (Tempo)
IF Mid $ (Tempo, n, a + 2) = "101" THEN o1 = o1 + 1
IL PROSSIMO
'

a0:
textbox8.Text = o1
textbox9.Text = o2
textbox10.Text = o3
textbox11.Text = o4
textbox12.Text = o5
```

```

textbox13.Text = o6
textbox14.Text = o7
textbox15.Text = o8
textbox16.Text = o9
textbox17.Text = oa
FINE
Pulsante PUBLIC SUB 7_Click ()
'deve ancora essere creato
FINE
Pulsante PUBLIC SUB 5_Click ()
'questa funzione dovrebbe essere un file 01
'esaminare
'Dovrebbe restituire il numero di 11,111,1111,1111 corse ecc.
DIM s AS String
DIM n AS intero
DIM T AS stringa
DIM Tempo AS String
Carattere DIM come stringa
un numero intero AS
DIM i1 AS Integer
DIM i2 AS Integer
DIM i3 AS Integer
DIM i4 AS Integer
DIM i5 AS Integer
DIM i6 AS Integer
DIM i7 AS Integer
DIM i8 AS Integer
DIM i9 AS Integer
DIM ia AS Integer
s = textarea1.Text
T = Trim (s)
i1 = 0
i2 = 0
i3 = 0
i4 = 0
i5 = 0
i6 = 0
i7 = 0
i8 = 0
i9 = 0
ia = 0
Tempo = RTrim $ (LTrim $(s))
' 'Rem 111111110111111110111111111
' Cls
a = 10
" Rem PRINT Mid $ (Temp, 1, 11)
IF Mid $ (Tempo, 1, a + 1) = "1111111110" ALLORA ia = ia + 1
' 'Rem PRINT Mid $ (Temp, Len (Temp) - 10, 11) IF Len
(Tempo) <a THEN GOTO b9
IF Len (Tempo) = a AND s = "1111111111" ALLORA ia = ia + 1 IF Len
(Tempo) = a THEN GOTO b9
IF Mid $ (Tempo, Len (Tempo) - a, a + 1) = "0111111111" ALLORA ia = ia + 1

FOR n = 1 TO Len (Tempo)
IF Mid $ (Tempo, n, a + 2) = "0111111110" ALLORA ia = ia + 1
' 'Rem PRINT Mid $ (Temp, n, a)
IL PROSSIMO
b9:
a = 9
IF Mid $ (Tempo, 1, 10) = "1111111110" THEN i9 = i9 + 1 IF Len
(Tempo) <a THEN GOTO b8
IF Len (Tempo) = a AND s = "111111111" ALLORA i9 = i9 + 1 IF Len
(Tempo) = a POI ANDARE b8
IF Mid $ (Tempo, Len (Tempo) - 9, 10) = "0111111111" THEN i9 = i9 + 1 FOR n = 1 TO
Len (Tempo)
IF Mid $ (Tempo, n, a + 2) = "0111111110" ALLORA i9 = i9 + 1

```

```
IL PROSSIMO
b8:
a = 8
IF Mid $(Tempo, 1, a + 1) = "11111110" THEN i8 = i8 + 1 IF Len
(Tempo) <a THEN GOTO b7
IF Len $(Tempo) = a AND s = "11111111" ALLORA i8 = i8 + 1 IF Len
(Tempo) = a POI ANDARE b7
IF Mid $(Tempo, Len $(Tempo) - a, a + 1) = "01111111" ALLORA i8 = i8 + 1 FOR n = 1 TO
Len $(Tempo)
IF Mid $(Tempo, n, a + 2) = "011111110" ALLORA i8 = i8 + 1
IL PROSSIMO
'
b7:
a = 7
IF Mid $(Tempo, 1, a + 1) = "11111110" THEN i7 = i7 + 1 IF Len
(Tempo) <a THEN GOTO b6
IF Len $(Tempo) = a AND s = "11111111" ALLORA i7 = i7 + 1 IF Len
(Tempo) = a POI ANDARE b6
IF Mid $(Tempo, Len $(Tempo) - a, a + 1) = "01111111" ALLORA i7 = i7 + 1 FOR n = 1 TO
Len $(Tempo)
IF Mid $(Tempo, n, a + 2) = "011111110" ALLORA i7 = i7 + 1
IL PROSSIMO
'
b6:
a = 6
IF Mid $(Tempo, 1, a + 1) = "1111110" THEN i6 = i6 + 1 IF Len
(Tempo) <a THEN GOTO b5
IF Len $(Tempo) = a AND s = "111111" ALLORA i6 = i6 + 1 IF Len
(Tempo) = a POI ANDARE b5
IF Mid $(Tempo, Len $(Tempo) - a, a + 1) = "011111" THEN i6 = i6 + 1 FOR n = 1 TO
Len $(Tempo)
IF Mid $(Tempo, n, a + 2) = "01111110" ALLORA i6 = i6 + 1
IL PROSSIMO
'
b5:
a = 5
'
IF Mid $(Tempo, 1, a + 1) = "111110" THEN i5 = i5 + 1 IF Len
(Tempo) <a THEN GOTO b4
IF Len $(Tempo) = a AND s = "11111" ALLORA i5 = i5 + 1 IF Len
(Tempo) = a POI ANDARE b4
IF Mid $(Tempo, Len $(Tempo) - a, a + 1) = "011111" THEN i5 = i5 + 1 FOR n = 1 TO
Len $(Tempo)
IF Mid $(Tempo, n, a + 2) = "0111110" ALLORA i5 = i5 + 1
IL PROSSIMO
b4:
a = 4
IF Mid $(Tempo, 1, a + 1) = "11110" THEN i4 = i4 + 1 IF Len
(Tempo) <a THEN GOTO b3
IF Len $(Tempo) = a AND s = "1111" ALLORA i4 = i4 + 1 IF Len
(Tempo) = a POI ANDARE b3
IF Mid $(Tempo, Len $(Tempo) - a, a + 1) = "01111" THEN i4 = i4 + 1 FOR n = 1 TO
Len $(Tempo)
IF Mid $(Tempo, n, a + 2) = "011110" ALLORA i4 = i4 + 1
IL PROSSIMO
'
b3:
a = 3
IF Mid $(Tempo, 1, a + 1) = "1110" THEN i3 = i3 + 1 IF Len
(Tempo) <a THEN GOTO b2
IF Len $(Tempo) = a AND s = "111" THEN i3 = i3 + 1 IF Len
(Tempo) = a THEN GOTO b2
IF Mid $(Tempo, Len $(Tempo) - a, a + 1) = "0111" THEN i3 = i3 + 1 FOR n = 1 TO
Len $(Tempo)
IF Mid $(Tempo, n, a + 2) = "01110" ALLORA i3 = i3 + 1 NEXT
```

```

'
b2:
a = 2
IF Mid $(Tempo, 1, a + 1) = "110" THEN i2 = i2 + 1 IF Len
(Tempo) <a THEN GOTO beins
IF Len (Tempo) = a AND s = "11" THEN i2 = i2 + 1
IF Len (Tempo) = a THEN GOTO beins
IF Mid $(Tempo, Len (Tempo) - a, a + 1) = "011" THEN i2 = i2 + 1 FOR n = 1 TO
Len (Tempo)
IF Mid $(Tempo, n, a + 2) = "0110" ALLORA i2 = i2 + 1
IL PROSSIMO
gambe:
a = 1
IF Mid $(Tempo, 1, a + 1) = "10" THEN i1 = i1 + 1 IF Len
(Tempo) <a THEN GOTO b0
IF Mid $(Tempo, Len (Tempo) - a, a + 1) = "01" THEN i1 = i1 + 1 FOR n = 1 TO
Len (Tempo)
IF Mid $(Tempo, n, a + 2) = "010" ALLORA i1 = i1 + 1
IL PROSSIMO
'

b0:
textbox18.Text = i1
textbox19.Text = i2
textbox20.Text = i3
textbox21.Text = i4
textbox22.Text = i5
textbox23.Text = i6
textbox24.Text = i7
textbox25.Text = i8
textbox26.Text = i9
textbox27.Text = ia
FINE
Pulsante PUBLIC SUB 10_Click ()
'Chaitin, un noto ricercatore casuale, ha fornito la seguente sequenza in un saggio
'L'it chaitin GJ 1975 Scientific American 232 S 47-52
t = "01101100110111100010"
textarea1.Text = t
FINE
Pulsante PUBLIC SUB 11_Click ()
'Chaitin, un noto ricercatore casuale ha fornito un saggio a IN
la seguente sequenza
«01101100110111100010
'Questo episodio è stato cambiato,
'in modo che dopo ogni cifra segue il suo opposto,
t = "01101001101001011010011010100101011001"
textarea1.Text = t
' Rem Come puoi vedere dalla distribuzione ineguale delle coppie, ad esempio, la
possibilità diminuisce
' Rem Anche se a prima vista senza alcuna conoscenza dell'origine
' Rem forse c'è da aspettarsi ancora più coincidenza
FINE
Pulsante PUBLIC SUB 12_Click ()
'Numero casuale generato da più lanci di monete reali
una sequenza casuale di 0 e 1
'era determinato.
t = "101101101010011100101100111000001111001010000111101010001001
10111101100001000101000111011100101011011111000001001101000011011
10111101010110000010001110110001000000100111100000111101001000110111
100101010000010110100001100011010001100111101111000011011100100110000
0111110010000001100001000000110101010000011000101100001110011110010000
11011111110010010100111110010001001000100100100001000101001110011100111
10110000010
1001111111001011110111011000111011010110000011101100111101011001110 "
textarea1.Text = t
FINE

```

```
Tasto PUBLIC SUB 13_Click ()  
'BB, l'episodio originale di Chaitin diventa solo 2 * scritti uno dopo l'altro  
t = "01101100110111000100110110011011100010"  
textarea1.Text = t  
FINE  
Pulsante PUBLIC SUB 14_Click ()  
'Questo comando conta nel sistema binario da 1 a 2000' elenca i numeri  
uno dopo l'altro senza spazio '012345678 ... 0.1.10.11.100.101.110 ecc.  
Come stringa AS  
  
z AS intero  
PER z = da 0 a 2000  
    t = bin $(z)  
    te = te & t  
IL PROSSIMO  
textarea1.Text = te  
FINE
```

Il programma è un po 'lungo. Può certamente essere migliorato. Ha la limitazione che registra solo corse fino a una lunghezza di 10.

0.135.4 Esercizio preliminare 3: Programmazione del problema con il run test

Quote: Se non l'hai programmato, non hai capito.

Sfortunatamente, il test di esecuzione non funziona con nessuna sequenza 01, ma solo quando il numero di 0 e 1 è più o meno lo stesso. Ad esempio, non è possibile controllare la casualità della sequenza 0000000000000000010 nel test di esecuzione.

Contrariamente all'esercizio preliminare 2, le entropie vengono ora calcolate utilizzando il Runstest per i numeri binari in cui il numero di zeri è uguale al numero di uno.

- La lunghezza della sequenza 01 può essere variata, qui s = 8. (la lunghezza è la variabile s nell'elenco)
- Il numero di zeri e uno è lo stesso per semplicità.
- Il programma si avvia da solo.
- I risultati vengono visualizzati nella finestra diretta.
- Valutazione:
 - I valori di p_z intorno a 0 sono un segno di elevata entropia. I valori di P_z superiori a 1,5 o inferiori a -1,5 sono un segno di bassa entropia

```
PUBLIC SUB Form_Open ()  
s AS intero  
'Lunghezza del numero binario  
z AS intero  
'Contatore da 0 a 2^s-1  
zz AS intero  
'Contrasta da 1 a tutte le varianti con ozaehler = izaehler  
t AS stringa  
'numero binario  
ocounter AS intero  
'Numero di zeri  
izaehler AS Integer  
'Numero di uno  
tt AS stringa 'numero  
binario come stringa n AS  
intero
```

```

'Lunghezza del numero binario
char AS stringa
UM AS galleggiante
varianza AS Float
svar AS float
'Radice quadrata della varianza
due AS interi
somma AS intero
esegui AS intero
nn AS Integer chari AS
String charnext AS String
run counter AS Integer pz
AS Float

'La variabile di controllo corrisponde al valore di entropia della
sequenza' I valori pz intorno a 0 sono un segno di alta entropia.
'I valori pz> 1.5 o <-1.5 sono un segno di bassa entropia s = 10

'Lunghezza della
sequenza 01 zz = 0
FOR z = 0 TO (2^s - 1) t = Bin
$ (z, s)
tt = Str (t)
"PRINT" tt = "& tt
count = 0
izaehler = 0
PER n = 1 TO Len (tt)
char = Mid $ (tt, n, 1)
    SELEZIONA CASE TRUE
    CASE char = "0"
        count = count + 1
    CASE char = "1"
        izaehler = izaehler + 1
    FINE SELEZIONA
IL PROSSIMO
'PRINT izaehler
'PRINT count
SE izaehler = ozaehler ALLORA
zz = zz + 1 t
= tt
STAMPA "zz =" & zz & "t =" & t;
'corri giù
UM = 2 * s / 2 * s / 2 / (s / 2 + s / 2) + 1 'PRINT "UM
=" & UM
due = 2 * s / 2 * s / 2 somma
= s
varianza = due *(due - somma) / (somma * totale * (somma - 1))
SE varianza <0 ALLORA varianza = -varianza
'PRINT "variance =" & variance
errore di rughe = 1
PER nn = 1 TO Len (t) - 1
    chari = Mid $ (t, nn, 1)
    charnext = Mid $ (t, nn + 1, 1)
    IF chari <> charnext THEN runzehler = runzehler + 1
IL PROSSIMO
'PRINT "runzaehler =" & runzaehler;
svar = Sqr (varianza)
pz = (runzehler - UM) / svar
PRINT "pz =" & Str (pz)
'PRINT Str (pz)
FINISCI SE
IL PROSSIMO
FINE

```

Output dei risultati:

```
zz = 1 t = 00001111 pz = -2.291287847478 zz = 2 t
= 00010111 pz = -0.763762615826 zz = 3 t =
00011011 pz = -0.763762615826 zz = 4 t =
00011101 pz = -0.763762615826 zz = 5 t =
00011275 pz = -0.763762615826 zz = 6 t = 00100111 pz =
0.763762615826 zz = 7 t = 00101011 pz =
0.763762615826 zz = 8 t = 00101101 pz =
0.763762615826 zz = 9 t = 00101110 pz = 0

zz = 10 t = 00110011 pz = -0.763762615826 zz = 11 t
= 00110101 pz = 0.763762615826 zz = 12 t =
00110110 pz = 0
zz = 13 t = 00111001 pz = -0.763762615826 zz = 14 t
= 00111010 pz = 0
zz = 15 t = 00111100 pz = -1.527525231652 zz = 16 t
= 01000111 pz = -0.763762615826 zz = 17 t =
01001011 pz = 0.763762615826 zz = 18 t = 01001101
pz = 0.763762615826 zz = 19 t = 01001110 pz = 0

zz = 20 t = 01010011 pz = 0.763762615826 zz = 21 t
= 01010101 pz = 2.291287847478 zz = 22 t =
01010110 pz = 1.527525231652 zz = 23 t =
01011001 pz = 0.763762615826 zz = 24 t =
01011010 pz = 1.5225 01011100 pz = 0
zz = 26 t = 01100011 pz = -0.763762615826 zz = 27 t
= 01100101 pz = 0.763762615826 zz = 28 t =
01100110 pz = 0
zz = 29 t = 01101001 pz = 0.763762615826 zz = 30 t
= 01101010 pz = 1.527525231652 zz = 31 t =
01101100 pz = 0
zz = 32 t = 01110001 pz = -0.763762615826 zz = 33 t
= 01110010 pz = 0
zz = 34 t = 01110100 pz = 0
zz = 35 t = 01111000 pz = -1.527525231652 zz = 36 t
= 10000111 pz = -1.527525231652 zz = 37 t =
10001011 pz = 0
zz = 38 t = 10001101 pz = 0
zz = 39 t = 10001110 pz = -0.763762615826 zz = 40 t
= 10010011 pz = 0
zz = 41 t = 10010101 pz = 1.527525231652 zz = 42 t
= 10010110 pz = 0.763762615826 zz = 43 t =
10011001 pz = 0
zz = 44 t = 10011010 pz = 0.763762615826 zz = 45 t =
10011100 pz = -0.763762615826 zz = 46 t = 10100011
pz = 0
zz = 47 t = 10100101 pz = 1.527525231652 zz = 48 t
= 10100110 pz = 0.763762615826 zz = 49 t =
10101001 pz = 1.527525231652 zz = 50 t =
10101010 pz = 2.291287847478 zz = 51 t =
10101100 pz = 0.763762615 10110001 pz = 0
zz = 53 t = 10110010 pz = 0.763762615826 zz = 54 t =
10110100 pz = 0.763762615826 zz = 55 t = 10111000
pz = -0.763762615826 zz = 56 t = 11000011 pz =
1.527525231652 zz = 57 t = 11000101 pz = 0

zz = 58 t = 11000110 pz = -0.763762615826 zz = 59 t
= 11001001 pz = 0
zz = 60 t = 11001010 pz = 0.763762615826 zz = 61 t =
11001100 pz = -0.763762615826 zz = 62 t = 11010001
pz = 0
zz = 63 t = 11010010 pz = 0.763762615826
```

```

zz = 64 t = 11010100 pz = 0.763762615826 zz = 65 t
= 11011000 pz = -0.763762615826 zz = 66 t =
11100001 pz = -1.527525231652 zz = 67 t = 11100010
pz = -0.763762615826 zz = 68 t = 11101000 pz = -
0.76376 =1526 69 t = 11101000 pz =
0.763762615826 zz = 70 t = 11110000 pz =
2.291287847478

```

Illustrazione Rappresentazione grafica dei valori di entropia di una sequenza 8 01. Più grandi sono le colonne, minore è l'entropia. Più la colonna è vicina a zero, maggiore è l'entropia.

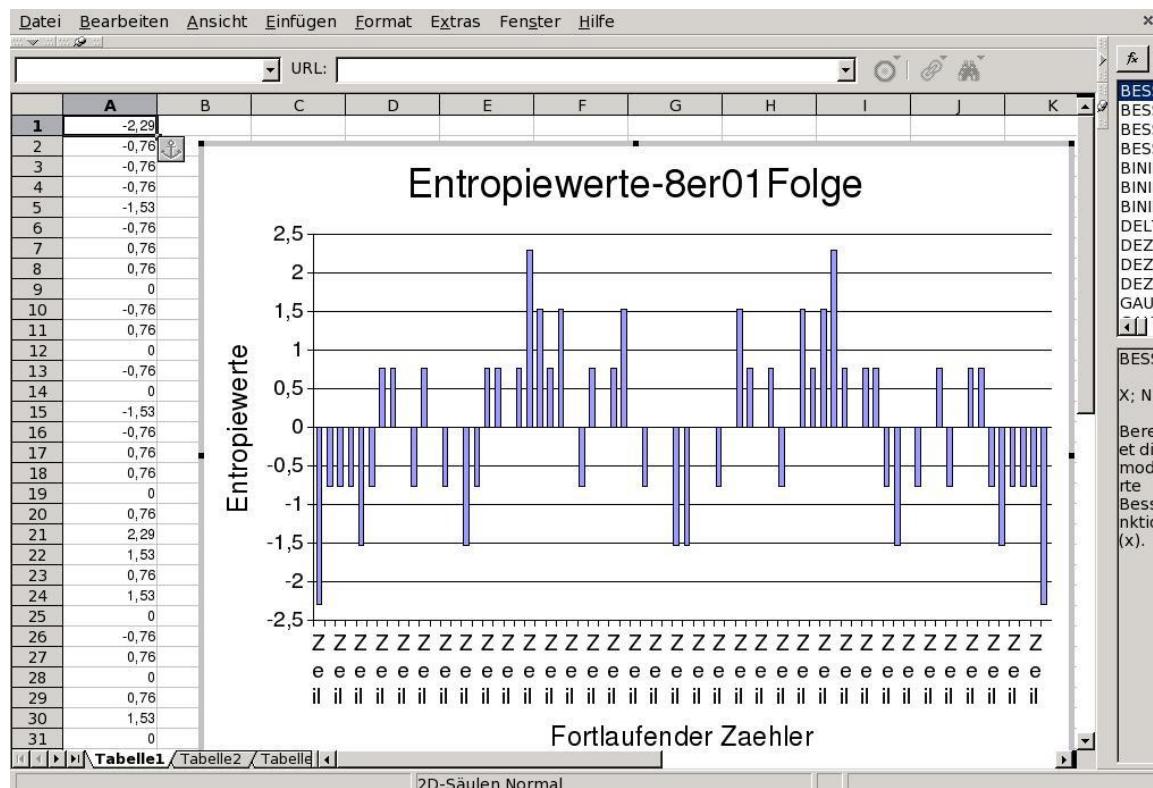


Fig.50

0.136 Calcoli casuali con la funzione casuale

Per il calcolo di numeri casuali (o numeri pseudocasuali) sono necessari i comandi Randomize (), RND e INT

Guarda anche <http://www.madeeasy.de/7/prgmini.htm#z>

Attenzione: Randomize () deve essere sostituito da RANDOMIZE nelle versioni più recenti.

0.136.1 Lancio della moneta

0.136.2 Moneta molto spesso, 01 sequenza casuale

Se vuoi una sequenza casuale più lunga di 0 e 1, puoi farlo con il seguente programma. Come con una moneta che lancia più e più volte, viene generata una sequenza di numeri casuali tra 0 e 1. Hai bisogno di un'area di testo e di un pulsante di comando per avviare il programma. Il testo di origine dietro il campo del comando è:

```
Pulsante PUBLIC SUB1_Click ()  
    " lanci multipli di monete per generare una sequenza casuale di 0 e 1  
    DIM un numero intero AS  
    DIM t AS stringa  
    DIM M AS Interio  
    Rendi casuale ()  
    PER a = 1 FINO A 20000  
        M = Int (2 * Rnd)  
        ' Genera un lancio compreso tra 0 e 1 t = t &  
        LTrim $ (Str $ (M))  
        IL PROSSIMO  
        Textarea1.Text = t  
    FINE
```

0.136.3 Arrotolare fino a 6

Questo programma genera numeri casuali compresi tra 1 e 6. come un cubo Il risultato viene visualizzato in una casella di testo. Hai bisogno di un pulsante di comando, un campo di etichetta di testo e una casella di testo per avviare il programma.

Vedi anche: immagine⁸⁰

Il testo sorgente dietro il campo di comando Dice è:

```
Pulsante PUBLIC SUB1_Click ()  
    DIM Cube AS Integer  
    Rendi casuale ()  
    Cubo = Int (Rnd (1,7))  
    ' Effettua un tiro tra 1 e 6  
    TextLabel1.Text = "Hai rotolato un" TextBox1.Text =  
    Cube  
    FINE
```

Sono stati utilizzati i seguenti comandi:

Randomize: inizializza il generatore di numeri casuali.

Note: l'argomento numero può essere qualsiasi espressione numerica legale. Il generatore di numeri casuali viene inizializzato per numero assegnandogli un nuovo valore iniziale. Se si omette il numero, il valore emesso dalla funzione timer viene utilizzato come nuovo valore iniziale. Quindi il comando Randomize Timer è effettivamente ridondante.

80 <http://www.madeeasy.de/7/prgwuerfel6.png>

Se Randomize non viene utilizzato, la funzione Rnd restituisce la stessa sequenza di numeri casuali ogni volta che il programma viene eseguito. Se vuoi che la sequenza di numeri casuali cambi ogni volta che il programma viene eseguito, imposta un'istruzione randomize senza argomenti all'inizio del programma.

Rnd Restituisce un numero casuale.

```
Rnd ([Min [, Max]])
```

Calcola un numero pseudocasuale utilizzando l'algoritmo di Lehmer.

Se non vengono specificati parametri, viene generato un numero casuale nell'intervallo [0, 1] compreso tra zero e uno.

Se viene specificato il primo parametro (= Min), viene generato un numero casuale nell'intervallo [0, Min] tra zero e il parametro.

Se si specificano i due parametri, viene generato un numero casuale nell'intervallo [Min, Max].

Esempio:

```
PRINT Rnd
```

=> 0,019539254718

```
STAMPA Rnd (2)
```

=> 0,040205506608

```
PRINT Rnd (Pi, Pi (2))
```

=> 3.204108046818

Int

Valore = Int (numero)

Taglia le cifre dopo il punto decimale e restituisce solo i numeri prima del punto decimale. In altre parole: restituisce il valore numerico prima del punto decimale, ovvero viene emesso il numero intero inferiore al numero originale.

Esempio:

```
PRINT Int (Pi)
```

=> 3

```
PRINT Int (-Pi)
```

=> -4

Contenuti

0.136.4 Controllare i dadi a 6 facce

Di seguito è elencato un programma che verifica la funzione casuale sulla base di un cubo. Funziona direttamente nella modalità terminale di Gambas. Quindi non hai bisogno di una forma.

```
' File di classe Gambas
STATIC PUBLIC SUB Main ()
DIM x AS Integer
DIM w AS Interger DIM
a AS Interger DIM b
AS Interger DIM c AS
Interger DIM d AS
Interger DIM e AS
Interger DIM f AS
Interger Randomize ()
PER x = 1 FINO A 1000000
'w = Int (Rnd (6) + 1) identico alla riga successiva w = Int (Rnd
(1,7))

SELEZIONA
CASE w
CASE 1
    a = a + 1
CASE 2
    b = b + 1
CASE 3
    c = c + 1
CASE 4
    d = d + 1
CASE 5
    e = e + 1
CASE 6
    f = f + 1
FINE
SELEZIONA
AVANTI
STAMPA a, b, c, d, e, f
FINE
```

0.136.5 dadi fino a 12

Hai bisogno di una casella di testo, un campo di etichetta di testo e un pulsante di comando per avviare il programma. Il problema è facile da risolvere allo stesso modo del dado a 6 facce:

```
Pulsante PUBLIC SUB1_Click ()
Cubo AS Integer
Rendi casuale ()
Dice = Int (Rnd (1,13)) 'Crea un tiro tra 1 e 12
TextLabel1.Text = "Hai tirato un"
TextBox1.Text = Cube
FINE
```

È più breve in modalità terminale:

```
STATICO PUBBLICO SUB Principale ()
Cubo AS Integer
Rendi casuale ()
Dice = Int (Rnd (1,13)) 'Crea un tiro tra 1 e 12
```



```
Cubo PRINT  
FINE
```

O anche più breve:

```
STATICO PUBBLICO SUB Principale ()  
Rendi casuale ()  
PRINT Int (Rnd (1,13))  
FINE
```

0.136.6 Lotto

Questo programma un po 'più complicato estraе casualmente sei numeri della lotteria e un numero aggiuntivo dai numeri da 1 a 49. L'intera cosa viene distribuita in modo ordinato e si evitano i duplicati.

Prima di accettare un programma casuale, dovresti eseguirlo 1000 volte e controllare l'equa distribuzione e l'imprevedibilità della sequenza numerica con il computer.

Sono necessari 2 pulsanti di comando e un'area di testo per avviare il programma.



Fig.51 Screenshot del programma dei numeri della lotteria

Il testo sorgente dietro il campo di comando Numeri del lotto recita:

```
' File di classe Gambas ar
AS Integer [] PUBLIC SUB
Form_Open ()
'Quando il modulo viene aperto, la matrice viene riempita con tutti zeri.

x AS intero
ar = NEW Integer []
ME.Text = "Numeri Lotto"
PER x = da 0 a 48
    ar.Add (Str (0))
IL PROSSIMO
FINE

Pulsante PUBLIC SUB1_Click ()
x AS intero
r AS intero
elemento AS Integer
txt come stringa
z AS intero
txt = ""
'L'array è completamente impostato a zero
PER x = da 0 a 48
    ar [x] = 0
IL PROSSIMO
'quindi la matrice viene riempita con 6 numeri casuali nei punti che corrispondono al
numero casuale
Rendi casuale ()
PER x = da 0 a 5
    r = Int (Rnd (1.49))
    SE r = ar [r] ALLORA x = x - 1
    'Se si verifica un doppelganger torna indietro di uno ar [r] = r

IL PROSSIMO
'vengono emessi solo numeri diversi da 0 (<> 0) PER OGNI elemento
IN ar
    IF elemento <> 0 THEN txt = txt & element & Chr (10)
IL PROSSIMO
'Cerca un numero aggiuntivo che non sia già stato estratto
RIPETERE
    z = Int (Rnd (1.49))
FINO A z <> ar [z]
'L'output di testo viene compilato
txt = txt & Chr (10) & "Numero aggiuntivo:" & z
textarea1.Text = txt
FINE

PUBLIC SUB Button2_Click ()
    ME.Chiudi
FINE
```

0.136.7 lettere casuali

Questo programma genera lettere casuali tra A e Z. Il risultato viene visualizzato in un'area di testo.

Hai bisogno di un'area di testo e 2 pulsanti di comando per avviare il programma. Quindi assomiglia a questo:

Il testo di origine dietro il campo di comando delle lettere casuali è:

```
Pulsante PUBLIC SUB1_Click ()
DIM Z AS Intero
Lettera DIM come stringa
Rendi casuale ()
Textarea1.Text = ""
PER Z = DA 1 A 50
    Lettera = Chr $ (Int (Rnd (65.91)))
    TextArea1.Text = TextArea1.Text e lettera
IL PROSSIMO
FINE
```

Il comando CHR \$ converte i numeri in lettere.

Character = Chr \$ (numero) restituisce il carattere che corrisponde al codice ASCII del numero.

Esempio 1:

```
STAMPA Chr $ (65)
```

Risultato: A.

Esempio 2:

```
Pulsante PUBLIC SUB1_Click ()
DIM x AS intero
PER x = da 30 a 133
    STAMPA Chr $ (x);
IL PROSSIMO
FINE
```

0.136.8 Punti casuali

Con il seguente programma puoi riempire un'area di disegno con punti casuali.

Per il programma hai bisogno di:

- un'area di disegno
- due pulsanti di comando.

L'intera cosa assomiglia a questo:

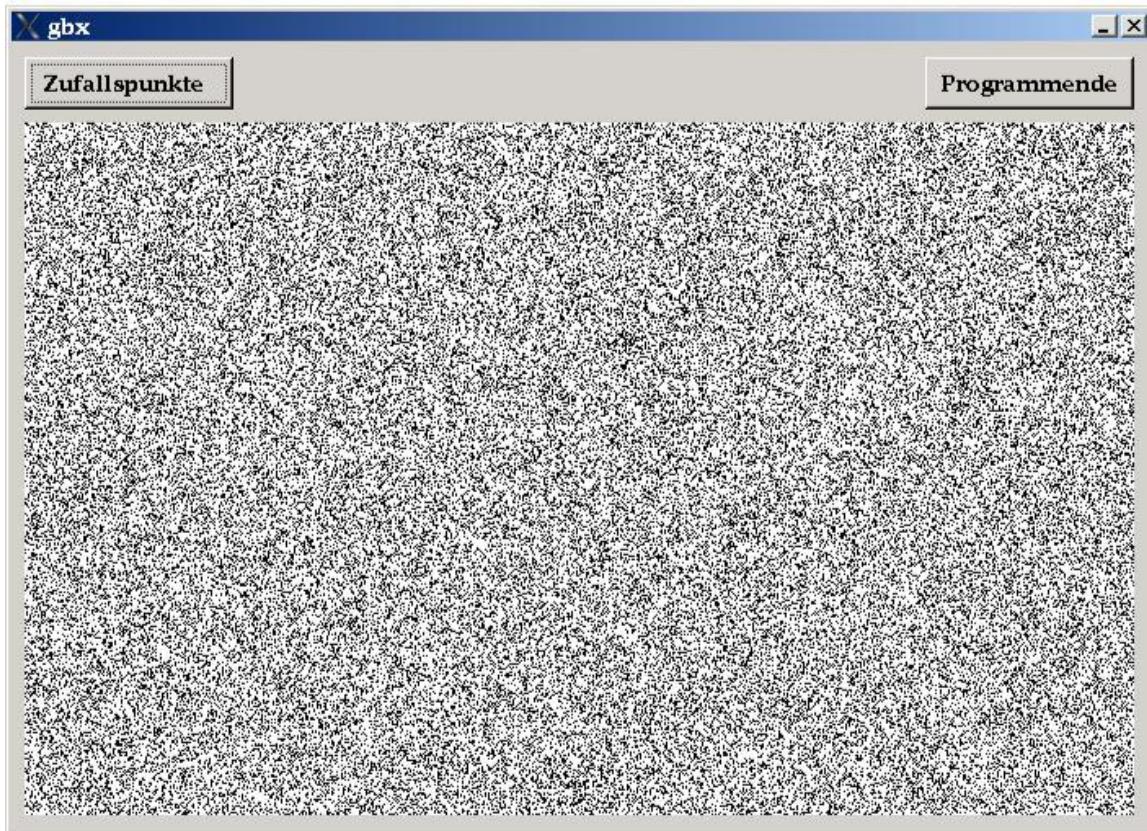


Fig.52

Il codice per questo:

```
Pulsante PUBLIC SUB1_Click ()  
x AS intero  
y AS intero  
z AS intero  
Draw.Begin (DrawingArea1)  
PER z = 1 FINO A 100000  
    x = Int (Rnd (0, DrawingArea1.Width))  
    y = Int (Rnd (0, DrawingArea1.Height))  
    Draw.Point (x, y)  
IL PROSSIMO  
Disegna Fine  
FINE  
PUBLIC SUB Button2_Click ()  
    ME.Chiudi  
FINE
```

0.136.9 quadrati casuali

Programma coriandoli

Con il seguente programma Coriandoli, un campo di disegno viene riempito con tanti quadratini di colore diverso. Casuale appare 3 volte nel programma: 1. Il colore è casuale

eletto. 2. La coordinata x dei quadrati è scelta a caso. 3. La coordinata y dei quadrati è scelta a caso.

Hai bisogno di una forma, su di essa metti un'area di disegno. Quindi hai bisogno di 2 pulsanti di comando.

L'intera cosa assomiglia a questo:

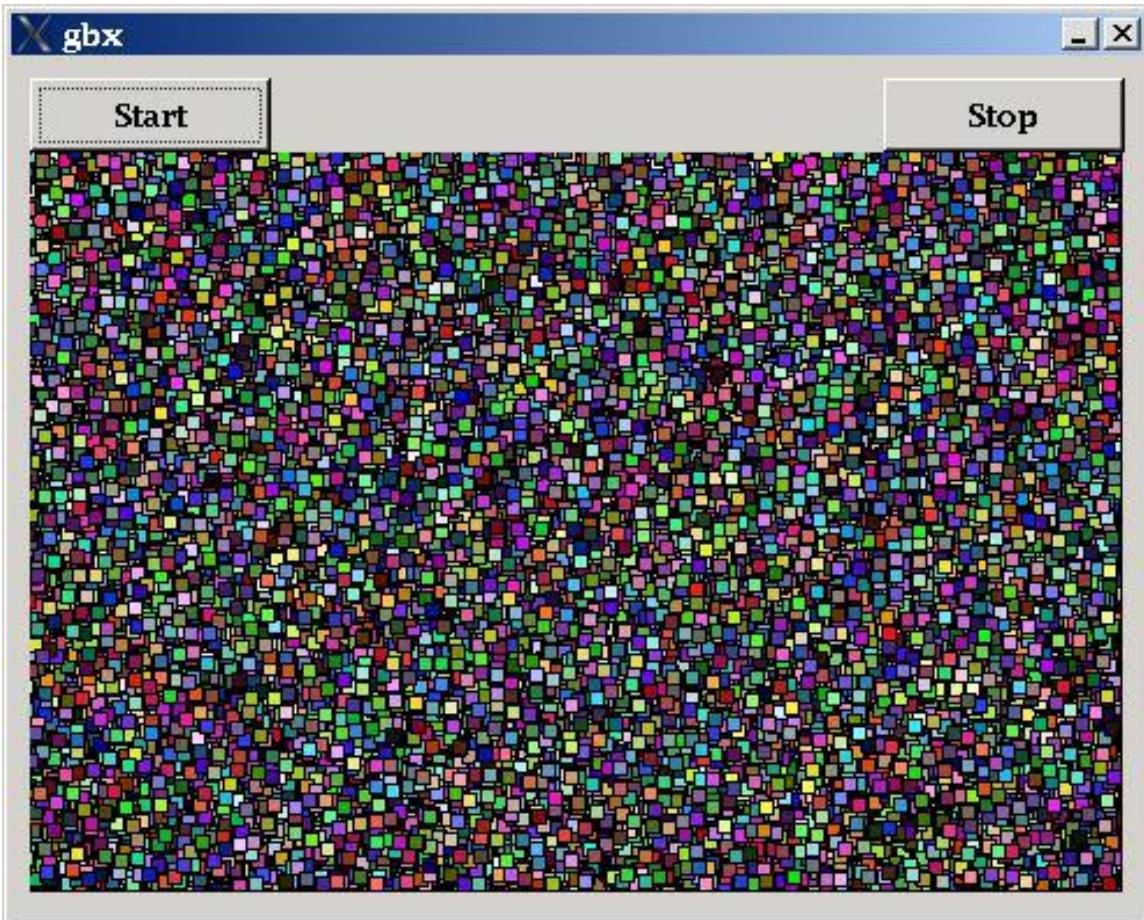


Fig.53

Il codice per questo:

```

PUBLIC SUB Form_Open ()
    Button1.Text = "Avvia"
    Button2.Text = "Stop"
FINE
Pulsante PUBLIC SUB1_Click ()
    i AS intero
    X AS intero
    Y AS intero
    Draw.Begin (DrawingArea1)
    PER i = 1 FINO A 200000
        Draw.FillColor = Int (Rnd (& HFFFFFF &))
        'PRINT Int (Rnd (& HFFFFFF &))
        Draw.FillStyle = Fill.Solid
        X = Int (Rnd (DrawingArea1.Width))
    
```

```
Y = Int (Rnd (DrawingArea1.Height))
Draw.Rect (X - 7, Y - 7, 7, 7)
IL PROSSIMO
Disegna Fine
FINE
PUBLIC SUB Button2_Click ()
    ME.Chiudi
FINE
```

0.136.10 Statistiche sul lancio di monete

1 DrawingArea, Immagine

Per creare grafici in Gambas dovrà familiarizzare con l'elemento di controllo dell'area di disegno (DrawingArea).



Fig.54

1.1 Le coordinate della DrawingArea1

Il seguente programmino fornisce le coordinate della DrawingArea1

Fai un nuovo progetto. Apri una nuova forma: fai clic su Forma con il pulsante destro del mouse.

Assegnare un nome al modulo o utilizzare il suggerimento Form1. Prendine uno

DrawingArea = area di disegno dalla casella degli strumenti. Non rinominarli.

Copia il codice dal basso nel programma. Avvia il programma con F5. L'output dei numeri corrisponde ai numeri nelle proprietà (= F4 = Proprietà) di DrawingArea1.

Notare quanto segue: l'angolo superiore sinistro è (X, Y), l'angolo inferiore destro è (X + DrawingArea1.Width, Y + DrawingArea1.Height)

Il codice:

```
PUBLIC SUB Form_Open ()  
STAMPA DrawingArea1.X  
STAMPA DrawingArea1.Y  
PRINT DrawingArea1.Width  
STAMPA DrawingArea1.Height  
FINE
```

Risultato

Se all'inizio non è stata modificata l'area di disegno, vengono emessi i seguenti valori

```
0  
0  
64  
32
```

2 immagini

Vedi Gambas: _Pictures¹

```
PUBLIC SUB Form_Open ()  
    STAMPA DrawingArea1.X  
    STAMPA DrawingArea1.Y  
    PRINT DrawingArea1.Width  
    STAMPA DrawingArea1.Height  
    FINE
```

¹ Capitolo 6.2 a pagina 216

3 colori

Vedi Gambas: _Colors¹

¹ Capitolo 6.18 a pagina 222

4 funzioni matematiche

Vedi Gambas: aritmetica¹

¹ Capitolo 0.113.3 a pagina 107

5 punti, linea, cerchio

In Gambas puoi disegnare solo in un campo grafico (= DrawingArea). In un campo grafico puoi disegnare punti, rettangoli, ellissi, linee e altre cose. È possibile utilizzare diversi colori e larghezze di penna.

5.1 punti

Il seguente programma di esempio traccia alcuni punti. È difficile vedere un singolo punto.

Per fare questo, crea un nuovo progetto. Apri una nuova forma: fai clic su Forma con il pulsante destro del mouse. Dalle il nome Fmain. Ottieni un pulsante di comando = pulsante dalla casella degli strumenti. Ottieni un DrawingArea = area di disegno dalla casella degli strumenti.

Copia il codice nel programma. Avvia il programma con F5. Presta attenzione alla colorazione delle proprietà, altrimenti non vedrai nulla.

```
Pulsante PUBLIC SUB1_Click ()  
Draw.Begin (DrawingArea1)  
Draw.Point (100,100)  
Draw.Point (100,101)  
Draw.Point (100,102)  
Draw.Point (100,103)  
Draw.Point (100,99)  
Draw.Point (100.98)  
Draw.Point (100.97)  
Draw.Point (100.96)  
Disegna Fine  
FINE
```

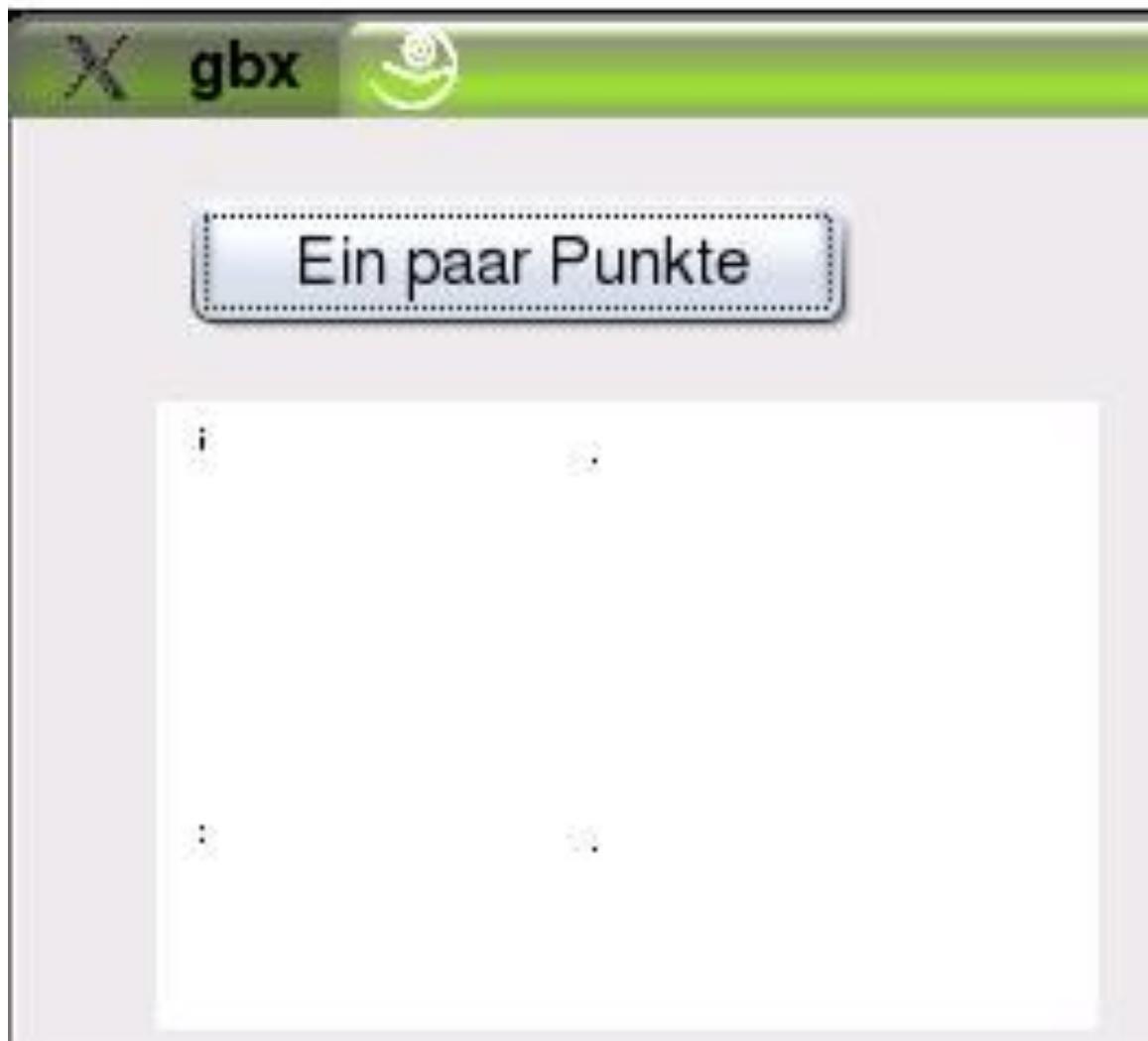


Fig.55

5.2 linee

Il prossimo programmino traccia alcune linee.

Fai un nuovo progetto. Apri una nuova forma: fai clic su Forma con il pulsante destro del mouse. Dalle il nome Fmain. Ottieni un pulsante di comando = pulsante dalla casella degli strumenti. Ottieni un DrawingArea = area di disegno dalla casella degli strumenti.

Copia il codice nel programma. Avvia il programma con F5. Presta attenzione alla colorazione delle proprietà, altrimenti non vedrai nulla.

```
Pulsante PUBLIC SUB1_Click ()  
Dichiara la variabile DIM B AS Integer.  
Draw.Begin (DrawingArea1)  
FOR B = DA 1 A 200 PASSO 10 'Inizializza loop.
```

```
Draw.Line (1, B, 500, B)
IL PROSSIMO
Disegna Fine
FINE
```

5.2.1 Larghezza della linea

Questo programma grafico mostra la programmazione di linee di diversi spessori. Sono necessari di nuovo un pulsante di comando e un DrawingArea. Il comando decisivo qui si chiama Draw.LineWidth Il codice seguente è dietro la riga di comando:

```
Pulsante PUBLIC SUB1_Click ()
Dichiara la variabile DIM B AS Integer.
Draw.Begin (DrawingArea1)
Draw.Line (10,100, 20, 100)
FOR B = 1 TO 100 STEP 10 'inizializza il loop.
    Draw.LineWidth = B 'Imposta la larghezza della penna.
    Draw.Line (10 + B, 100, 20 + B, 100)
IL PROSSIMO
Disegna Fine
FINE
```

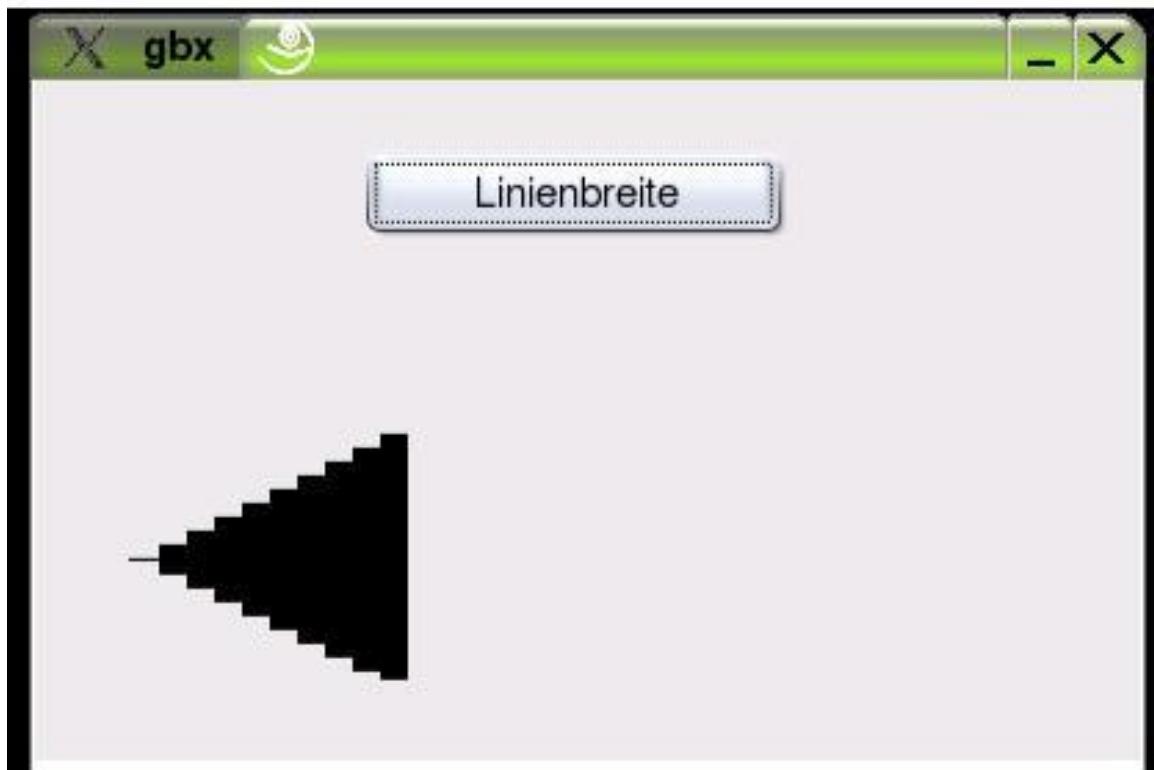


Fig.56

5.2.2 Qualche riga in più

```
Pulsante PUBLIC SUB1_Click ()  
Draw.Begin (DrawingArea1)  
' Disegna una linea orizzontalmente attraverso il centro della forma  
Draw.Line (0, ME.Height / 2, ME.Width, ME.Height / 2)  
' Disegna una linea perpendicolare al centro della forma Draw.Line  
(ME.Width / 2, 0, ME.Width / 2, ME.Height)  
' Disegna una linea dall'angolo superiore sinistro all'angolo inferiore destro Draw.Line (0,  
0, ME.Width, ME.Height)  
' Disegna una linea dall'angolo superiore destro all'angolo inferiore sinistro Draw.Line  
(ME.Width, 0,0, ME.Height)  
Disegna Fine  
FINE
```

Perché le linee non sono centrate correttamente?

Aggiungi il seguente codice subito dopo la prima riga:

```
DrawingArea1.Resize (ME.Width, ME.Height)  
DrawingArea1.Background = & HFFFFFF &
```

Sembra meglio adesso?

Se hai la linea

```
DrawingArea1.Resize (ME.Width, ME.Height)
```

dopo il comando `Draw.Begin (DrawingArea1)`, compare un messaggio di errore:

```
QPaintDevice: impossibile distruggere il dispositivo di verniciatura in fase di verniciatura  
Errore X: BadDrawable (parametro Pixmap o Window non valido) 9  
Codice operativo principale: 66  
Codice operativo minore: 0  
ID risorsa: 0x2e00010  
X Errore: RenderBadPicture (parametro Picture non valido) 180  
Codice operativo principale: 153  
Codice operativo minore: 5  
ID risorsa: 0x2e00025
```

5.2.3 Colore della linea

Se vuoi cambiare il colore della tua linea, il seguente programma ti aiuterà

```
Draw.Begin (DrawingArea1)  
Draw.ForeColor = & HFFFFFF  
' Colore bianco  
' Draw.ForeColor = & H0000FF & linea blu  
' Draw.ForeColor = & HFF00FF & viola  
' Draw.ForeColor = & HFF0000 & rosso  
' Draw.ForeColor = & H00FF00 & verde  
' Draw.ForeColor = & HFFFF00 & giallo  
' Draw.ForeColor = & H00FFFF & turchese
```

```
' & H000000 & = Black Draw Line
(1, 130, 500, 400) Draw.End
```

5.3 rettangolo

Il seguente programma disegna un rettangolo.

Fai un nuovo progetto. Apri una nuova forma: fai clic su Forma con il pulsante destro del mouse. Dalle il nome Fmain. Ottieni un pulsante di comando = pulsante dalla casella degli strumenti. Ottieni un DrawingArea = area di disegno dalla casella degli strumenti.

Assegna un nome all'area di disegno. A tale scopo, contrassegnare l'area di disegno con il mouse. Premi F4 e otterrai le proprietà dell'area di disegno. Nella seconda riga sotto (Nome) inserisci il tuo nome da invece di DrawingArea.

Copia il codice nel programma. Avvia il programma con F5. Presta attenzione alla colorazione delle proprietà, altrimenti non vedrai nulla.

Si prega di prestare attenzione a quanto segue:

Le coordinate dell'area di disegno vanno dall'angolo superiore sinistro (DrawingArea.X = 0, DrawingArea.Y = 0) all'angolo inferiore destro (DrawingArea.Width = any, DrawingArea.Height = any).

Le prime 2 cifre delle coordinate del rettangolo indicano le coordinate assolute dell'angolo superiore sinistro del rettangolo nell'area di disegno. Nell'esempio sono selezionati 100 e 100. Gli altri 2 numeri indicano le coordinate dell'angolo inferiore destro del rettangolo rispetto all'angolo superiore sinistro. Nell'esempio 90 e 90. Le coordinate assolute dell'angolo inferiore destro sono quindi 190 e 190 nell'esempio.

Questo è diverso con il comando Draw.Line. Qui sono state scelte le coordinate assolute.

Come esercizio puoi provare a disegnare le 2 diagonali nel rettangolo con una linea di disegno.

Quali sono le coordinate per questo. Soluzione vedi sotto.

```
Pulsante PUBLIC SUB1_Click ()
Draw.Begin (da)
Draw.Rect (100, 100, 90, 90)
Disegna Fine
FINE
```

Variante con diagonali

```
Pulsante PUBLIC SUB1_Click ()
Draw.Begin (da)
Draw.Rect (100, 100, 90, 90)
Draw.Line (100, 100, 190, 190)
Draw.Line (100, 190, 190, 100)
```

Disegna Fine
FINE

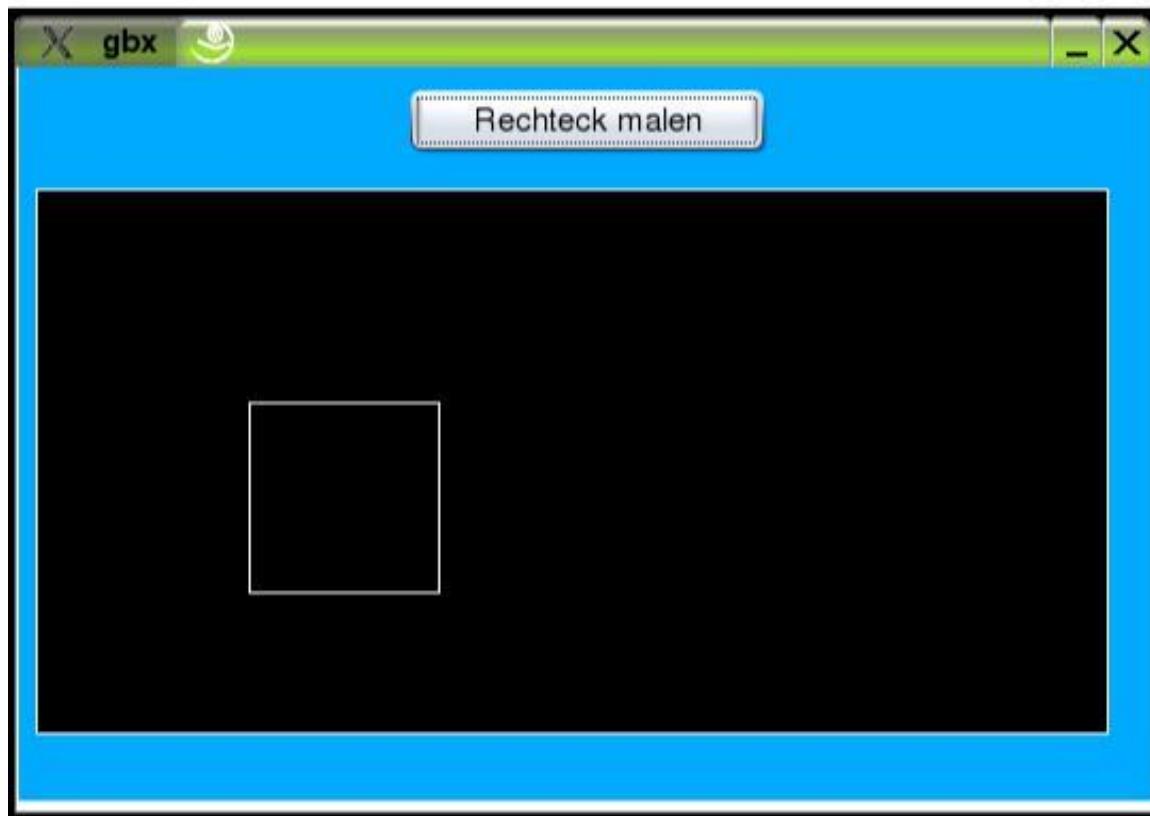


Fig.57

5.3.1 Riempire il rettangolo con il colore

Il programma di esempio disegna un rettangolo e lo riempie di vernice bianca.

```
Pulsante PUBLIC SUB1_Click ()  
Draw.Begin (DrawingArea1)           'Usa DrawingArea1 come "tela"  
Draw.FillColor = Color.white  
Draw.FillStyle = 1  
Draw.ForeColor = Color.white 'anche il bordo sarà bianco Draw.Rect  
(100, 100,200,200) Draw.End  
  
FINE
```

5.4 cerchio

Questo programma grafico mostra la programmazione di un grafico circolare:

Dietro il comando alcuni cerchi è il seguente codice:

```
Pulsante PUBLIC SUB1_Click ()
    Draw.Begin (Drawingarea1)
    Draw.Ellipse (10, 50, 90, 90)
    Draw.Ellipse (10, 50, 50, 50)
    Draw.Ellipse (10, 50, 20, 20)
    Disegna Fine
FINE
```

Hai bisogno di un'area di disegno (classe Drawingarea) e un pulsante della classe Commandbutton sul tuo form per far partire il programma. Gioca con le dimensioni individuali e guarda la grafica cambiare.

Semplifica il programma in modo che venga disegnato solo un cerchio.

Si prega di prestare attenzione a quanto segue:

- Il primo numero dopo il comando Draw.Ellipse indica la coordinata x dell'angolo superiore sinistro di un rettangolo in cui l'ellisse o il cerchio è tangenzialmente incorporato.
 - Questo valore può essere chiamato xloe (valore x nell'angolo in alto a sinistra)
- Il secondo numero indica la coordinata y dell'angolo superiore sinistro.
 - Questo valore può essere chiamato yloe (valore y nell'angolo in alto a sinistra)
- Il terzo numero indica la coordinata x dell'angolo inferiore destro. Il valore non è un valore assoluto ma relativo all'angolo superiore sinistro.
 - Questo valore può essere chiamato rxrue (valore x relativo, angolo in basso a destra)
- Il quarto numero indica la coordinata y dell'angolo inferiore destro.
 - Questo valore può essere chiamato ryhue (valore y relativo, angolo in basso a destra)

Se vuoi programmare un cerchio, il terzo e il quarto parametro devono essere gli stessi, perché questo è l'unico modo in cui il rettangolo che si tocca diventa un quadrato. Solo un quadrato può toccare un cerchio con tutti e 4 i lati.

```
Draw.Ellipse (xloe, yloe, rxrue, ryhue)
```

```
Per il cerchio, rxrue = ryhue
```

Verificare se queste specifiche sono rispettate nel programma circolare di cui sopra. Modificare i parametri 3 e 4 e vedere cosa succede.

Per rendere il tutto chiaro, è meglio programmare un cerchio e il quadrato circostante.

```
Pulsante PUBLIC SUB1_Click ()
    Draw.Begin (Drawingarea1)
    Draw.Ellipse (10, 50, 90, 90)
    Draw.Rect (10, 50, 90, 90)
    Disegna Fine
FINE
```

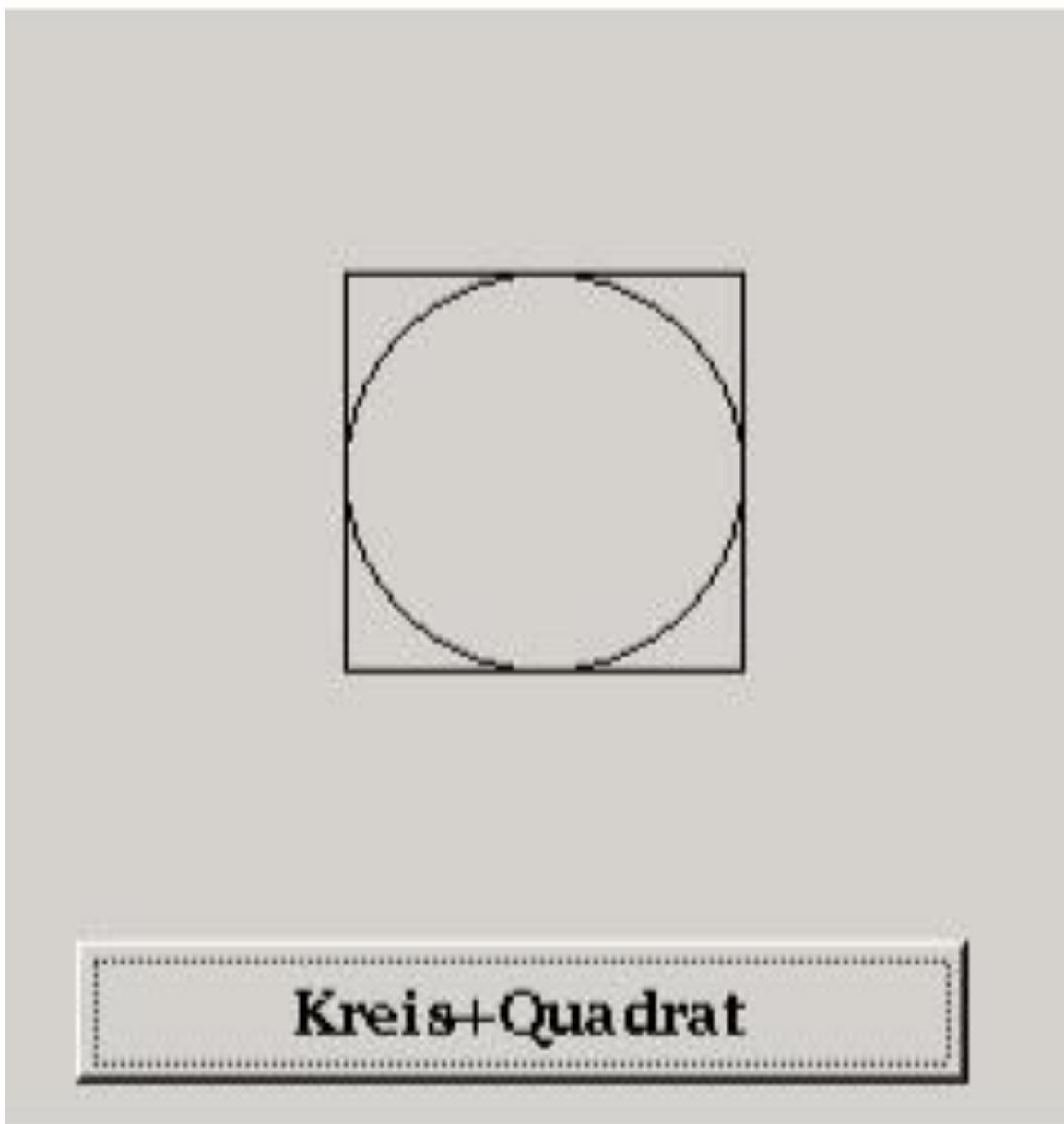


Fig.58

Il raggio del cerchio può essere calcolato da queste specifiche:

Raggio = (terzo o quarto parametro di Draw.Ellipse) / 2

Anche le coordinate del centro del cerchio possono essere calcolate da queste specifiche.

$mx = xloe + rxrue / 2$
 $mio = yloe + rxrue / 2$

Come puoi quindi calcolare i 2 punti focali di un'ellisse, se $rxrue$ e $ryrue$ sono diversi ??

5.4.1 Il tubo

Un altro esempio con molti cerchi diversi. Il programma è stato programmato con il controllo del menu.

Vedi Gambas: Menu¹ Un DrawingArea è necessario anche nel modulo.

Ecco come appare:

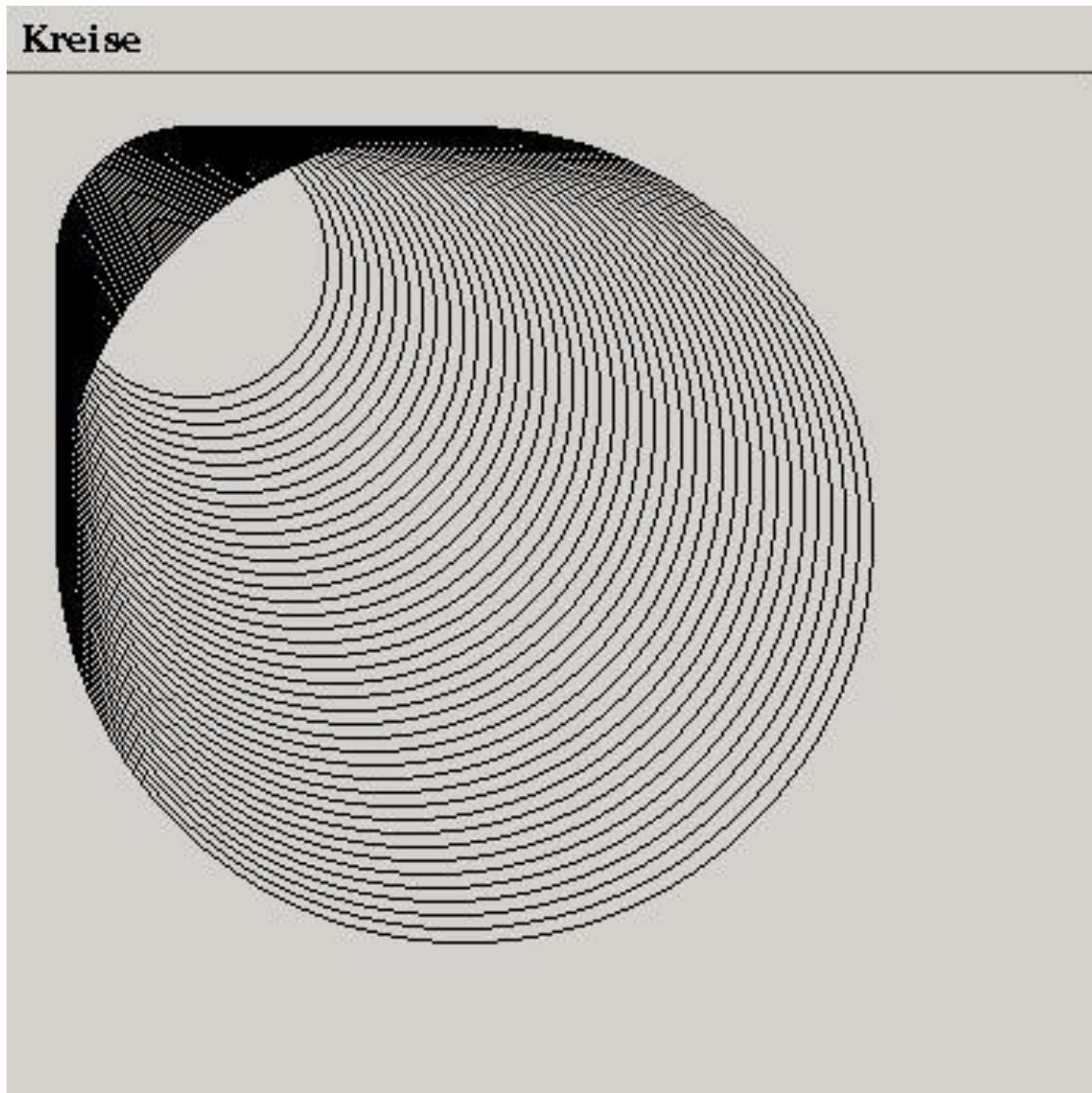


Fig.59

¹ Capitolo 6.72 a pagina 278

Il codice per questo:

```
PUBLIC SUB Form_Open ()  
    DrawingArea1.Resize (ME.Width, ME.Height)  
FINE  
PUBLIC SUB Menu2_Click ()  
x AS intero  
Draw.Begin (Drawingarea1)  
PER x = da 0 a 200 FASE 5  
    Draw.Ellipse (10, 10, x + 100, x + 100,5)  
IL PROSSIMO  
Disegna Fine  
FINE  
PUBLIC SUB Menu3_Click ()  
    ME.Chiudi  
FINE
```

5.4.2 Cerchi concentrici

L'esempio seguente disegna diversi cerchi concentrici in un DrawingArea. Hai bisogno

- un'area di disegno (campo grafico)
- un pulsante di comando

per avviare il programma.

```
PUBLIC SUB Form_Open ()  
    Drawingarea1.cached = TRUE  
    IF Drawingarea1.width > Drawingarea1.height THEN  
        Drawingarea1.width = Drawingarea1.height ELSE  
        Drawingarea1.height = Drawingarea1.width  
    FINISCI SE  
    Button1.Text = "Cerchie"  
    ME.Text = "Cerchi concentrici"  
    FINE  
    Pulsante PUBLIC SUB1_Click ()  
    mx AS intero  
    'coordinata x del punto centrale  
    il mio AS intero  
    Coordinata y del punto centrale  
    'Raggio = mx - a  
    un numero intero AS  
    'coordinata x dell'angolo superiore sinistro  
    b AS intero  
    'coordinata y nell'angolo superiore sinistro  
    c AS intero  
    'a + c fornisce la coordinata x dell'angolo inferiore destro d AS Integer  
    'b + d fornisce la coordinata y dell'angolo inferiore destro  
    mx = Drawingarea1.width / 2  
    Draw.Begin (Drawingarea1)  
    a = 0  
    PER a = 0 TO mx FASE 10  
        b = a  
        c = 2 * (mx - a)  
        d = c  
        draw.Ellipse (a, b, c, d)
```

IL PROSSIMO
disegnare
FINE

La figura seguente mostra uno screenshot del programma. Tuttavia, una piccola cosa è sbagliata. Hai notato.

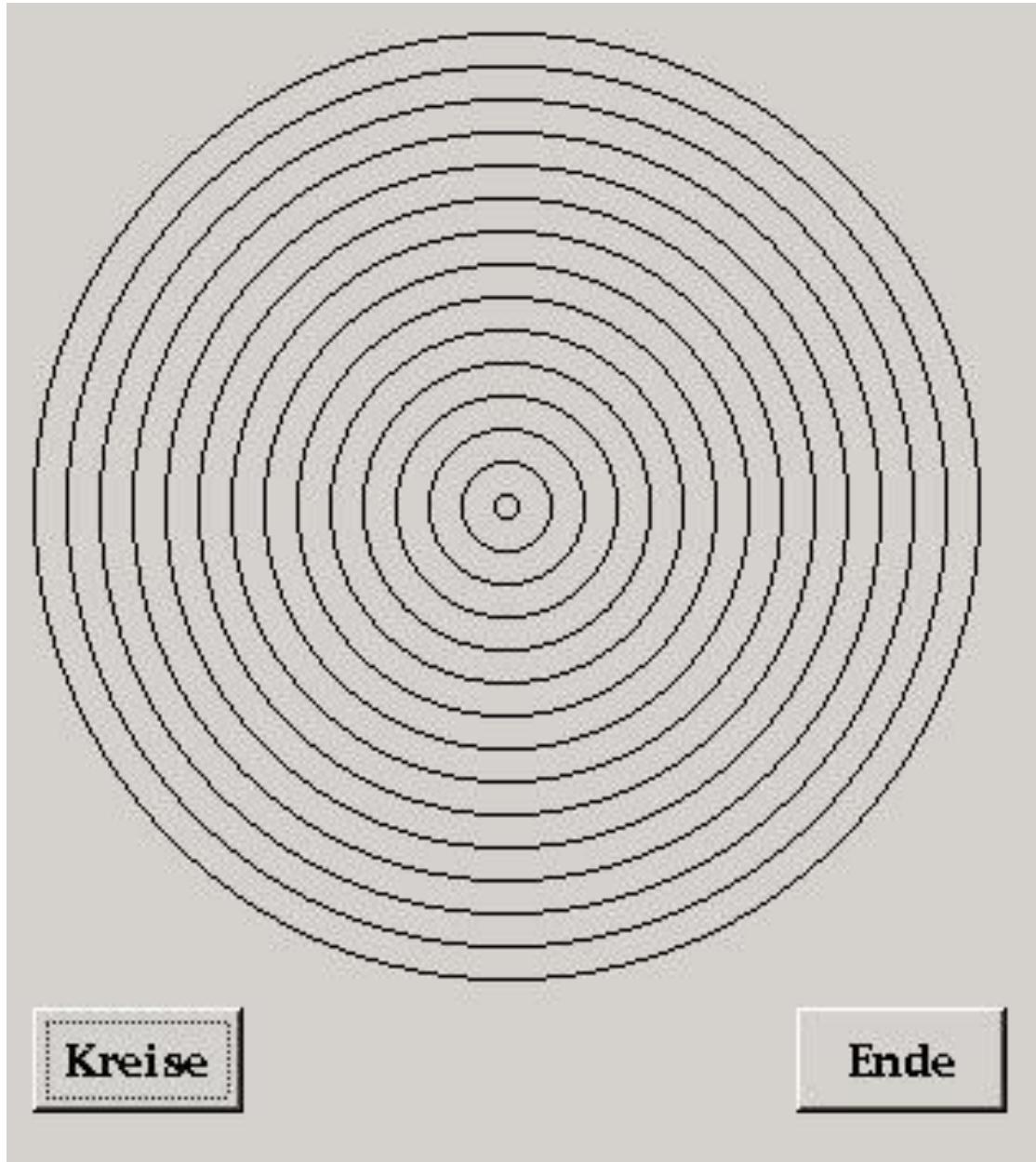


Fig.60

Il secondo pulsante di comando e il relativo codice mancano nel programma precedente.

```
PUBLIC SUB Button2_Click ()  
    ME.Chiudi  
    FINE
```

5.5 Grafico a torta

Se aggiungi altri 2 parametri al comando Disegna (ellisse) oltre ai 4 parametri necessari, otterrai segmenti circolari che possono essere utilizzati bene per un grafico a torta.

- Parametro 5, numero di gradi in cui deve iniziare il segmento
- Parametro 6, numero di gradi in cui il segmento deve terminare.

Esempio:

```
Pulsante PUBLIC SUB1_Click ()  
    Draw.Begin (Drawingarea1)  
    Draw.Ellipse (10, 50, 90, 90, 10, 90)  
    Disegna Fine  
    FINE
```

Ottieni un quarto di cerchio che inizia a dieci gradi a ovest e corre in senso antiorario.

Guarda anche: <http://www.binara.com/gambas-wiki/bin/view/Gambas/ DrawFigures>

5.6 ellisse

Per comprendere i parametri più importanti del comando Draw.Ellipse, puoi provare il seguente programma:

5.6.1 Layout

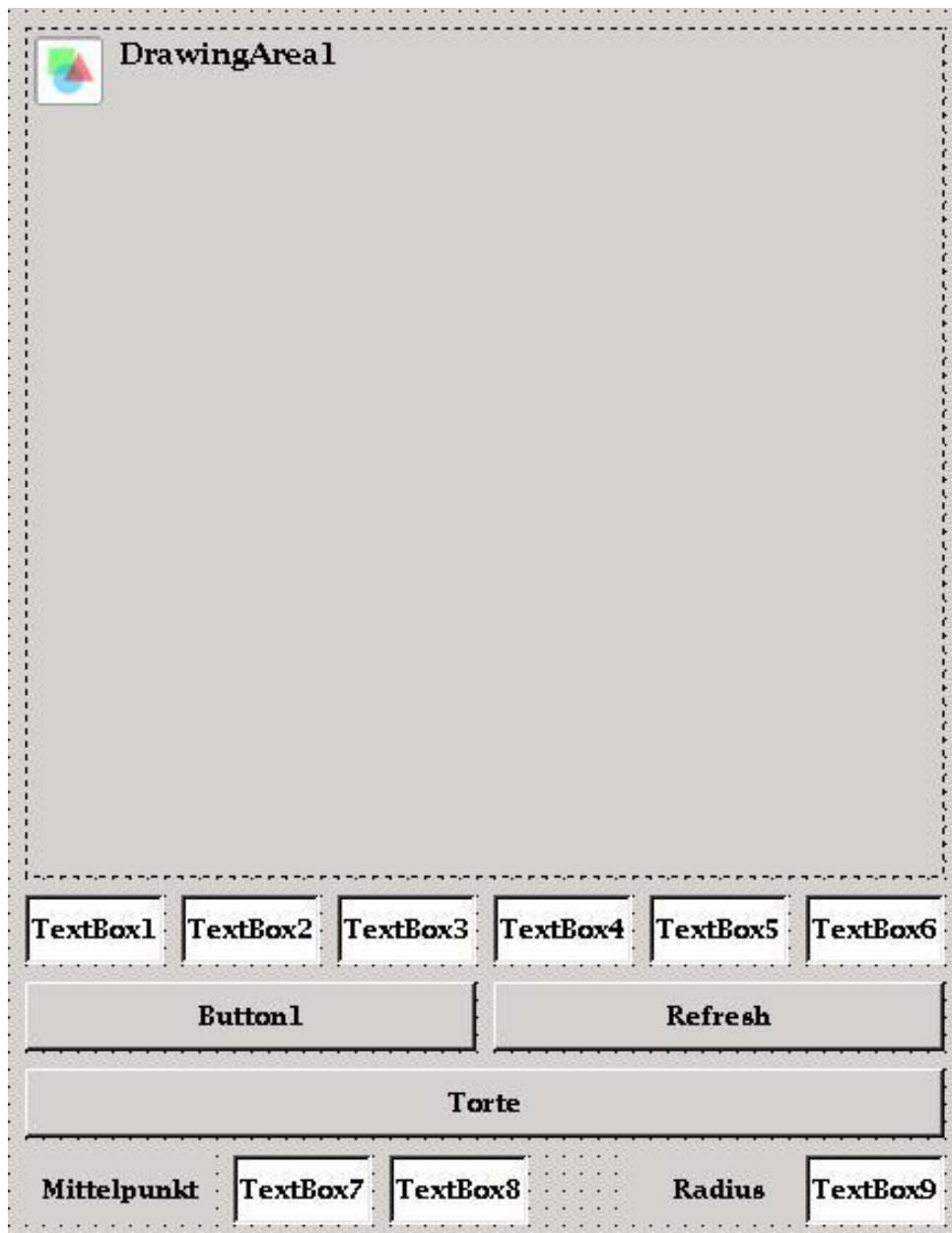


Fig.61

Hai bisogno:

- 9 caselle di testo nel modulo
- 1 area di disegno
- 3 pulsanti di comando
- 2 campi etichetta

sul modulo per avviare il programma.

5.6.2 Il codice

```
PUBLIC SUB Form_Open ()  
    textbox1.Text = "0"  
    textbox2.Text = "0"  
    textbox3.Text = Str (Drawingarea1.Width)  
    textbox4.Text = Str (Drawingarea1.Height)  
    textbox5.Text = "10"  
    textbox6.Text = "10"  
    textbox7.Text = Str (0 + Drawingarea1.Width / 2)  
    textbox8.Text = Str (0 + Drawingarea1.Height / 2)  
    textbox9.Text = Str (Drawingarea1.Width / 2)  
    Button1.Text = "Cerchio"  
    FINE  
    Pulsante PUBLIC SUB1_Click ()  
        un numero intero AS  
        b AS intero  
        c AS intero  
        d AS intero  
        a = Val (Textbox1.text)  
        b = Val (Textbox2.text)  
        c = Val (Textbox3.text)  
        d = Val (Textbox4.text)  
        Draw.Begin (Drawingarea1)  
        Draw.ellipse (a, b, c, d)  
        Disegna Fine  
        textbox7.Text = Str (a + c / 2)  
        textbox8.Text = Str (b + d / 2)  
        textbox9.Text = Str (c / 2)  
        CATTURARE  
        message.Info ("Inserisci un numero in ogni campo!")  
        FINE  
    PUBLIC SUB Button2_Click ()  
        Draw.Begin (Drawingarea1)  
        Drawingarea1.Refresh  
        Disegna Fine  
        FINE  
    Pulsante PUBLIC SUB 3_Click ()  
        un numero intero AS  
        b AS intero  
        c AS intero  
        d AS intero  
        e AS float  
        f AS galleggiante  
        a = Val (Textbox1.text)  
        b = Val (Textbox2.text)  
        c = Val (Textbox3.text)  
        d = Val (Textbox4.text)  
        e = Val (Textbox5.text)  
        f = Val (Textbox6.text)  
        Draw.Begin (Drawingarea1)  
        Draw.ellipse (a, b, c, d, e, f)  
        Disegna Fine  
        CATTURARE  
        message.Info ("Inserisci un numero in ogni campo!")  
        FINE
```

Ed è così che il programma guarda al runtime:

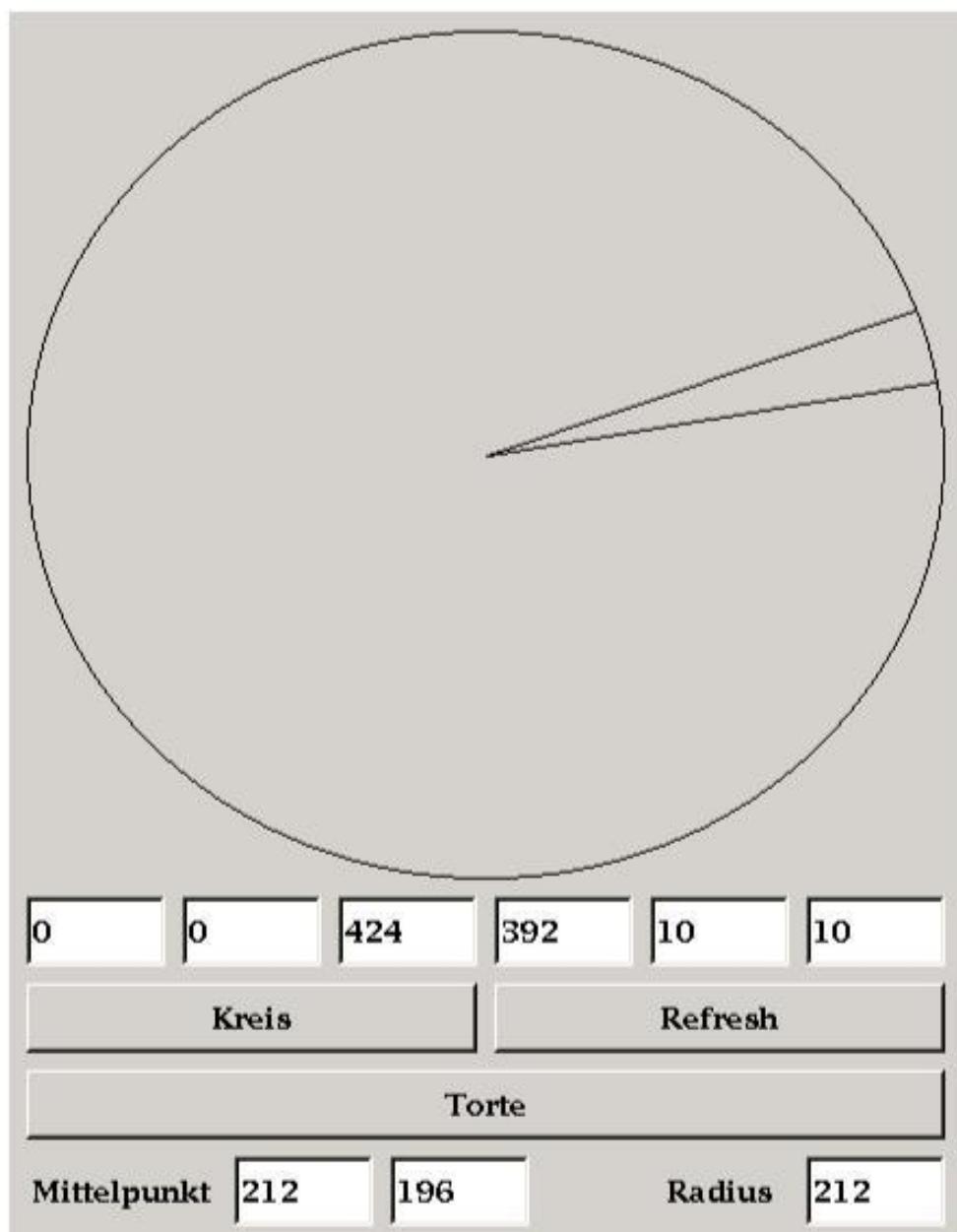


Fig.62

6 La stella marina

"Il piccolo G. non sapeva nemmeno che si potevano trovare stelle così belle sott'acqua."

Il programma Starfish mostra il passaggio da una struttura ordinata al caos. Di solito, le stelle marine hanno effettivamente 5 zampe. Una cosa è andata persa nel nostro programma.

Per avviare il programma, è necessario un DrawingArea e un pulsante di comando.

6.1 Layout

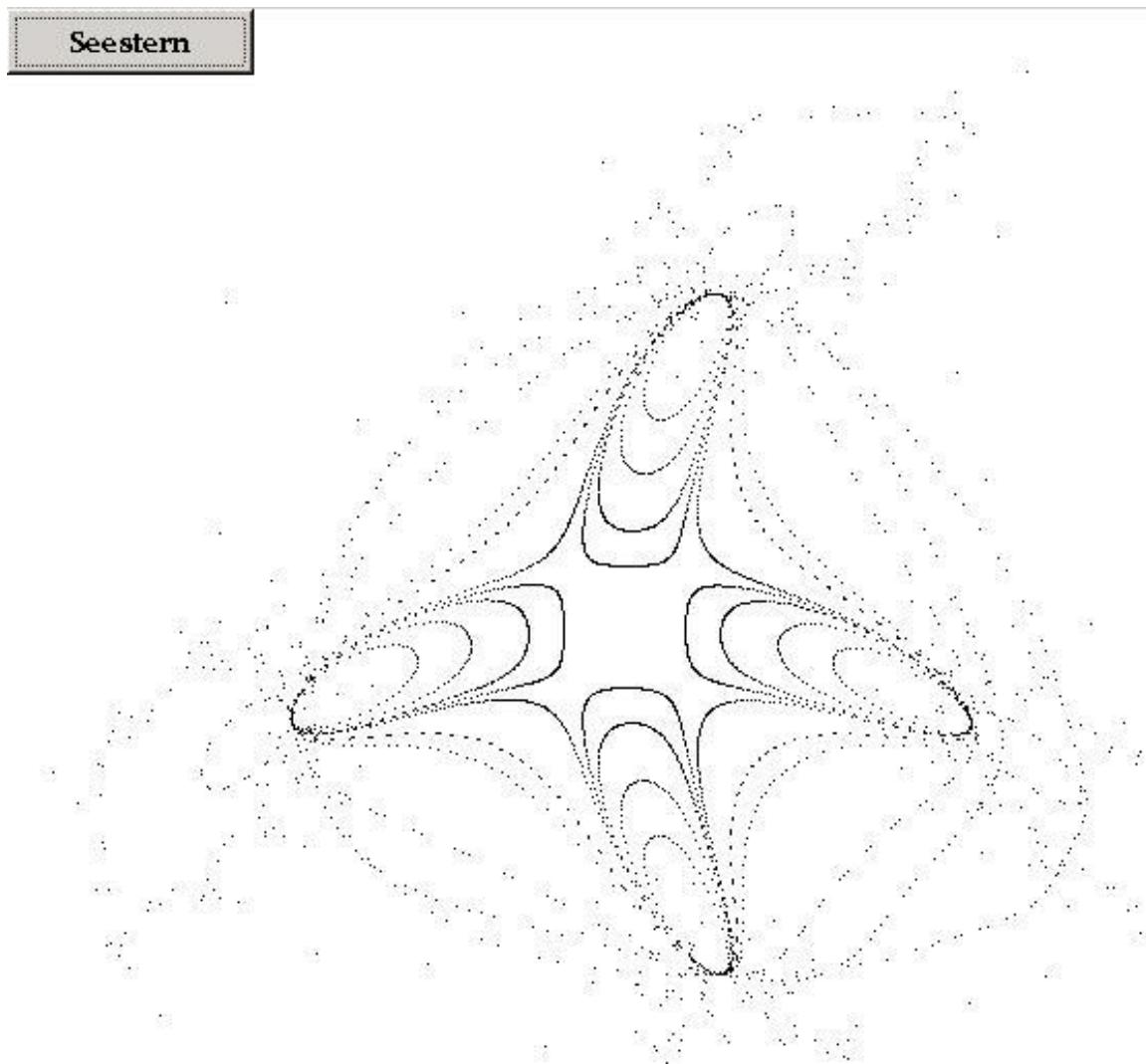


Fig.63

6.2 Il codice

```
PUBLIC SUB Form_Open ()  
    DrawingArea1.Width = 1000  
    DrawingArea1.H = 1000  
    DrawingArea1.X = 0  
    DrawingArea1.Y = 0  
    DrawingArea1.Background = & HFFFFFF &  
    Button1.Text = "Starfish"  
FINE  
Pulsante PUBLIC SUB1_Click ()  
    E AS Integer  
    AW AS galleggiante  
    H AS intero
```

```
X AS galleggiante
Y AS galleggiante
I AS intero
Un galleggiante AS
B AS galleggiante
C AS galleggiante
Draw.Begin (DrawingArea1)
PER E = 1 A 13
SE E = 1 ALLORA
AW = 55
H = 100
FINISCI SE
SE E = 2 ALLORA
AW = 80
H = 120
FINISCI SE
SE E = 3 ALLORA
AW = 88
H = 120
FINISCI SE
SE E = 4 ALLORA
AW = 90
H = 150
FINISCI SE
SE E = 5 ALLORA
AW = 90,1
H = 300
FINISCI SE
SE E = 6 ALLORA
AW = 90,2
H = 490
FINISCI SE
SE E = 7 ALLORA
AW = 90,21
H = 1000
FINISCI SE
SE E = 8 ALLORA
AW = 90,22
H = 600
FINISCI SE
SE E = 9 ALLORA
AW = 90,25
H = 250
FINISCI SE
SE E = 10 ALLORA
AW = 90,3
H = 180
FINISCI SE
SE E = 11 ALLORA
AW = 91
H = 100
FINISCI SE
SE E = 12 ALLORA
AW = 92
H = 100
FINISCI SE
SE E = 13 ALLORA
AW = 92,2
H = 410
FINISCI SE
X = 0,1513
Y = -0,5388
'Pi = 3,141593:
A = AW * Pi / 180 '
'Rem conversione di un angolo di 360 gradi in radianti PER I = 0
TO H
```

```
B = X * Cos (A) - (Y - X * X) * Peccato (A)
C = X * Peccato (A) + (Y - X * X) * Cos (A)
X = B
Y = C
'Draw.Point (400 + E, 400 + E)
Draw.Point (B. * 300 + 300, c * 300 + 300)
'STAMPA B, C
IL PROSSIMO
'FOR z = 1 TO 1000000: NEXT z: Rem time delay can be
variare
IL PROSSIMO
Disegna Fine
FINE
```

6.3 Immagine, Immagine, Area di disegno, Picturebox ecc

6.3.1 Qual è la differenza tra un'immagine e un'immagine?

Sotto Linux, le immagini e la grafica vengono visualizzate con X11. Il sistema X ha un componente di rete. Un'immagine elaborata da un programma può essere visualizzata su un computer completamente diverso nella rete.

Un'immagine in Gambas è un'immagine limitata a un programma. Quindi puoi accedervi più velocemente e fare di più con l'immagine. (Allungarsi, voltarsi, ecc.)

Un'immagine in Gambas è un'immagine che viene visualizzata sotto X, indipendentemente dal fatto che ciò avvenga sulla stessa macchina o altrove nella rete è aperta. Qualsiasi modifica all'immagine è quindi una deviazione tramite una chiamata di rete o un tubo locale. Ecco perché funziona molto più lentamente.

Nei programmi con un uso intensivo delle immagini, passerai da un formato all'altro secondo necessità e convertirai quando risulterà utile.

Il formato dell'immagine è quindi buono per la manipolazione dell'immagine, il formato dell'immagine è più adatto per scopi di visualizzazione puri.

6.4 Area grafica (DrawingArea)

vedi Gambas: _Grafik¹

6.5 Campo immagine (Picturebox)

6.6 Scorri campo di visualizzazione

Il controllo ScrollView è molto utile per lavorare con immagini o grafici. Puoi trovarlo nella casella degli strumenti (F6) accanto a DrawingArea.

¹ Capitolo 0.136.10 a pagina 188



Fig.64

Il campo di visualizzazione a scorrimento può essere utilizzato anche per visualizzare immagini più grandi, poiché fornisce una barra di scorrimento verticale e orizzontale.

Se desideri utilizzare il campo di visualizzazione a scorrimento, devi prima inserirlo nel modulo. Lo rendi un po 'più grande. Per fare ciò, estrarlo più grande nell'angolo in basso a destra. Quindi lascialo segnato. Puoi riconoscerlo dai piccoli rettangoli bianchi sul bordo dell'elemento di controllo. Ora inserisci un DrawingArea o un Picturebox nel campo della visualizzazione a scorrimento. Anche questi vengono tirati un po 'più grandi.

È quindi necessario utilizzare il codice per adattare i due elementi di controllo l'uno all'altro. L'esempio di programma mostra come funziona in Load a picture into a Picturebox. più in basso.

6.7 Carica un'immagine in una Picturebox

Con questo programmino uno screenshot viene caricato in un Picturebox.

Se non sai cos'è uno screenshot, dai un'occhiata qui:

<http://de.wikipedia.org/wiki/Screenshot>

Per avviare il programma, è necessario un modulo, un pulsante di comando, un campo di visualizzazione a scorrimento. Posiziona una casella immagine in questo campo di visualizzazione a scorrimento.



Fig.65

Codice:

```
Pulsante PUBLIC SUB1_Click ()  
    immagine COME NUOVA Immagine  
    immagine = Desktop.Grab ()  
    "Lo screenshot è stato acquisito  
    PictureBox1.Picture = immagine  
    FINE
```

Puoi migliorare questo codice adattando il Picturebox al campo di visualizzazione a scorrimento:

```

Pulsante PUBLIC SUB1_Click ()
immagine COME NUOVA Immagine
'Form1.Hide
'La forma è nascosta
immagine = Desktop.Grab ()
"Lo screenshot è stato acquisito
Picturebox1.Ridimensiona (larghezza foto, altezza foto)
'Rendi il Picturebox grande quanto l'immagine
Picturebox1.X = 0
Picturebox1.y = 0
'Adatta il Picturebox alla Scrollview
PictureBox1.Picture = immagine
'Form1.Show
FINE

```

6.8 Conversione di un'immagine in un'immagine

Nell'esempio seguente viene generato uno screenshot. Questo viene convertito in un'immagine e visualizzato in un DrawingArea.

Per avviare il programma è necessario:

- Un modulo
 - Un pulsante di comando = pulsante di comando
 - Un campo ScrollView = campo con opzioni di movimento verticale e orizzontale
 - Un'area di disegno nel campo della visualizzazione a scorrimento

Il codice:

```

Pulsante PUBLIC SUB1_Click ()
immagine COME NUOVA Immagine
i COME NUOVA immagine
immagine = Desktop.Grab ()
'uno screenshot è preso qui
i = immagine.immagine
'qui l'immagine viene convertita in un'immagine
Drawingarea1.Cached = TRUE
DrawingArea1.Resize (i.Width, i.Height)
DrawingArea1.Clear ()
Draw.Begin (DrawingArea1)
  Draw.Image (i, 0, 0)
Disegna Fine
DrawingArea1.Visible = TRUE
DrawingArea1.Refresh
FINE

```

6.9 Carica un'immagine con finestra di dialogo in una casella immagini

Con il seguente programma è possibile creare un'immagine con il seguente formato: jpg; jpeg; png; Carica bmp in un Picturebox con una finestra di dialogo.

Per avviare il programma è necessario

- una casella delle immagini dalla casella degli strumenti
- un pulsante di comando nel Picturebox

Codice:

```
Pulsante PUBLIC SUB1_Click ()  
p COME da foto  
Dialog.Path = "/ usr / share / wallpapers"  
SE NON esiste (Dialog.Path) ALLORA  
    Dialog.Path = System.Home  
FINISCI SE  
  
Dialog.Filter = ["File immagine (*.jpg; *.jpeg; *.png; *.bmp) "," Tutti i file (*.*) "]  
IF Dialog.OpenFile () THEN RETURN  
p = NUOVA immagine  
p.Load (Dialog.Path)  
Picturebox 1. Ridimensiona (p. Larghezza, p. Altezza)  
Picturebox1.Picture = p  
PictureBox1.Move (0, 0, ME.ClientW, ME.ClientH)  
Picturebox1.Visible = TRUE  
CATTURARE  
    Message.Warning (Error.Text & "!" )  
FINE
```

Con questo programma noc disturba il fatto che l'immagine venga visualizzata due volte durante il caricamento. Come puoi evitarlo?

6.10 Carica un'immagine nel Picturebox senza finestra di dialogo

Attenzione. Questo esempio è un po 'complicato.

Il seguente programma può essere utilizzato per caricare un'immagine in Picturebox. Il programma si basa sul programma di esempio blights, fornito con Gambas.

Inizia un nuovo progetto. Ottieni una nuova forma con il tasto destro del mouse sulla forma nel progetto. Chiama la forma come vuoi. Posizionare 2 pulsanti di comando e una casella immagine nel form. Nella directory del programma 2 salvate i file immagine blon.xpm e bloff.xpm, che potete trovare sotto blights negli esempi Gambas.

Se non riesci a trovare le immagini, prendi altri file di immagini o dai un'occhiata nella seguente directory: / opt / gambas / share / gambas / examples / Basic

Con i pulsanti di comando On e Off è ora possibile caricare immagini diverse in PictureBox1.

Il codice del programma ha questo aspetto:

```
"File di classe Gambas  
Pulsante PUBLIC SUB1_Click ()  
PictureBox1.Picture = Immagine ["bloff.xpm"]  
FINE  
-----  
PUBLIC SUB Button2_Click ()  
PictureBox1.Picture = Immagine ["blon.xpm"]  
FINE
```

Salva immagine senza finestra di
6.11 dialogo

Salva immagine con finestra di
6.12 dialogo

6.13 Ottieni un'immagine dagli appunti

6.14 Carica l'immagine negli appunti

6.15 Ridimensiona

Diversi elementi grafici di Gambas conoscono il metodo di ridimensionamento. Come suggerisce il nome, puoi usarlo per modificare le dimensioni di un oggetto grafico sullo schermo.

L'esempio seguente mostra come funziona:

```
PUBLIC SUB Form_Open ()  
    Textbox2.Text = "200"  
    Textbox3.Text = "200"  
    Button1.Text = "Ridimensiona Button1"  
    Button1.Width = 200  
    Button1.Height = 50  
FINE  
Pulsante PUBLIC SUB1_Click ()  
    Button1.Resize (Val (textbox1.text), Val (textbox2.text))  
FINE
```

Per avviare il programma sono necessarie 2 caselle di testo e un pulsante di comando sul modulo. Se si preme il pulsante di comando dopo aver avviato il programma, la sua dimensione viene modificata in base ai valori nelle 2 caselle di testo. Quando il programma viene avviato, la subroutine Form_Open viene elaborata e rende alcune utili impostazioni predefinite. Nella subroutine Button1.Click, ridimensionato, ovvero la dimensione viene modificata. In alternativa puoi ovviamente inserire anche:

```
Button1.Width = Val (textbox1.text)  
Button1.Height = Val (textbox2.text)
```

Quindi puoi ridimensionare senza il metodo di ridimensionamento.

Esempio 2: ridimensionamento automatico di un'area di testo in una forma:

```
PUBLIC SUB Form_Resize ()  
    TextArea1.Move (0, 0, ME.ClientWidth, ME.ClientHeight)  
FINE
```

Hai bisogno di un modulo e un'area di testo per avviare il programma. Non importa dove sia l'area di testo, è adattata alla forma. Anche se continui a spostare la forma, l'area di testo rimarrà fedele alla forma.

6.16 Memorizzato nella cache

6.17 Aggiorna

6.18 Screenshot

Con questo programmino uno screenshot viene caricato in un Picturebox.

Se non sai cos'è uno screenshot, dai un'occhiata qui:

<http://de.wikipedia.org/wiki/Screenshot>

Per avviare il programma, è necessario un modulo, un pulsante di comando, un campo di visualizzazione a scorrimento. Posiziona una casella immagine in questo campo di visualizzazione a scorrimento.

Codice:

```
Pulsante PUBLIC SUB1_Click ()  
    immagine COME NUOVA Immagine  
    immagine = Desktop.Grab ()  
    "Lo screenshot è stato acquisito  
    PictureBox1.Picture = immagine  
FINE
```

6.19 colori

È possibile modificare facilmente il colore di diversi oggetti tramite la finestra delle proprietà dell'oggetto (primo piano, sfondo, ecc.). La finestra delle proprietà viene aperta in modalità bozza come al solito con F4.

È inoltre possibile modificare le impostazioni del colore durante il runtime. Questo è inteso di seguito

Viene mostrato un esempio di programma:

Questo mini programma mostra i colori RGB. È possibile immettere valori da 0 a 255 nelle caselle di testo e osservare il colore. Un valore esadecimale viene creato come una stringa dai 3 numeri decimali:

```
sHEX = Hex $(r, 2) & Hex $(g, 2) & Hex $(b, 2)  
SHEX = "& H" & SHEX & "&"
```

Se vuoi far funzionare il programma, hai bisogno dei seguenti campi sul tuo modulo, che devi ottenere dalla casella degli strumenti:

- 3 campi di etichette di testo
- 3 campi di testo (Textbox 1-3) per l'immissione del valore RGB
- 1 campo di comando (= pulsante di comando)
- 1 area di disegno (= DrawingArea) per la rappresentazione del colore

L'intera cosa assomiglia a questo:



Fig.66

Codice del programma

```
Pulsante PUBLIC SUB1_Click ()
DIM sHex AS String
DIM r AS Integer
DIM g AS Integer
DIM b AS Integer
IF Textbox1.text = "" THEN Textbox1.Text = 0
IF Textbox2.text = "" THEN Textbox2.Text = 0
IF Textbox3.text = "" THEN Textbox3.Text = 0
r = Val (Textbox1.Text)
g = Val (Textbox2.Text)
b = Val (Textbox3.Text)
sHEX = Hex $(r, 2) & Hex $(g, 2) & Hex $(b, 2)
SHEX = "& H" & SHEX & "&"
```

```
DrawingArea1.BackColor = Val (sHEX)
FINE
```

6.20 spettro dei colori

Lo spettro cromatico naturale della luce varia dal rosso scuro all'arancio, dal giallo, dal verde al blu e al viola.

colore	Lunghezza d'onda	RGB
rosso	~ 625-740 nm	#DF0000
arancia	~ 590-625 nm	#FF8000
giallo	~ 565-590 nm	#FFFF00
verde	~ 520-565 nm	#00FF00
Ciano	~ 500-520 nm	#00FFFF
blu	~ 450-500 nm	#0000FF
indaco	~ 430-450 nm	#0000AF
viola	~ 380-430 nm	#50007F

Se vuoi guardare questo spettro di colori, il seguente programma ti aiuterà. Può essere migliorato se usi un cursore per selezionare i colori.

Hai bisogno di un modulo, un DrawingArea e un pulsante di comando per il programma. Se si preme più volte il pulsante di comando, il colore continua a cambiare. Inizia con il rosso e finisce con il viola.

Codice del programma

```
sp AS intero
PUBLIC SUB Form_Open ()
sp = 1
drawingarea1.BackColor = & HC2020C &
FINE
Pulsante PUBLIC SUB1_Click ()
DIM ss AS intero
ss = 1 s
= sp
drawingarea1.BackColor = Val (spettro (ss)) sp = sp +
20
'Qui puoi scegliere i salti di colore, aggiungerli a sp 5,10,20,50 o 100
SE sp> 1120 ALLORA sp = 1
'se è viola torna al rosso
FINE
FUNZIONE PUBBLICA spettro (ss AS Integer) AS String x AS
Integer
sHEX AS String DIM
s AS Interger DIM r AS
Interger DIM g AS
Interger DIM b AS
Interger s = 0

PER x = da 1 a 255
'dal rosso al giallo
```



```

r = 255
b = 0
sHEX = Hex $(r, 2) & Hex $(x, 2) & Hex $(b, 2)
SHEX = "& H" & SHEX & "&"
s = s + 1
SE s = ss ALLORA RITORNA sHEX
IL PROSSIMO
's = 255
PER x = 255TO 1 PASSO -1
'dal giallo al verde
g = 255
b = 0
sHEX = Hex $(x, 2) & Hex $(g, 2) & Hex $(b, 2)
SHEX = "& H" & SHEX & "&"
s = s + 1
'PRINT s
SE s = ss ALLORA RITORNA sHEX
IL PROSSIMO
's = 510
PER x = 1TO 255
'dal verde al ciano
r = 0
g = 255
sHEX = Hex $(r, 2) & Hex $(g, 2) & Hex $(x, 2)
SHEX = "& H" & SHEX & "&"
s = s + 1
'PRINT s
SE s = ss ALLORA RITORNA sHEX
IL PROSSIMO
's = 765
PER x = 255 A 1 PASSO - 1
'dal ciano al blu
r = 0
b = 255
sHEX = Hex $(r, 2) & Hex $(x, 2) & Hex $(b, 2)
SHEX = "& H" & SHEX & "&"
s = s + 1
'PRINT s
SE s = ss ALLORA RITORNA sHEX
IL PROSSIMO
's = 1020
b = 255
PER x = 1TO 100
'dal blu al viola
g = 0
sHEX = Hex $(x, 2) & Hex $(g, 2) & Hex $(b, 2)
SHEX = "& H" & SHEX & "&"
b = b - 2
s = s + 1
'PRINT s
'STAMPA shex
SE s = ss ALLORA RITORNA sHEX
IL PROSSIMO
's = 1120
FINE

```

6.21 Come usare MouseDown, MouseMove e MouseUp

Quando si preme il pulsante sinistro del mouse, viene avviato un evento mousemove a intervalli regolari.

Il programma

La stella marina

Ottieni una nuova forma e chiamala Fmaus. Dichiaralo come modulo di partenza.

Prendi 3 aree di disegno dalla casella degli strumenti e assegna loro un nome in Proprietà: da1, da2 e da3 intorno. Fai una copia di quello indicato di seguito. Codifica e compila il programma. È in esecuzione? In tal caso, sperimenta con il Mouse e colori intorno.

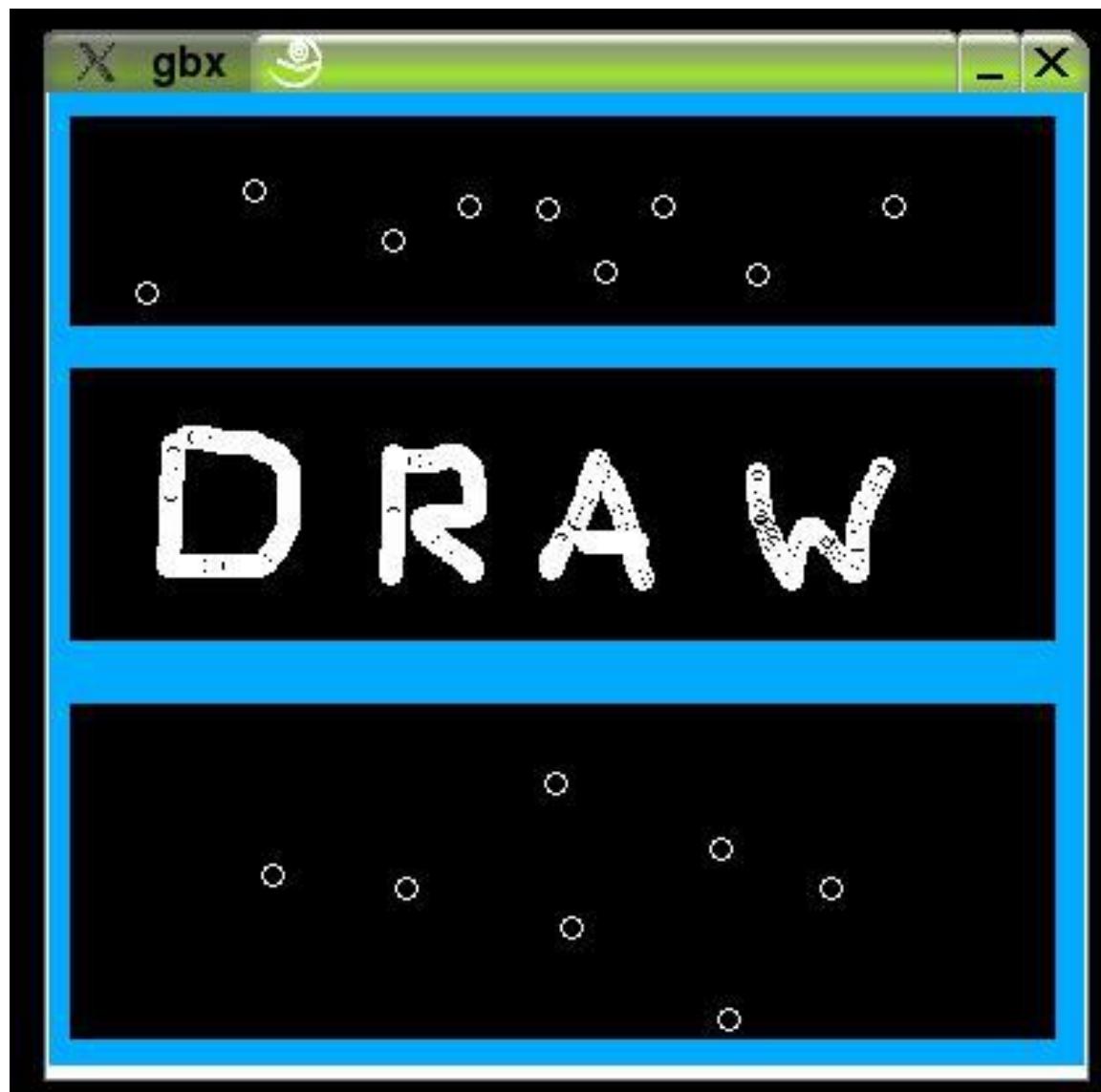


Fig.67

Fare clic nei campi neri! Nel secondo campo devi tenere premuto il pulsante del mouse. Nel terzo campo devi rilasciare il mouse.

Il codice:

```
STATICO PUBBLICO SUB Principale ()  
hForm AS Fmaus
```

```
hForm = NUOVO Fmaus
hForm.show
FINE
PUBLIC SUB da1_MouseDown ()
    Draw.Begin (da1)
        Draw.Ellipse (Mouse.X, Mouse.Y, 9, 9)
    Disegna Fine
FINE
PUBLIC SUB da2_MouseMove ()
    Draw.Begin (da2)
        Draw.Ellipse (Mouse.X, Mouse.Y, 9, 9)
    Disegna Fine
FINE
PUBLIC SUB da3_MouseUp ()
    Draw.Begin (da3)
        Draw.Ellipse (Mouse.X, Mouse.Y, 9, 9)
    Disegna Fine
FINE
```

6.22 Le coordinate del mouse

Il seguente programma funziona solo in un campo grafico (= DrawingArea).

Il programma fornisce continuamente le coordinate del mouse spostandole sul campo grafico e tenendo premuto il tasto sinistro del mouse.

Fai un nuovo progetto. Apri una nuova forma: fai clic su Forma con il pulsante destro del mouse. Dagli un nome. Ottieni un DrawingArea = area di disegno dalla casella degli strumenti.

Ottieni 2 caselle di testo dalla casella degli strumenti.

La forma quindi assomiglia a questo:

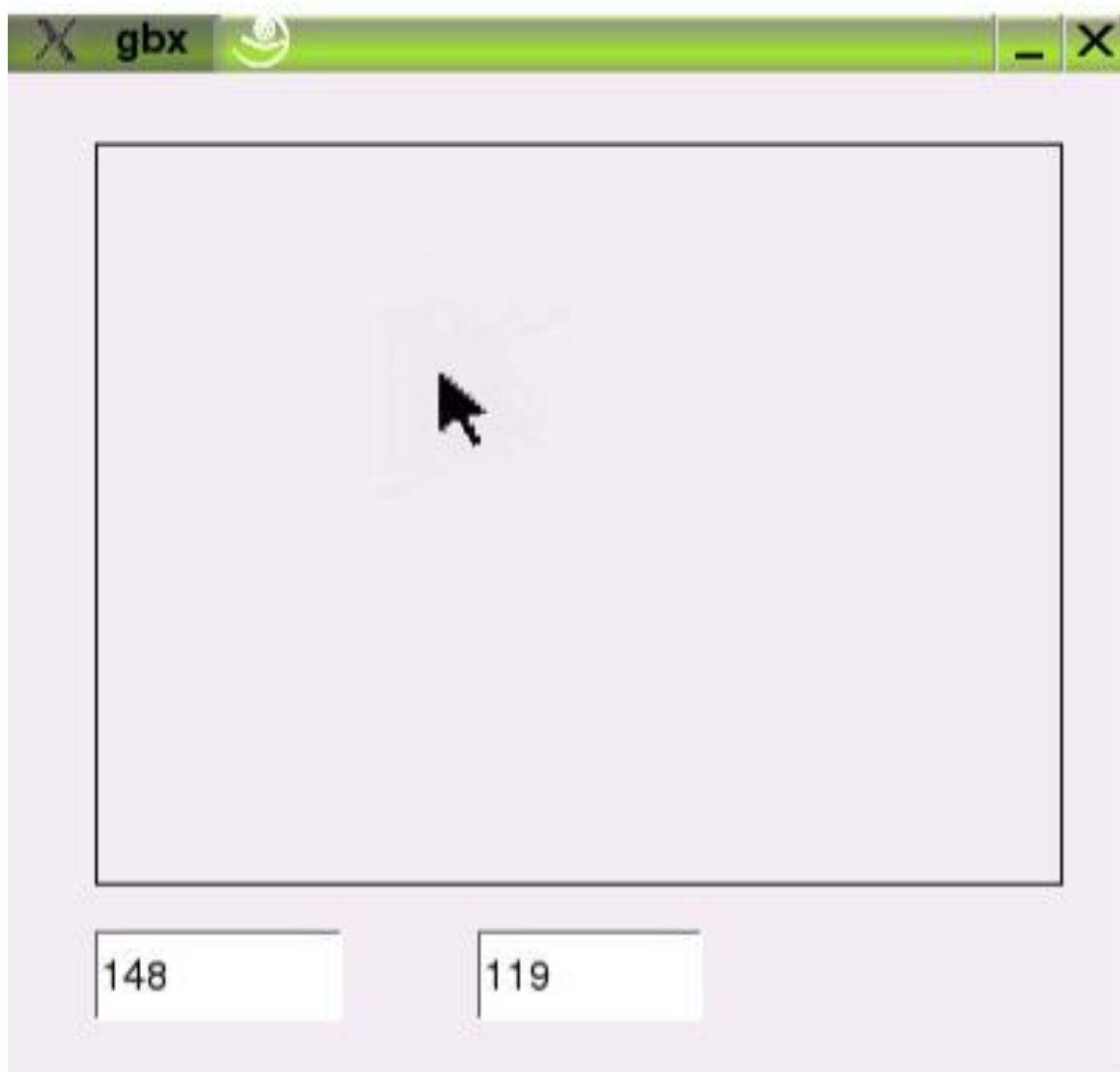


Fig.68

Copia il codice dal basso nel programma. Avvia il programma con F5. Presta attenzione alla colorazione delle proprietà, altrimenti non vedrai nulla. Prestare attenzione anche alla colorazione della forma. Le proprietà del colore della forma vengono trasferite alle proprietà dei componenti della forma.

Muovi il mouse nell'area di disegno e premi il pulsante sinistro del mouse. Quindi le coordinate del mouse vengono visualizzate nelle caselle di testo.

Notare quanto segue: l'angolo superiore sinistro è (0,0), l'angolo inferiore destro è (Drawin-gArea1.Width, DrawingArea1.Height)

Il nuovo codice:

```
PUBLIC SUB DrawingArea1_MouseMove ()  
Textbox1.text = Mouse.X
```

```
Textbox2.text = Mouse.Y  
FINE
```

Come si ottengono le coordinate assolute?

Il vecchio codice: Gambas più vecchio di 0.65

```
PUBLIC SUB DrawingArea1_MouseMove (X AS Integer, Y AS Integer, Button  
AS intero)  
Textbox1.text = X  
Textbox2.text = Y  
FINE
```

6.23 Drag and drop, prendere e trascinare con il mouse.

Il programma mostra le proprietà di trascinamento della selezione. Trascini un'immagine avanti e indietro sulla forma con il mouse finché premi il pulsante sinistro del mouse.

Il programma funziona con un campo immagine. Ma puoi anche spostare altri oggetti allo stesso modo.

Fai un nuovo progetto. Apri una nuova forma: fai clic su Forma con il pulsante destro del mouse. Dagli un nome. Usa F6 per ottenere un'immagine dalla casella degli strumenti e lascia il nome come Immagine1.

La forma quindi assomiglia a questo:

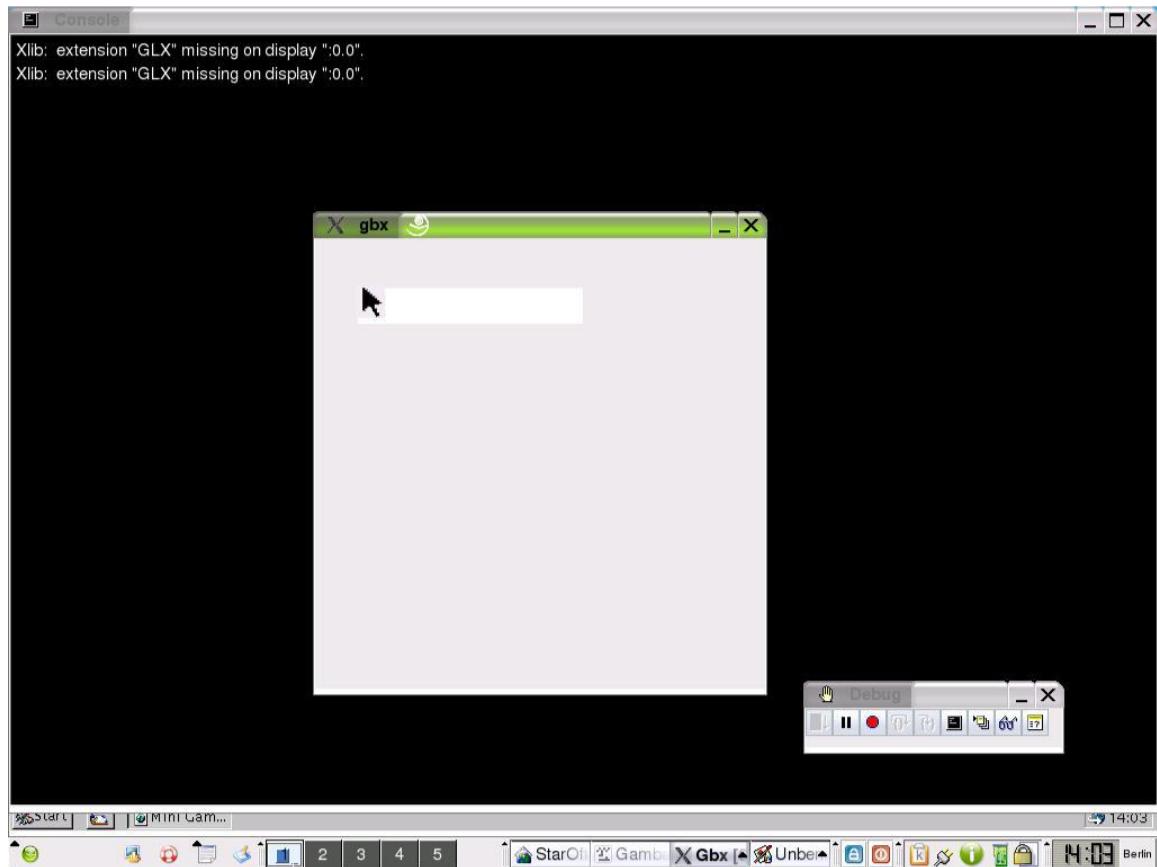


Fig.69

Copia il codice dal basso nel programma. Avvia il programma con F5. Presta attenzione alla colorazione delle proprietà, altrimenti non vedrai nulla. Prestare attenzione anche alla colorazione della forma. Le proprietà del colore della forma vengono trasferite alle proprietà dei componenti della forma.

Muovi il mouse nell'area di disegno e premi il pulsante sinistro del mouse. Quindi il campo dell'immagine viene spostato in base alle coordinate del mouse.

Il codice:

```
PUBLIC SUB Form_MouseMove ()  
Image1.X = Mouse.X  
Image1.Y = Mouse.Y  
FINE
```

Se vuoi che l'immagine passi al mouse con il primo clic del mouse, aggiungi il seguente codice:

```
PUBLIC SUB Form_MouseDown ()  
Image1.X = Mouse.X  
Image1.Y = Mouse.Y  
FINE
```

vecchio codice

```
PUBLIC SUB Form_MouseMove (X AS Integer, Y AS Integer, Button AS  
Numero intero)  
Image1.X = X  
Image1.Y = Y  
FINE
```

6.24 Interrogare i pulsanti del mouse

Il pulsante sinistro del mouse viene interrogato nel seguente programma. Se lo premi, il mouse dipinge una forma rotonda. Se si preme il pulsante destro, per dipingere viene utilizzata una forma rettangolare.

È necessaria un'area di disegno sul modulo per avviare il programma.

```
PUBLIC SUB Form_Open ()  
DrawingArea1.Cached = FALSE  
DrawingArea1.BackColor = & HFFFFFF &  
'Sfondo bianco  
FINE  
PUBLIC SUB DrawingArea1_MouseMove ()  
IF Mouse.Button = 1 THEN  
    Draw.Begin (DrawingArea1)  
    Draw.Ellipse (Mouse.X, Mouse.Y, 9, 9)  
    Disegna Fine  
ALTRO  
    Draw.Begin (DrawingArea1)  
    Draw.Rect (Mouse.X, Mouse.Y, 9, 9)  
    Disegna Fine  
FINISCI SE  
FINE
```

6.25 Definire una diversa forma del cursore tramite le proprietà

È possibile definire la forma del puntatore del mouse (cursore) nelle proprietà di alcuni elementi di controllo. Se prendi un modulo vuoto e guardi le sue proprietà con il tasto F4, c'è una riga chiamata Mouse al centro della tabella delle proprietà. Se fai clic su questa riga nella parte destra, ottieni una selezione di diverse forme di mouse:

- Freccia
- Vuoto
- Attraversare
- Personalizzato
- predefinito
- Orizzontale
- Indicando
- SizeAll
- Taglia E.
- Taglia H.

- TagliaN
- TagliaNE
- TagliaNESW
- TagliaNW
- TagliaNWSE
- TagliaS
- TagliaSE
- Taglia SW
- TagliaV
- TagliaW
- SplitH
- SplitV
- testo
- Verticale
- Aspettare

Prova diverse forme di mouse e avvia semplicemente il programma con F5, vedrai quindi la forma del cursore del mouse selezionata.

La forma del puntatore può essere selezionata nuovamente tramite alcuni elementi di controllo oppure è già preimpostata.

È inoltre possibile modificare il puntatore del mouse nella sequenza del programma. Il seguente programma mostra come eseguire questa operazione: Sono necessari un modulo e 2 pulsanti di comando per farlo funzionare.

```
Pulsante PUBLIC SUB1_Click ()  
    Form1.Mouse = Mouse.Wait  
    FINE  
PUBLIC SUB Button2_Click ()  
    Form1.Mouse = Mouse.Pointing  
    FINE
```

6.26 Il mouse casuale

Puoi guardare tutte le forme del puntatore del mouse se provi il seguente programma.

Per farlo funzionare, tutto ciò di cui hai bisogno è un timer sul tuo modulo.

```
PUBLIC SUB Form_Open ()  
    timer1.Enabled = TRUE  
    timer1.Delay = 300  
    FINE  
PUBLIC SUB Timer1_Timer ()  
    x AS intero  
    Mouse AS intero  
    Rendi casuale  
    x = Int (Rnd (1.15))  
    form1.Mouse = x - 2  
    FINE
```

All'avvio del programma, il timer viene attivato e la sua temporizzazione (Delay) impostato su un valore medio di 300. I numeri casuali compresi tra 1 e 15 vengono generati nella subroutine del timer. Questi numeri casuali vengono assegnati alla proprietà del puntatore del mouse. Pertanto, la forma del mouse varia in modo casuale tutte le forme possibili. Le proprietà del mouse possono essere trovate sotto il componente Gambas Gb.QT.

La coincidenza nel programma di cui sopra ti dà fastidio? Quindi prova la seguente variante:

```
Pulsante PUBLIC SUB1_Click ()  
x AS intero  
Mouse AS intero  
PER x = da -1 a 13  
    form1.Mouse = x  
    ATTENDERE 0.5  
    Button1.Text = "Mouse" & Str (x)  
IL PROSSIMO  
FINE
```

In questo programma, tutte le forme del mouse vengono provate con un ritardo di mezzo secondo (Wait 0.5). Per avviare il programma è necessario un modulo e un pulsante di comando.

La forma del puntatore del mouse può essere impostata utilizzando i seguenti valori costanti, se la proprietà del mouse è selezionata con

```
Mouse AS intero
```

Sono definiti.

- Valore predefinito = -1
- Freccia = 0
- Croce = 2
- Aspetta = 3
- Testo = 4
- Taglia S = 5
- TagliaE = 6
- TagliaNESW = 7
- TagliaNWSE = 8
- SizeAll = 9
- Vuoto = 10
- SplitV = 11
- SplitH = 12
- Puntamento = 13

6.27 Un'altra immagine del cursore con il comando del cursore

È possibile assegnare un nuovo simbolo al cursore con il comando Cursore. Il seguente programma di esempio dovrebbe mostrare come eseguire questa operazione.


```
PUBLIC SUB Form_Open ()  
    hPict COME Immagine  
    hCursore AS cursore  
    hPict = Immagine ["neuepfeil.png"]  
    hCursor = NUOVO Cursore (hPict)  
    ME.Cursor = hCursor  
    FINE
```

Devi ricreare il file immagine neuepfeil.png con un programma di pittura e salvarlo nella directory in cui si trova il tuo programma gambas. Se manca il file immagine, viene visualizzato un messaggio di errore Oggetto null, poiché manca l'oggetto immagine da caricare per la classe Cursor.

6.27.1 Teoria

Dietro il comando Cursor c'è una classe in cui qualsiasi immagine può essere caricata come puntatore del mouse. La classe può essere generata in fase di esecuzione. Questo viene fatto anche nel programma di esempio sopra. Il comando crea un nuovo cursore da un oggetto immagine. La punta del mouse può essere definita in modo più preciso, generalmente viene utilizzato l'angolo superiore sinistro dell'immagine del mouse.

La sintassi per il cursore è quindi:

```
hCursor = NEW Cursor (Picture AS Picture [, X AS Integer, Y AS Integer])
```

X e Y sono le coordinate della punta del mouse (hot spot)

6.28 domanda

Come puoi cambiare il puntatore del mouse nella forma dell'orologio (occupato o in attesa) e viceversa?

Soluzione:

```
ME.Mouse = Mouse.Wait  
...  
ME.Mouse = Mouse.Normal
```

6.29 Emissione di testo

L'output di testo può essere eseguito con il comando di stampa nella finestra diretta o in una casella di testo o area di testo.

Il programma Gambas più breve con output di testo è

```
PUBLIC SUB Form_Open ()
STAMPA 1
FINE
```

Hai bisogno di un modulo vuoto per avviarlo.

Inizia un nuovo progetto. Chiamalo come vuoi. Ottieni una nuova forma con il tasto destro del mouse. Chiamali come vuoi. Puoi anche usare semplicemente F.

Quindi copiare il codice del programma nell'area del codice del programma Gambas Form.class.

Avvia il programma con F5

Questo programma utilizza il comando di stampa. Tutto ciò che fa il programma: visualizza il numero uno nella finestra diretta.

Se si desidera modificare il programma per ottenere l'output di testo in una casella di testo, il codice avrà questo aspetto:

```
PUBLIC SUB Form_Open ()
Textbox1.Text = "1"
FINE
```

Se provi ad avviare il programma con un form vuoto, riceverai un messaggio di errore.

```
Identificatore sconosciuto: textbox1 riga 5 in Form1.class
```

Quindi hai bisogno di una casella di testo nel modulo. Puoi ottenerlo dalla casella degli strumenti con F6 e fare doppio clic sulla casella di testo.

6.29.1 Stampa

Il buon vecchio comando di stampa è andato un po 'fuori moda. Tuttavia, puoi ancora usarlo molto bene. Il comando print reindirizza l'output del programma allo standard output (solitamente la console di testo).

Questo è diverso da Visual Basic. Lì il comando di stampa ha emesso il suo risultato nel modulo. Lo ha fatto indipendentemente dalle cose che altrimenti si sarebbero trovate nella forma. Di conseguenza, l'output era spesso difficile da leggere. Aveva anche lo svantaggio di non essere in grado di rompere le linee.

È possibile utilizzare molto bene il comando print in Gambas per la risoluzione dei problemi se lo si utilizza per reindirizzare il valore delle variabili alla console di testo e visualizzarle. Molti testi di aiuto per Gambas utilizzano anche il comando print.

Ora prova il seguente esempio:

```
Pulsante PUBLIC SUB1_Click ()
DIM k AS Integer
PER k = da 1 a 10
PRINT k;
```

```
IL PROSSIMO  
STAMPA  
STAMPA 1,2,3,4,5,6,7,8,9,10  
PER k = da 1 a 10  
    STAMPA k  
IL PROSSIMO  
FINE
```

Ottieni una nuova forma. Posiziona un pulsante di comando su di esso. Puoi ottenerlo dalla casella degli strumenti con F6. In modalità progettazione, fare clic sul pulsante di comando e assegnargli il codice del programma di seguito. Se il programma mostrato qui è ancora troppo complicato per te, inserisci semplicemente Print 1 o Print "Hello world!"

Avvia il programma con F5 e premi il pulsante di comando, quindi dovresti fare quanto segue

Ricevi output:

```
12345678910  
1   2   3   4 °   5   6 °   7 °   8 °   9   10  
1  
2  
3  
4 °  
5  
6 °  
7 °  
8 °  
9  
10
```

Termini il programma con un clic sul rombo nell'angolo in alto a destra.

Il comando di stampa emette le stampe sul dispositivo di output standard. Le espressioni vengono prima convertite in un output di testo con la funzione stringa Str () .

Il testo deve essere tra virgolette per il comando di stampa:

Print "Hello World" è corretto. Print Hello World non funziona.

Le virgolette possono essere omesse per i numeri interi. Print "1" è identico a print 1.

Se non sono presenti virgole o punti e virgola dopo l'ultimo carattere, viene aggiunto automaticamente un nuovo comando di riga.

Se nel comando di stampa viene visualizzata una virgola, viene inserito un salto di tabulazione (codice ASCII 9). Di solito il salto di tabulazione corrisponde a 5 spazi.

I due punti dopo il comando di stampa non funzionano in Gambas.

```
Stampa 1: Stampa 2
```

dà un messaggio di errore.

Errore di sintassi

6.29.2 Stringa \$

Il comando String \$ può essere utilizzato per produrre più volte un carattere o una stringa di caratteri

Sintassi:

```
String = String $ (frequenza, stringa)
```

Esempio:

```
PRINT String $ (12, "*")
```

Produzione: *****

```
PRINT String $ (2, "Gambas")
```

Edizione: GambasGambas

6.30 inserimento del testo

Il testo viene inserito tramite una casella di testo o un'area di testo.

È possibile limitare l'immissione di testo all'immissione di determinati caratteri. L'esempio seguente mostra come funziona. Utilizza il comando STOP EVENTO:

È necessaria una casella di testo nel modulo per avviare il programma:

```
"La mia casella di testo elabora solo cifre.  
PUBLIC SUB TextBox1_KeyPress ()  
    IF Instr ("0123456789", Key.Text) = 0 THEN  
        STOP EVENTO  
    FINISCI SE  
END SUB
```

Puoi ottenere qualcosa di simile con il seguente codice. Ciò che è interessante qui è l'uso inizialmente incomprensibile del comando If Then:

```
PUBLIC SUB Form_Open ()  
    ME.Text = "Sono consentiti solo numeri!"  
    FINE  
PUBLIC SUB TextBox1_KeyPress ()  
    IF key.Code >= 48 AND key.Code <= 57 THEN
```

```
ELSE IF key.Code = key.BackSpace THEN  
ELSE IF key.Code = key.Delete THEN  
ALTRO  
STOP EVENTO  
FINISCI SE  
FINE
```

6.31 Elimina testo

Con il seguente programmino puoi cancellare una casella di testo vuota. È piuttosto semplice. Ottieni una casella di testo dalla casella degli strumenti. Quindi un altro pulsante di comando. Per favore, riferisci <http://www.madeeasy.de/7/prgtxtdel.htm>

```
Pulsante PUBLIC SUB1_Click ()  
TextBox1.Text = ""  
FINE
```

6.32 Incollare il testo insieme a &

Puoi incollare il testo in Gambas con il comando &. Troverai il simbolo sopra il numero 6. Non puoi semplicemente incollare il testo insieme al simbolo più, perché a differenza di VB, il + in Gambas è riservato solo all'addizione matematica.

Il programma di esempio incolla 2 parti di testo insieme e le invia in una TextArea. È abbastanza facile. Ottieni tre TextAreas dalla casella degli strumenti. Poi altri due pulsanti. Rinomina i pulsanti nelle Proprietà (= richiama le proprietà con il tasto F4).

Per favore, riferisci <http://www.madeeasy.de/7/prgtxtpplus.htm>

Il codice del programma ha questo aspetto:

```
Pulsante PUBLIC SUB1_Click ()  
TextArea1.Text = "1a parte"  
TextArea2.Text = "2a parte"  
FINE  
  
PUBLIC SUB Button2_Click ()  
TextArea3.Text = "1a parte" & "e 2a parte"  
FINE
```

Anche il codice seguente è corretto:

```
TextArea3.Text = TextArea1.Text e TextArea2.Text
```

TextArea3.Text = TextArea1.Text + TextArea2.Text restituisce un messaggio True o T.

6.33 Nuova riga con & Chr \$ (10) e Chr \$ (13)

I caratteri Chr \$ (10), Chr \$ (13) hanno un significato speciale. Non simboleggiano una lettera come Chr \$ (60) ma cambiano invece la linea. I vecchi nomi erano LF e CR, cioè avanzamento riga e ritorno a capo. Vengono ancora dalla zona delle macchine da scrivere.

Si noti che in Gambas il segno più + può essere utilizzato solo come simbolo matematico. Per incollare le corde usi &.

Il programma di esempio incolla insieme 2 parti di testo e inserisce una nuova riga tra di loro. Il risultato viene emesso in una terza TextArea.

Ottieni tre TextAreas dalla casella degli strumenti. Poi altri due pulsanti. Rinomina i pulsanti nelle Proprietà (= richiama le proprietà con il tasto F4).

<http://www.madeeasy.de/7/prglinefeed.htm>

Il codice del programma ha questo aspetto:

```
Pulsante PUBLIC SUB1_Click ()
TextArea1.Text = "1a parte"
TextArea2.Text = "2a parte"
FINE
```

```
PUBLIC SUB Button2_Click ()
TextArea3.Text = "1a parte" & Chr $ (10) & Chr $ (13) & "e 2a parte"
FINE
```

6.34 Etichetta di testo in grassetto, corsivo e colorato

L'output di testo in un'etichetta di testo può essere reso corsivo, grassetto o colorato. Il seguente programma mostra come funziona.

Ottieni un'etichetta di testo dalla casella degli strumenti. Poi un altro pulsante. Rinomina il pulsante in Proprietà (= richiama proprietà con il tasto F4).

<http://www.madeeasy.de/7/prgtextrlabel.htm>



Fig.70 Screenshot del programma di etichette di testo

Fare clic sul pulsante in modalità bozza e immettere il codice del programma dal basso. Sfortunatamente, il testo di origine viene visualizzato in modo errato sotto l'input Wiki. Pertanto, in questo paragrafo wiki, fai clic su Modifica in alto a destra e prendi il controllo del testo sorgente wiki del programma direttamente dalla versione di modifica. Quindi funziona correttamente.

```
Pulsante PUBLIC SUB1_Click ()  
"TextLabel1.Text = " <b> TextLabel </b> <br> "&  
<i> Qui puoi influenzare l'output di testo </i> <br>" & <u> Ecco come funziona  
</u> <br>" &  
<font color = red> anche in rosso! </font>"  
FINE
```

6.35 Il comando split

Il comando Dividi divide una stringa di testo in più parti. L'output di testo è separato da un carattere separatore. Il carattere separatore è preimpostato su una virgola. Se vuoi un separatore diverso devi specificarlo. Nell'esempio fornito, uno spazio viene utilizzato come carattere separatore.

6.35.1 Suddividere una frase nelle sue parole

Con il comando split puoi facilmente dividere una frase nelle sue parole:

Programma di esempio in modalità terminale

```
STATICO PUBBLICO SUB Principale ()
Elenco DIM come stringa []
Elemento DIM AS String
DIM separata come stringa
DIM text1 AS stringa
separato = ""
text1 = "Questa è una frase composta da più parole."
list = Split (text1, separating)
PER OGNI elemento IN lista
    Elemento PRINT
    IL PROSSIMO
FINE
```

Output nella finestra diretta:

```
Questo
è
un
frase
Con
parecchi
Parole.
```

6.35.2 Esempio 2: somma di una serie di numeri

Vedi il programma di aritmetica e sommatoria di Gambas Addy

6.35.3 Esempio 3 del comando split

Il programma di esempio mostra come utilizzare il comando split.

L'esempio qui fornito è un po 'complicato perché nell'output l'ordine delle parole viene scambiato da dietro a davanti.

Ottieni una nuova forma. Posiziona un campo di testo (= casella di testo) su di esso.
(Prendi l'elemento dalla casella degli strumenti con F6 e fai doppio clic.)

Posiziona un campo di comando su di esso e rinominalo come mostrato nell'immagine:

Ottieni due campi di etichette di testo dalla casella degli strumenti.

Codice del programma:

```
STATICO PUBBLICO SUB Principale ()
hForm AS F
hForm = NUOVO F
hForm.show
FINE
Pulsante PUBLIC SUB1_Click ()
myAr AS string []
myStr AS stringa
outstr AS stringa
```

La stella marina

```
x AS intero
outstr = ""
myAr = Split (TextBox1.Text, "")
FOR x = 1 TO myAr.length
    outstr = outstr & "" & myAr [myAr.length - x]
    IL PROSSIMO
    TextLabel2.Text = outstr
    FINE
```

Sintassi divisa

```
Array = Split (String [, Separators, Escape])
```

Divide una stringa in sottostringhe delimitate da separatori. È possibile specificare caratteri di escape: eventuali caratteri separatori racchiusi tra due caratteri di escape vengono ignorati nel processo di divisione.

Nota che Split richiede solo tre argomenti: se vuoi usare più separatori, dovresti passarli come secondo parametro, concatenati in una singola stringa.

Per impostazione predefinita, il carattere virgola è il separatore e non sono presenti caratteri di escape.

Questa funzione restituisce una matrice di stringhe riempita con ciascuna sottostringa rilevata.

esempio

```
DIM Elt AS String []
DIM Sb AS String
Elt = Split ("Gambas Almost Means BASIC! 'Accetto?", "", "")  
PER CIASCUNO Sb IN Elt
    STAMPA Sb
    IL PROSSIMO
```

Produzione:

```
Gamberi
Quasi
Si intende
DI BASE
!
essere d'accordo?
```

6.36 Il comando Sostituisci - Sostituisci caratteri

Con il comando Sostituisci \$ puoi sostituire lettere o sillabe in un testo.

Il comando ha la seguente sintassi:

```
Risultato = Sostituisci $ (testo, modello di ricerca, modello di sostituzione)
```


Il comando sostituisce il modello di ricerca della sillaba con il nuovo modello sostitutivo in ogni punto del testo. L'intera cosa viene salvata nel risultato. Nota il risultato, il testo, il modello di ricerca, il modello sostitutivo sono nomi scelti arbitrariamente Se il testo è un testo vuoto, anche il risultato è zero. Se il modello di ricerca è vuoto (zero), allora risultato = testo.

6.36.1 Piccoli esempi dalla guida di Gambas

```
PRINT Sostituisci $ ("Gambas is basic", "bas", "BAS")
```

>> GamBAS è BASic

```
PRINT Sostituisci $ ("Gambas is basic", "a", "")
```

>> Gmbs è bsic

```
PRINT Sostituisci $ ("Gambas is basic", "", "-")
```

>> Gambas - è - di base

6.36.2 Programma di esempio

Nell'esempio seguente puoi osservare l'azione di scambio e modificarla come desideri.

6.36.3 Layout

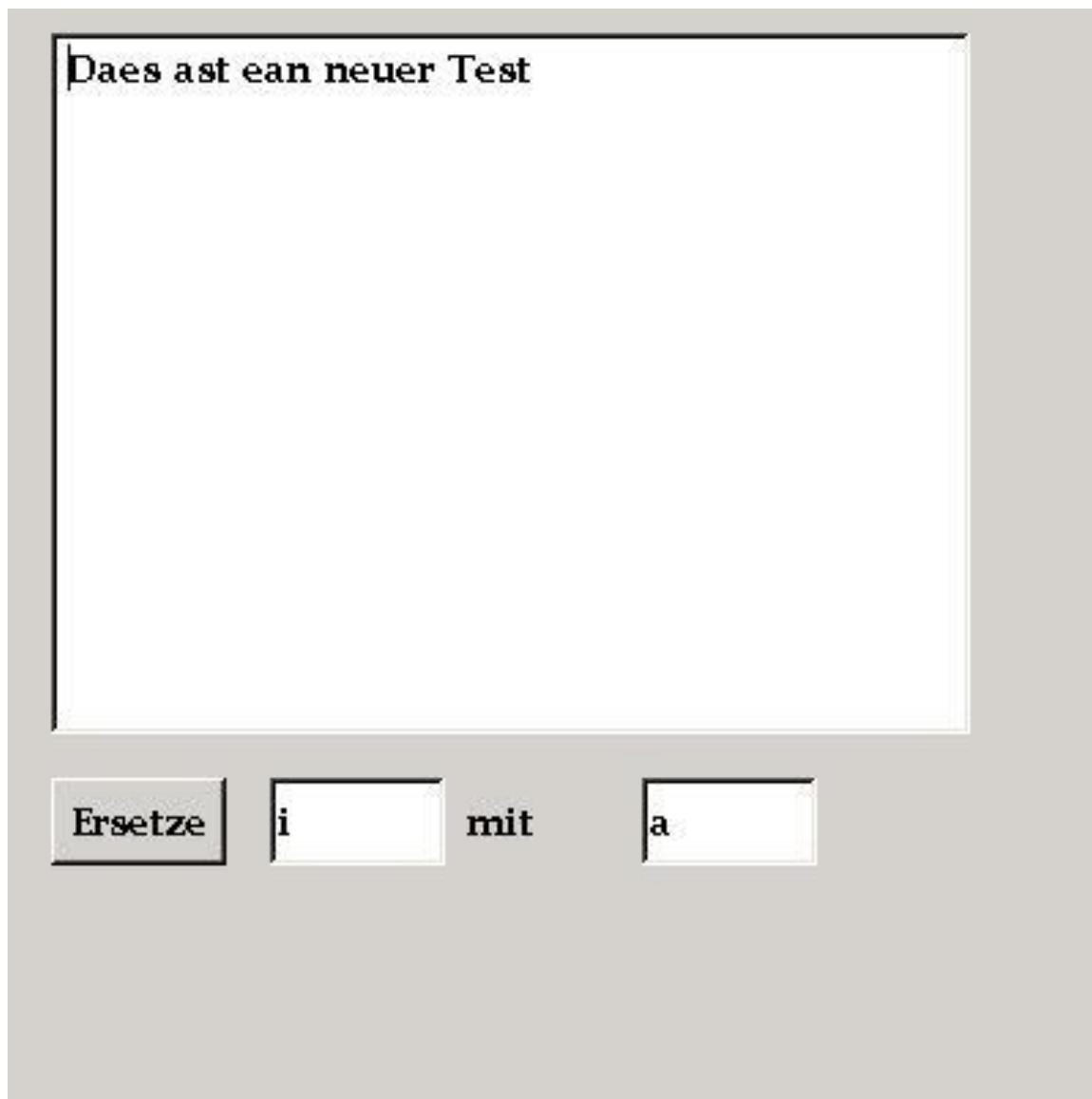


Fig.71

Hai bisogno:

- 1 area di testo
- 1 pulsante di comando
- 2 caselle di testo
- 1 etichetta

6.36.4 Il codice

```
PUBLIC SUB Form_Open ()  
    ME.Text = "Sostituisci comando"  
    Textarea1.Text = "Questo è un nuovo test"  
    Textbox1.Text = "i"  
    Textbox2.Text = "a"  
    Label1.Text = "con"  
    Button1.Text = "Sostituisci"  
FINE  
Pulsante PUBLIC SUB1_Click ()  
    una stringa AS  
    a = Sostituisci $(Textarea1.Text, Textbox1.text, Textbox2.text)  
    textarea1.Text = a  
FINE
```

6.37 Instr Qual è la posizione della parte?

Il comando InStr cerca una stringa breve in una stringa più lunga.

```
print Instr ("modificato", "ea")
```

Risultati nell'output:

```
2
```

La sequenza di testo ea si trova dalla posizione 2 in processato: processato.

Il breve programma

```
STATICO PUBBLICO SUB Principale ()  
PRINT Instr ("Questo testo verrà modificato in un attimo", "ei")  
PRINT Instr ("Questo testo verrà elaborato in un attimo", "ei", 28)  
PRINT Instr ("Questa stringa verrà ora elaborata immediatamente", "on")  
FINE
```

si traduce nell'output diretto:

```
26th  
36  
0
```

- 26 è il primo riferimento.
- 36 è il riferimento dopo la posizione 28
 - Se si immette un numero intero N dopo una virgola dopo la sequenza di testo da cercare, il comando inizierà a cercare solo dall'ennesima posizione nel testo.
- Non c'è nulla di simile nel testo, quindi viene emesso 0.

6.38 RInStr

Il comando RInStr funziona come il suo omonimo InStr, solo che cerca dall'estremità destra. (da destra quindi RInStr)

Esempio con tutte le possibilità di RInStr

```
STATICO PUBBLICO SUB Principale ()  
PRINT RInstr ("Questo testo verrà ora modificato", "ei")  
PRINT RInstr ("Questo testo verrà ora modificato", "ei", 28)  
PRINT RInstr ("Questo testo verrà elaborato a breve", "a")  
FINE
```

Lo stesso esempio in modalità grafica:

```
Pulsante PUBLIC SUB1_Click ()  
PRINT RInstr ("Questo testo verrà ora modificato", "ei")  
PRINT RInstr ("Questo testo verrà ora modificato", "ei", 28)  
PRINT RInstr ("Questo testo verrà elaborato a breve", "a")  
FINE
```

Per ottenere l'esempio è necessario un pulsante di comando nel modulo.

Produzione:

```
36  
26th  
0
```

6.39 Subst segnaposto nel testo

Subst può essere utilizzato per inserire un segnaposto variabile in un testo. La parte da sostituire inizia sempre con & seguita da un numero. (& 1, & 2, & 3 ecc.). L'inserimento avviene uno dopo l'altro.

```
print Subst ("Modifica questo testo in un & 1", "Casella di testo")
```

& 1 verrà sostituito dalla casella di testo.

```
print Subst ("Sostituisci & 1 testi in & 2.", "Tutto", "Casella di testo")
```

Produzione:

```
Sostituisci tutti i testi nella casella di testo.
```

6.40 Determina la lunghezza di una stringa (una sequenza di testo) Len

Il comando Len può essere utilizzato per misurare la lunghezza di una stringa.

```
print Len ("La lunghezza sarà misurata allo stesso modo.")
```

Produzione:

```
32
```

Esempio:

```
PUBLIC SUB Form_Open ()  
    Button1.Text = "Lunghezza"  
    Textbox1.Text = "Questo è un test"  
    Textbox2.Text = ""  
FINE  
Pulsante PUBLIC SUB1_Click ()  
    textbox2.Text = Str (Len (textbox1.text))  
FINE
```

Hai bisogno di un pulsante di comando e 2 caselle di testo nel modulo per ottenere l'esempio. Se non sai cosa significa il comando str, dai un'occhiata qui: Gambas: Conversione di variabili²Str converte ogni altra variabile in una stringa. Prova l'esempio senza Str. Funziona anche questo.

6.41 CHR () Converte il codice numerico ASCII in caratteri

Il comando CHR converte i numeri in lettere e caratteri

Sintassi:

```
Carattere = Chr $ (numero)
```

Restituisce il carattere che corrisponde al codice ASCII del numero. Puoi fare a meno del carattere stringa precedentemente comune \$.

```
Chr (numero) è uguale a Chr $ (numero)
```

L'opposto del comando CHR è il comando ASC.

Esempio 1:

```
STAMPA Chr $ (65)
```

Edizione: A

Programma di esempio per il comando CHR in modalità terminale

```
STATICO PUBBLICO SUB Principale ()
DIM x AS intero
PER x = da 30 a 133
STAMPA x, Chr $ (x)
IL PROSSIMO
STAMPA
PRINT "Interruzione riga CHR (10)", Chr $ (10)
PRINT "Return CHR (13)", Chr $ (13)
PRINT "spazio Chr $ (32)", "a"; Chr $ (32); "a"
STAMPA "0 CHR48", Chr $ (48), Str (0)
STAMPA "9 CHR57", Chr $ (57), Str (9)
PRINT "@ CHR64", Chr $ (64), Str ("@")
STAMPA "A CHR65", Chr $ (65), Str ("A")
STAMPA "Z CHR90", Chr $ (90), Str ("Z")
PRINT "a CHR97", Chr $ (97), Str ("a")
PRINT "z CHR122", Chr $ (122), Str ("z")
FINE
```

6.42 ASC () Converte i caratteri alfanumerici in codici numerici ASCII

Il comando Asc restituisce il codice ASCII di una lettera o di un carattere. Se la posizione non è specificata, viene restituito il codice ASCII del primo carattere.

sintassi

```
Asc (String [, Position])
```

Esempio:

```
PRINT Asc ("Gambas")
```

Edizione: 71

```
PRINT Asc ("Gambas", 3)
```

Edizione: 109

Esempio in modalità terminale:

```
STATICO PUBBLICO SUB Principale ()
DIM test AS String
```

```

DIM x AS intero
test = "1234567 Questo è un test"
FOR x = 1 TO Len (test)
    PRINT Asc (test, x)
IL PROSSIMO
FINE

```

6.43 Str () converte altre variabili in una stringa di testo

Se hai un numero o un'altra variabile che vuoi convertire in testo, allora dovresti usare il comando Str (). Gambas esegue semplici conversioni automaticamente senza il comando Str, ma non fare affidamento su questi automatismi. In caso di dubbio, aggiungi una Str.

Esempio: hai bisogno di una casella di testo e di un pulsante di comando per farlo funzionare.

```

PUBLIC SUB Form_Open ()
    ME.Text = "Str converte altre variabili in testo"
    Button1.Text = "Avvia"
    textbox1.Text = ""
FINE
Pulsante PUBLIC SUB1_Click ()
    textbox1.Text = Str (8 * 37)
    'textbox1.Text = 8 * 37
    'Provalo una volta senza str
FINE

```

Vedi anche Gambas: Conversione di variabili # str \$³

6.44 set di caratteri

Il seguente programmino restituisce il set di caratteri corrente del sistema. Tutto ciò di cui hai bisogno è una forma vuota.

```

PUBLIC SUB Form_Open ()
PRINT System.Charset
FINE

```

Emissione ad es. Nella finestra diretta:

```
ISO-8859-15
```

Vedi anche CONV \$

³ Capitolo 6.145 a pagina 354

6.45 Selezione del carattere tramite una finestra di dialogo standard

Nel seguente programma di esempio, il carattere di un'area di testo può essere selezionato tramite una finestra di dialogo standard.

Hai bisogno di un'area di testo e un pulsante di comando per questo.

```
Pulsante PUBLIC SUB1_Click ()  
IF Dialog.SelectFont () THEN RETURN  
Textarea1.Font = Dialog.Font  
FINE
```

Ovviamente puoi anche definire il carattere tramite le proprietà dell'area di testo o codificarlo direttamente nel programma.

Un programma di esempio mostra la codifica diretta durante il runtime. Hai bisogno di un modulo e un'area di testo per avviare il programma.

```
PUBLIC SUB Form_Open ()  
Form1.Font.Name = "Utopia"  
Form1.Font.Bold = TRUE  
Form1.Font.Italic = TRUE  
Form1.Font.Size = "24"  
Form1.Font.StrikeOut = FALSE  
Form1.Font.Underline = TRUE  
FINE
```

6.46 Elenca tutti i caratteri disponibili

È possibile utilizzare il seguente programma di esempio per elencare tutti i caratteri disponibili. È necessario un pulsante di comando per avviare il programma. I caratteri vengono visualizzati nella finestra diretta.

```
Pulsante PUBLIC SUB1_Click ()  
DIM c AS stringa  
PER OGNI font c IN  
STAMPA c  
IL PROSSIMO  
FINE
```

6.47 Testo negli appunti

Gli appunti sono una delle invenzioni più pratiche da quando esisteva il PC. Possono essere utilizzati anche sotto i gamberi.

Il seguente programmino copia un testo negli appunti con il comando 1 e lo recupera con il comando 2.

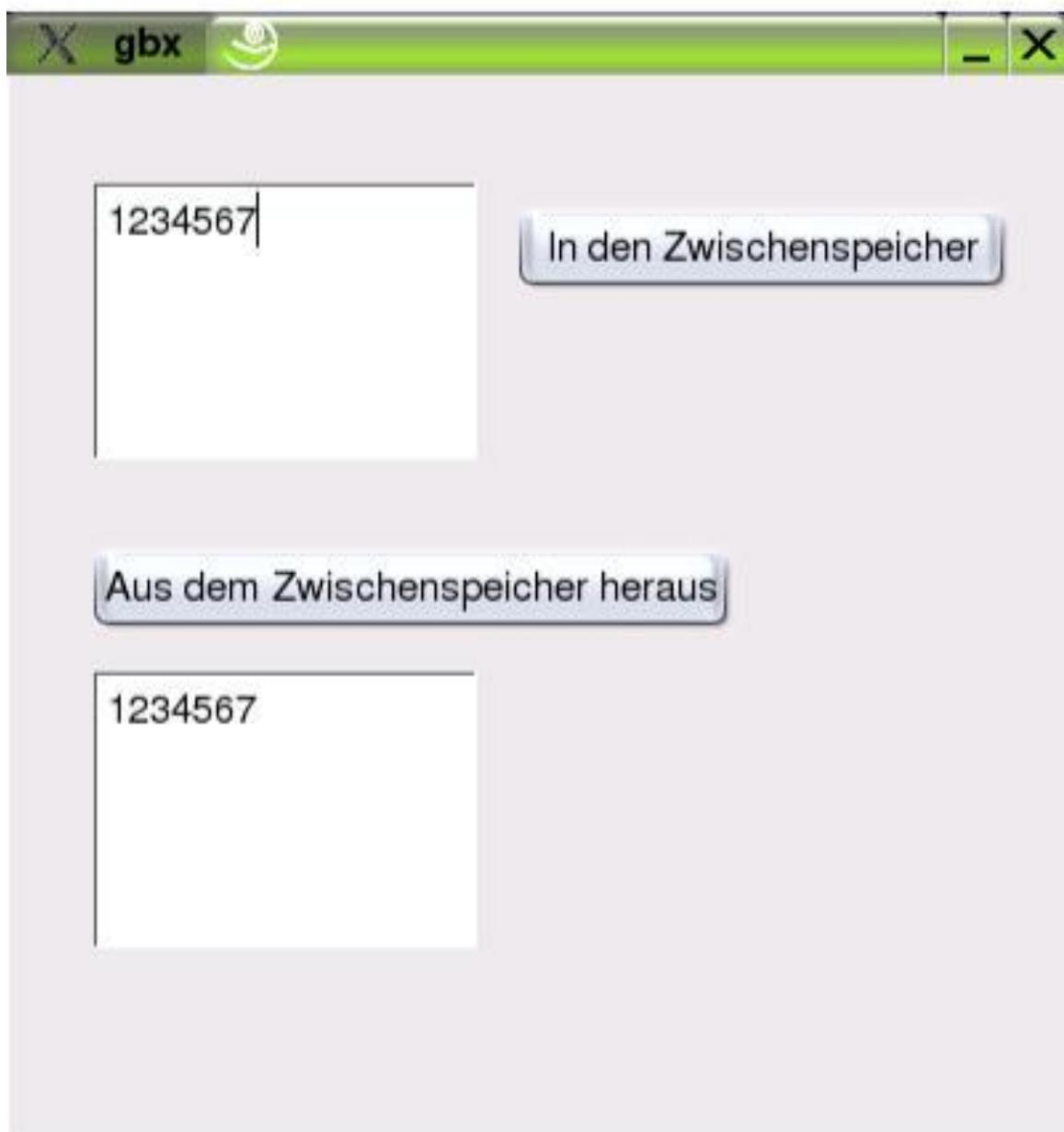


Fig.72

Il seguente codice di programma si trova dietro i campi di comando dentro e fuori dal buffer:

```
Pulsante PUBLIC SUB1_Click ()  
DIM t AS stringa  
Appunti. Cancella  
' Elimina appunti. t =  
textarea1.Text Clipboard.Copy (t)  
  
'Copia il testo negli appunti. textarea1.Text =  
""  
FINE
```

```
PUBLIC SUB Button2_Click ()  
DIM una stringa $ AS  
'Ottieni testo dagli appunti.  
textarea2.Text = Clipboard.Paste ()  
FINE
```

Versione breve:

```
"File di classe Gambas  
Pulsante PUBLIC SUB1_Click ()  
Appunti.Copia (Textarea1.Text)  
'Copia il testo negli appunti.  
FINE  
  
PUBLIC SUB Button2_Click ()  
textarea2.Text = Clipboard.Paste ()  
«Tiralo fuori di nuovo  
FINE
```

6.48 Teoria degli appunti

La classe degli appunti è statica. Che cosa significa ??

Gli appunti hanno le seguenti proprietà:

- formato
- genere

Ha i seguenti metodi

- Chiaro
- copia
- incolla

Ha le seguenti costanti

- Immagine
- Nessuna
- testo

Non puoi istanziare questa classe. Riferimento a: Componente dell'interfaccia utente grafica - gb.qt

6.49 Come puoi copiare i grafici negli appunti e portarli di nuovo fuori?

In Linux puoi copiare uno screenshot negli appunti con Ctrl e il pulsante. Come funziona sotto i gamberi?

6.50 Come posso modificare gli appunti di Klipper (KDE)?

Lo strumento appunti Klipper è disponibile in KDE. Come funziona la manipolazione delle cesoie sotto i gamberi?

6.51 Panoramica dei comandi

Data	Restituisce la data di sistema corrente
formato	abilita uscite differenti per ora e data
IsDate	Restituisce un valore booleano che indica se un'espressione è in un file La data può essere convertita
minuto	Restituisce un numero intero compreso tra 0 e 59 che rappresenta il minuto in rappresenta l'ora
Mese	Restituisce un numero intero compreso tra 0 e 12 che rappresenta il mese in L'anno rappresenta
Adesso	Restituisce la data e l'ora correnti sul computer
Tempo, tempo \$	Restituisce una data di tipo che indica l'ora di sistema corrente
Timer	Restituisce il numero di secondi trascorsi dall'inizio del programma. i geni sono
Giorno feriale	Restituisce il giorno della settimana come numero intero (1 = domenica, 2 = lunedì, eccetera.)
Anno	Restituisce l'anno come numero intero
Aspettare	Attende per un periodo di tempo specificato

6.52 Comando di attesa (pausa)

La sintassi del comando wait è la seguente.

ATTENDI [pausa]

Chiama il ciclo degli eventi. Se è stato immesso un valore per la pausa, rimane nel ciclo di attesa fino allo scadere del tempo di pausa. Pausa è un numero in virgola mobile. Se vuoi programmare un ciclo di attesa di 100 ms, devi farlo

ATTENDERE 0.1

accedere.

Gli input da tastiera o mouse non sono possibili durante il ciclo di attesa. Solo gli eventi di disegno, timer e scrittura di file come Process_Write stanno ancora scadendo.

Esempio:

L'esempio seguente mostra l'uso del comando wait. Hai bisogno di un modulo con un pulsante di comando e una casella di testo.


```
Pulsante PUBLIC SUB1_Click ()  
    ATTENDI 5  
    'attende 5 secondi  
    ATTENDERE 0.1  
    'attende 100 millisecondi  
    textbox1.Text = ""  
FINE
```

6.53 tempo

Comandi importanti per il tempo sono il tempo e adesso. Il timer è un cronometro che percorre un certo intervallo di tempo e quindi avvia un evento. Attenzione: la funzione timer è diversa dal timer. La funzione timer misura il tempo in secondi trascorso dall'inizio del programma.

Le conversioni semplici dovrebbero essere in grado di convertire i secondi in minuti, ore e giorni.
Anche il tutto al contrario.

6.53.1 Comando del tempo

Un buon esempio di una semplice funzione è il comando Time. Questa funzione fornisce l'ora di sistema corrente. Puoi testare questa funzione con il seguente comando:

```
Tempo di stampa
```

Il seguente piccolo programma senza grafica mostra l'ora nella finestra diretta:

```
STATICO PUBBLICO SUB Principale ()  
    Tempo di STAMPA  
FINE
```

Se vuoi che l'ora venga aggiornata ancora e ancora, devi costruire in un ciclo. Il programma quindi si presenta così:

```
STATICO PUBBLICO SUB Principale ()  
    DIM i AS intero  
    PER i = da 1 a 100  
        Tempo di STAMPA  
        ATTENDERE 0.1  
        IL PROSSIMO  
    FINE
```

Il programma emette il tempo ogni 100 millisecondi per 10 secondi.

Un esempio simile per l'output grafico aggiorna continuamente l'output dell'ora. Per fare ciò, usi il timer, l'elemento di controllo del timer. Questo inizialmente non ha nulla a che fare con l'ora corrente, ma attiva un evento dopo un certo tempo.

Il seguente piccolo programma di esempio combina il timer e la funzione dell'ora in un'informazione sull'ora costantemente aggiornata, come puoi trovarla sul tuo PC in basso a destra.

Inizia un nuovo progetto. Chiamalo tempo trascorso. Quindi posiziona una casella di testo sulla tua forma. È inoltre necessario il controllo del timer con l'icona dell'orologio. Ottieni entrambi con F6 e un doppio clic sul simbolo dalla casella degli strumenti. Assicurati che il timer con la proprietà Enabled debba essere prima attivato. Puoi impostarlo nelle proprietà del timer in fase di progettazione o puoi farlo nel codice del programma con la riga

```
Timer1.Enabled = TRUE
```

raggiungere.

```
PUBLIC SUB Timer1_Timer ()  
    Timer1.Enabled = TRUE  
    TextBox1.Text = Ora  
    FINE
```

Il programma mostra l'ora corrente nella casella di testo. Viene aggiornato ogni secondo (= 1000 millisecondi).

6.53.2 Funzione timer

La funzione timer non ha nulla a che fare con il controllo timer con lo stesso nome.

La funzione timer emette il tempo in secondi trascorsi dall'inizio del programma.

Questo è diverso in Visual Basic. Qui la funzione timer restituisce il numero di secondi trascorsi dalla mezzanotte con una precisione di 1/100 di secondo.

Il codice seguente è dietro il comando Tempo dall'avvio del programma:

```
Pulsante PUBLIC SUB1_Click ()  
    TextBox1.Text = Timer  
    FINE
```

Esempio:

```
Timer STAMPA
```

Variabile di risultato

```
0.291657006775
```

La funzione può essere utilizzata anche per determinare il tempo di esecuzione di un programma o di una parte di un programma.

Esempio:

```
Avvio = timer  
'Esegui il ciclo  
Fine = timer  
Tempo = fine - inizio  
'Il tempo è misurato in secondi
```

È possibile utilizzare la funzione timer insieme all'istruzione randomize per generare un valore iniziale per la funzione Rnd.

6.53.3 Comando di formattazione per l'output di data e ora

Il comando Format \$ è molto ampio nelle sue possibilità. Di seguito solo l'ora e La funzione data può essere considerata. In generale funziona così:

```
Stringa = Formato $ (Espressione [, Formato])
```

Il comando Formato converte un'espressione in una stringa di testo, a seconda di come lo desideri tu stesso.

La funzione utilizza le impostazioni specifiche del paese del tuo computer e deve quindi essere verificata più volte, soprattutto quando si passa da una lingua all'altra.

I formati di data definiti dall'utente possono essere emessi come segue:

yy	restituisce l'anno in 2 cifre
yyyy	restituisce l'anno in 4 cifre
m	uscite il mese
mm	restituisce il mese in 2 cifre
mmm	restituisce il mese sotto forma di testo abbreviato, ad esempio agosto = agosto
mmmm	restituisce il mese per intero
d	uscite la giornata
dd	emette il giorno in 2 cifre
ddd	emette il giorno della settimana in forma abbreviata
dddd	emette il giorno della settimana per intero
/	restituisce il separatore della data
H	emette l'ora
hh	emette l'ora in 2 cifre
n	emette i minuti
nn	emette i minuti in 2 cifre
S	emette i secondi
ss	emette i secondi in 2 cifre
:	emette il separatore orario

Oltre al comando di formattazione, il programma di esempio riepilogato in modalità terminale utilizza anche la funzione Now per l'istante temporale corrente. Adesso = adesso.

```
STATICO PUBBLICO SUB Principale ()
Formato STAMPA $ (ora, "mm / gg / aaaa hh: mm: ss")
Formato STAMPA $ (ora, "g / m / aa h: m: s")
Formato STAMPA $ (ora, "gg gg mmm aaaa")
Formato STAMPA $ (ora, "ggg gg mmmm aaaa")
FINE
```

L'output nella finestra diretta con l'impostazione del paese tedesco è simile a questo:

```
12/11/2004 8:11:10 PM
12/11/04      20:11:10
Ven 12 novembre 2004
Venerdì 12 novembre 2004
```

Esempi personalizzati:

comand o	produzione
Formato STAMPA \$ (ora, "mm / gg / aaaa hh: mm: ss")	15/04/2002 09:05:36
Formato STAMPA \$ (ora, "g / m / aa h: m: s")	15/04/02 9: 5:36
Formato STAMPA \$ (ora, "gg gg mmm aaaa")	Lunedì 15 aprile 2002
Formato STAMPA \$ (ora, "ggg gg mmmm aaaa")	Lunedì 15 aprile 2002

6.53.4 Problema con l'indicazione dello zero all'indietro

Vedere la nota di InselAX⁴.

Messaggio di testo originale:

Ho anche risolto un altro problema. Ha a che fare con il tipo di dati "DATE". Non tutte le date sono uguali! Il problema è sorto quando ho inserito l'ora corrente, che viene trasferita nel mio modulo di input e continua a funzionare lì con l'aiuto di un timer, (utilizzando un file di configurazione) su GMT (Greenwich Mean Time; qui in Germania -1 ora per l'inverno tempo; -2 ore per l'ora legale) voleva cambiare. Il mio timer ha iniziato a funzionare all'indietro quando è passato dalle 23:59 alle 00:00 ora locale ... per qualsiasi motivo ...!?!?.

La soluzione è scrivere qualcosa del genere:

```
DIM gmt AS Date
DIM ora locale AS Date
Correzione DIM AS Date

ora locale = ora (ora)
gmt = correzione dell'ora locale
```

(dove la correzione è Val ("01:00:00") e 01:00:00 proviene da un file)

Se dovessi scrivere ora, ad esempio:

```
TextBoxZeit.Text = gmt
```

quindi il timer andrebbe all'indietro quando si cambia e conta fino a 00:00, a partire da 01:00:00.

Il problem solving è prima di tutto:

```
time = gmt + Val ("00:00:00")
```

poi

```
TextBoxZeit.Text = gmt
```

6.53.5 Problema con l'ora di Greenwich

Vedere la nota di InselAX⁵.

Il Greenwich Mean Time (qui in Germania ora locale - 1 ora) scorre in una casella di testo GMT, che ha funzionato anche per 23 ore. Per 1 ora dalle 00:00 all'01: 00 l'ora corretta è stata visualizzata in GMT (dalle 23:00 alle 00:00), ma la lettura della casella di testo ha portato solo "00/00/00". Ha descritto il problema a Benoit e mi ha chiesto di inviargli il mio software. Ieri mi ha mandato un cerotto ... lo allego anche a te. C'era un bug nella conversione di

Fluttua per data. Con la patch funziona. Devi copiarlo in "./src/exec" (dove si trova il tuo gambas ...) e poi ricompilare i gambas.

Vedi: gbx_value.c⁶

6.53.6 Libro Gambas> Ora e data> "Problema con ore zero, ora all'indietro" e "Problema con l'ora di Greenwich"

Il problema è nel modo in cui la data viene utilizzata per memorizzare la data e l'ora.

La data è un clone del tipo di dati Double con un intervallo di valori limitato da -657434 a 2958465.99998843

La parte intera del numero viene interpretata come una data e abilita una data da "01.01.100" a "31.12.9999". La parte frazionaria viene interpretata come tempo utilizzando la seguente formula:

5 Capitolo 6.51 a pagina 253

[http://de.wikibooks.org/wiki/Gambas%3A%20Zeit%20und%20Datum%3A%20gbx%](http://de.wikibooks.org/wiki/Gambas%3A%20Zeit%20und%20Datum%3A%20gbx%20value.c)

6 ° 20value.c

```
(Secondi dalla mezzanotte) / (24*60*60)
```

Quindi:

```
0 <= tempo <1
```

Poiché l'ora non può mai essere negativa, un segno negativo viene ignorato.

Se ora guardi il tuo programma nel libro Gambas, la parte pre-decimale viene cancellata nella determinazione dell'ora locale dalla funzione ora. dopo le 0, gmt riceve un valore negativo (-1 ora) ignorando il segno, diventa l'1. Nell'ora successiva (fino a 0:59:59), gmt viene fornito con valori da -1 ora a -1 secondo, che vengono convertiti in "1:00:00" - "0:00:01". Quindi il tempo sembra solo scorrere all'indietro.

```
DIM gmt AS Date
DIM ora locale AS Date
Correzione DIM AS Date

ora locale = ora (ora)
gmt = ora locale - correzione "correzione =" 1:00:00 "
```

Quindi ecco un bug nel programma! Devi assicurarti che gmt non diventi mai negativo. La colpa è nella posizione della funzione tempo. Confronta il seguente programma:

```
DIM gmt AS Date
DIM oggi AS Date
Correzione DIM AS Date

oggi = Adesso
gmt = Time (oggi - correzione)
```

Qui la parte prima della virgola (data) viene cancellata solo dopo aver dedotto la correzione. Di conseguenza, gmt non diventa negativo e il calcolo del tempo funziona correttamente. Il programma può essere ulteriormente semplificato:

```
DIM gmt AS Date
Correzione DIM AS Date

gmt = Time (Now - correzione)
```

A proposito, puoi interrogare la data corretta allo stesso modo:

```
data = Int (Adesso - correzione)
```

Quindi non ci sono bug, è solo una questione di corretta programmazione! (e l'esatta conoscenza del tipo di dati Date) - InselAX⁷ ° 16:28, 15 giugno 2008 (CEST)

6.54 data

I comandi importanti per la data sono data, giorno della settimana, ora, formato e formato \$

Le seguenti domande sono interessanti:

- Quale giorno della settimana ha una data specifica? Vedi funzione giorno della settimana
- Quanti giorni ci sono tra due date diverse?

6.54.1 Oggi

La data e l'ora odiene possono essere ottenute con il comando Now.

Esempio:

```
STAMPA ora
```

Produzione:

```
16/05/2002 22:31:30  
«È passato così tanto tempo da quando Benoit lo ha programmato
```

La seguente applet restituisce la data odierna in una casella di testo:

Vedi: prgheute.png⁸ °

Il codice seguente è dietro il comando Today:

```
Pulsante PUBLIC SUB1_Click ()  
    TextBox1.Text = "Oggi è il" & Str $ (Adesso)  
FINE
```

Nota per i programmi VB6: In VB6 Str \$ deve essere sostituito da CStr - InselAX⁹ 17:58,
17.Giugno 2008

(CEST) Esempio 2:

Il comando Date riduce l'output di data e ora solo alla data. Per fare ciò, prova il seguente programma. Hai solo bisogno di una forma vuota per aviarlo.

7 <http://de.wikibooks.org/wiki/Benutzer%3AInselAX>

8 <http://www.madeeasy.de/7/prgheute.png>

9 <http://de.wikibooks.org/wiki/Benutzer%3AInselAX>

```
PUBLIC SUB Form_Open ()
    STAMPA ora
    STAMPA data (ora)
FINE
```

Poiché la data viene salvata nel tipo di dati "Date" come parte intera, Date (Now) può anche essere sostituita da Int (Now) - InselAX¹⁰ 17:59, 17 giugno 2008 (CEST)

Nota per i programmi VB6: in VB6, Date deve essere sostituito da DateValue - InselAX¹¹ 17:58, 17 giugno 2008 (CEST)

6.54.2 Giorno della settimana

Programma di esempio:

Emette il giorno della settimana di oggi. Il programma funziona in modalità terminale:

```
STATICO PUBBLICO SUB Principale ()
    PRINT WeekDay (ora)
    STAMPA "Oggi è";
    IF WeekDay (Now) = 1 THEN PRINT "Monday"
    IF WeekDay (Now) = 2 THEN PRINT "Tuesday"
    IF WeekDay (Now) = 3 THEN PRINT "Wednesday"
    IF WeekDay (Now) = 4 THEN PRINT "Thursday"
    IF WeekDay (Now) = 5 THEN PRINT "Friday"
    SE WeekDay (adesso) = 6 ALLORA STAMPA "Sabato"
    IF WeekDay (Now) = 7 THEN PRINT "Sunday"
FINE
```

Il programma utilizza il comando Weekday e il comando Now di Gambas.

Il programma presuppone che il lunedì sia il primo giorno della settimana. In VB6, Weekday ha un altro parametro (opzionale) che specifica il primo giorno della settimana (1 = Sun 2 = Lun 3 = Tue 4 = Wed 5 = Thu 6 = Fri 7 = Sat), con domenica come impostazione predefinita. Affinché il programma funzioni correttamente in VB6, ogni "WeekDay (Now)" deve essere sostituito da WeekDay (Now, 2).

In alternativa, il giorno della settimana può anche essere visualizzato con l'istruzione di formato. Formato (ora, "ddd") per il giorno della settimana ridotto a 2 lettere e formato (ora, "dddd") per il giorno della settimana scritto - InselAX¹² ° 18:23, 17 giugno 2008 (CEST)

6.54.3 Anno bisestile

Il seguente programma di esempio calcola per una voce di anno se l'anno è bisestile. In un anno bisestile c'è un 29 febbraio.

Dietro il comando L'anno è bisestile? è il codice seguente:

¹⁰ <http://de.wikibooks.org/wiki/Benutzer%3AInselAX>

¹¹ <http://de.wikibooks.org/wiki/Benutzer%3AInselAX>

¹² <http://de.wikibooks.org/wiki/Benutzer%3AInselAX>

```
Pulsante PUBLIC SUB1_Click ()  
    DIM leapyear AS String  
  
    'nessun input  
    IF TextBox1.Text = "" THEN  
        TextBox2.Text = "Non è stato fornito nessun anno!"  
        RITORNO  
    FINISCI SE  
  
    'Non inserire alcun numero o numero con un punto (ad es. 1234,5)  
    IF NOT IsInteger (Val (TextBox1.Text)) O TextBox1.Text LIKE "*.*"  
    POI  
        TextBox2.Text = "\" & TextBox1.Text & "\"non è un anno!"  
        RITORNO  
    FINISCI SE  
  
    IF Val (TextBox1.Text) MOD 400 = 0 THEN  
        leapyear = "è un anno bisestile."  
    ELSE IF Val (TextBox1.Text) MOD 100 = 0 THEN  
        leapyear = "è l'inizio del secolo, ma non è un anno bisestile."  
    ELSE IF Val (TextBox1.Text) MOD 4 = 0 THEN  
        leapyear = "è un anno bisestile."  
    ALTRO  
        leapyear = "non è un anno bisestile."  
    FINISCI SE  
  
    TextBox2.Text = "L'anno" & Val (TextBox1.Text) & leapyear END
```

Wikipedia fornisce ulteriori informazioni sul tema dell'anno bisestile¹³.

6.54.4 IsDate

La funzione IsDate restituisce un valore VERO o FALSO, che indica se una variabile del tipo di data può essere interpretata come una data valida. Ciò consente anche un controllo per un anno bisestile, ad esempio.

IsDate ("02/29/2008") = True

IsDate ("02/29/2009") = False

-- InselAX^{14th} 18:46, 17 giugno 2008

(CEST) Sintassi: IsDate (Variant)

6.54.5 CDate

La funzione CDate converte un'espressione in un tipo di dati di data.

Sintassi: CDate (espressione)

13 <http://de.wikipedia.org/wiki/Schaltjahr>

14 <http://de.wikibooks.org/wiki/Benutzer%3AInselAX>

L'argomento dell'espressione deve essere un'espressione stringa o un'espressione numerica che può essere interpretata come una data. I valori consentiti per le informazioni sui dati sono i valori dal 1 gennaio 100 (-657434) al 31 dicembre 9999 (2958465). Il giorno 0 è il 30/12/1899. Questo giorno non può essere raggiunto con CDate perché CDate interpreta 0 come un tempo senza una data - InselAX¹⁵ ° 19:11, 17 giugno 2008 (CEST)

6.54.6 anni

La funzione Year restituisce un numero intero compreso tra 100 e 9999 che rappresenta l'anno in una data immessa come argomento.

Sintassi: anno (data)

Note: l'argomento data è qualsiasi espressione numerica o stringa che può rappresentare una data e / o un'ora dal 1 gennaio 100 al 31 dicembre 9999, dove 1 gennaio 1900 è uguale a 2. I numeri a sinistra del punto decimale in data rappresentano la data, i numeri a destra rappresentano l'ora. I numeri negativi rappresentano date precedenti al 30 dicembre 1899.

Se la data ha il valore zero, anche la funzione year restituisce il valore zero.

6.55 domande

Esiste un comando in Gambas che può essere utilizzato per modificare la data? (In Freebasic c'è il comando SETDATE: SETDATE mese \$ + "-" + giorno \$ + "-" + anno \$)

Esiste un comando in Gambas che può essere utilizzato per modificare l'ora? (In Freebasic c'è il comando SETTIME per questo)

Sotto Linux, la data e l'ora appartengono ai compiti "sovranii", solo gli utenti risp. I programmi con diritti di root possono modificarli. I comandi Gambas corrispondenti sarebbero sotto Linux inutile.

Esiste un comando di misurazione del tempo?

Vedi sopra: funzione timer¹⁶

Per programmare il suono è necessario il file gb.sdl e ovviamente una scheda audio funzionante integrata in Linux. Vedi ad es.<http://www.binara.com/gambas-wiki/> bin / view / Gambas / GbSdl Guarda anche: <http://www.binara.com/gambas-wiki/> bin / view / Gambas / FutureSDLComponent

C'è un ottimo esempio sull'argomento del suono in Gambas: negli esempi forniti con Gambas, un MusicPlayer è incluso nel codice sorgente. per esempio.

¹⁵ <http://de.wikibooks.org/wiki/Benutzer%3AInselAX>

¹⁶ Sezione 6.53.2 a pagina 255

```
"File di classe Gambas

STATICO PRIVATO $ bDoNotMove AS Booleano

PUBLIC SUB btnOpen_Click ()

    IF Dialog.OpenFile () THEN RETURN

        lblTitle.Text = File.Name (Dialog.Path)
        ' Approssimazione ...
        sldPos.MaxValue = 600

        Music.Load (Dialog.Path)
        btnPlay_Click

    FINE

PUBLIC SUB btnPlay_Click ()

    timMusic.Enabled = TRUE
    Music.Play

    FINE

PUBLIC SUB btnPause_Click ()

    Musica, pausa

    FINE

PUBLIC SUB btnStop_Click ()

    Musica. Stop
    timMusic.Enabled = FALSE
    lblPos.Text = ""
    sldPos.Value = 0

    FINE

PUBLIC SUB timMusic_Timer ()

    DIM iPos AS Integer
    DIM iVal AS Integer
    DIM sPos AS String
    DIM iInd AS Integer

    iPos = Music.Pos

    SE NON $ bDoNotMove ALLORA
        Object.Lock (sldPos)
        SE iPos > sldPos.MaxValue ALLORA
            sldPos.MaxValue = sldPos.MaxValue * 2
        FINISCI SE
        sldPos.Value = iPos
        Object.Unlock (sldPos)
    FINISCI SE

    PER iInd = da 0 a 2

        iVal = iPos MOD 60
        iPos = iPos \ 60
        SE IIND ALLORA
            sPos = Formato (iVal, "00") & ":" & sPos
        ALTRO
            sPos = formato (iVal, "00")
        FINISCI SE
```

```

IL PROSSIMO

lblPos.Text = sPos

FINE

PUBLIC SUB sldPos_Change ()

    Music.Pos = sldPos.Value

FINE

PUBLIC SUB sldPos_MouseDown ()

    $ bDoNotMove = TRUE

FINE

PUBLIC SUB sldPos_MouseUp ()

    $ bDoNotMove = FALSE

FINE

PUBLIC SUB sldVolume_Change ()

    Music.Volume = 1 - sldVolume.Value / sldVolume.MaxValue

FINE

```

Con Gambas 2 puoi salvare e modificare immagini da webcam o schede TV.

Il prerequisito è che sia un dispositivo v4l (video per linux).

La classe "VideoDevice" permette il controllo del dispositivo, la memorizzazione di un'immagine in un file File e supporta l'interfaccia di streaming Gambas con i suoi consueti comandi (LEGGI, SEEK, FLUSH, ecc.).

Per favore, riferisci <http://gambas.gnulinex.org/v4l/>

6.56 Salva file

Nel seguente programma di esempio, il testo di un'area di testo viene salvato in un file sul disco rigido. L'intera operazione viene eseguita tramite la finestra di dialogo standard. Hai bisogno di un'area di testo e di un pulsante di comando per avviare il programma.

Esempio:

```

PUBLIC SUB Form_Open ()
    Textarea1.Text = "Questo è un test"
FINE
Pulsante PUBLIC SUB1_Click ()
    Dialog.SaveFile ()
    File.Save (Dialog.Path, TextArea1.Text)
    CATTURARE
    SE ERRORE ALLORA RITORNO
FINE

```

6.57 Salva file senza finestra di dialogo

Puoi anche salvare un file senza una finestra di dialogo standard. In questo caso, tuttavia, devi inserire il percorso e il nome del file nel programma. L'esempio seguente mostra come funziona. Hai bisogno di un'area di testo e di un pulsante di comando per farlo funzionare.

Programma:

```
PUBLIC SUB Form_Open ()  
Textarea1.Text = "Questo è un test"  
FINE  
Pulsante PUBLIC SUB1_Click ()  
File.Save (system.Home & "/test.txt", TextArea1.Text)  
CATTURARE  
SE ERRORE ALLORA RITORNO  
FINE
```

La directory utente corrente viene selezionata con il comando system.home, ad esempio / home / tuonome.

Controlla, preferibilmente con Konqueror o la console, se il file è stato realmente salvato. Prova cosa succede se ometti la routine di errore con Catch.

(Non importa fintanto che il percorso è corretto. Se il percorso non è corretto, viene visualizzato un messaggio di errore: Il file o la directory non esiste.) Provare cosa succede se il file in cui si desidera salvare esiste già. (Il file esistente verrà sovrascritto indipendentemente dalla perdita.) Prova cosa succede se il percorso non è corretto. (Non ci sono messaggi di errore fintanto che la routine di errore con catch è incorporata !!)

Variante di programma: è necessario un solo pulsante di comando.

```
Pulsante PUBLIC SUB1_Click ()  
File.Save (system.Home & "/test.txt", "Questo è il tuo testo")  
CATTURARE  
SE ERRORE ALLORA RITORNO  
FINE
```

6.58 Carica file 1

Il file passwd si trova sulla maggior parte dei sistemi Linux. Vogliamo caricarlo in un'area di testo

Per fare questo devi sapere dove si trova il file: è nella directory: / ecc

Per il programma hai bisogno di:

- un modulo
- un'area di testo

Codice:

```
PUBLIC SUB Form_Open ()  
s come stringa  
s = File.Load ("/ etc / passwd")
```

```
'Il file deve esistere
'altrimenti non può essere caricato.
textarea1.Text = s
FINE
```

Come puoi vedere, il file contiene le password di tutti gli utenti che hanno effettuato l'accesso al tuo sistema Linux.

Questo file è leggibile ma protetto da scrittura.

6.59 Carica file 2

Se hai salvato il file test.txt nella tua home directory nella sezione Salva file, puoi ricaricarlo facilmente:

```
Pulsante PUBLIC SUB1_Click ()
DIM sLine AS String
sLine = File.Load (system.home & "/test.txt")
PRINT sLine
FINE
```

Un altro esempio:

Hai un file file.text nella tua directory e vuoi caricare questo file in Gambas:

L'esempio seguente mostra come funziona:

```
Pulsante PUBLIC SUB1_Click ()
DIM sLine AS String
sLine = File.Load ("File.text")
'Il file deve esistere
'altrimenti non può essere caricato.
PRINT sLine
FINE
```

Hai bisogno di un modulo con un pulsante di comando. L'output avviene nella finestra diretta. Per avviare il programma, devi prima creare il file file.text con un editor. Quindi salva il file nella directory del tuo progetto Gambas. Ad esempio, il file si trova nella directory per me:

```
file: /home/rhoff/gambas-1.0/laden2/
```

6.60 Carica file con finestra di dialogo standard

Con la finestra di dialogo standard è possibile scegliere il percorso e il nome del file durante il caricamento.

Per l'esempio hai bisogno di un modulo, un pulsante di comando, un campo etichetta e un'area di testo.

```
Pulsante PUBLIC SUB1_Click ()  
IF Dialog.OpenFile () THEN RETURN  
Label1.Text = Dialog.Path  
TextArea1.Text = File.Load (Dialog.Path)  
FINE
```

6.61 Caricamento e salvataggio di un file con Addyplus

Un piccolo programma applicativo dovrebbe mostrare come caricare e salvare una colonna di numeri. A tal fine, il piccolo programma di sommatoria Addy verrà ampliato e convertito in Addyplus.

Guarda anche http://de.wikibooks.org/wiki/Gambas:_Rechnen#Addy_der_Totalizzatore

Ecco come appare la forma ora:

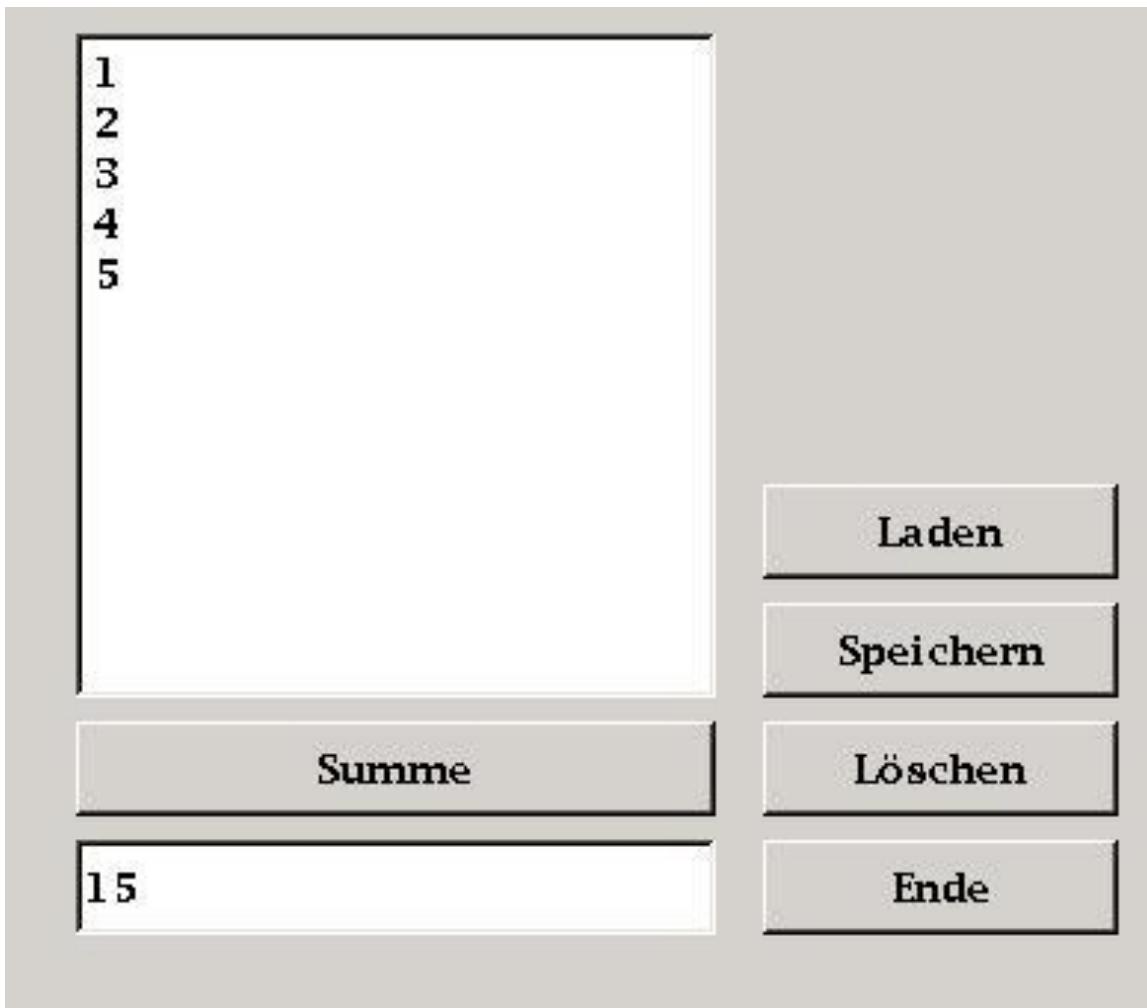


Fig.73

Quindi hai bisogno

- un'area di testo
- una casella di testo
- e 5 pulsanti di comando nel modulo per avviare il programma.

Ecco come appare il codice:

```

PUBLIC SUB Form_Open ()
    ME.Text = "Addy, il totalizzatore"
    Textarea1.Text = ""
    Textbox1.Text = ""
FINE
Pulsante PUBLIC SUB1_Click ()
    Testo DIM AS String
    Somma DIM AS Float
    Elenco DIM come stringa []
    DIM post AS String
    text = textarea1.Text
    list = Split (text, Chr (10))
    PER OGNI voce IN lista
        somma = somma + val (elemento)
    IL PROSSIMO
    textbox1.Text = sum
    CATTURARE
    Message.Info ("Si prega di non inserire un ritorno nell'ultima riga!
Premi il tasto backspace! ")
FINE
Pulsante PUBLIC SUB 3_Click ()
    Dialog.SaveFile ()
    File.Save (Dialog.Path, TextArea1.Text)
CATTURARE
SE ERRORE ALLORA RITORNO
FINE
PUBLIC SUB Button2_Click ()
IF Dialog.OpenFile () THEN RETURN
TextArea1.Text = File.Load (Dialog.Path)
FINE
Pulsante PUBLIC SUB 4_Click ()
SELEZIONA Message.Question ("Attenzione! Il campo di testo verrà
eliminato.", "Annulla", "Elimina")
    CASO 1
        RITORNO
    CASO 2
        Textarea1.Text = ""
    FINE SELEZIONA
FINE
Pulsante PUBLIC SUB 5_Click ()
ME.Close
FINE

```

Immettere una serie di numeri nell'area di testo, ad esempio

```

1
2
3

```

Quindi premere Salva. Immettere un nome file, ad esempio s1 e salvare la prima riga di numeri.

6.62 Comando Dir

Matrice del nome del file = Dir (Directory [, File pattern])

Questo comando restituisce una matrice di stringhe contenente i nomi dei file nella directory che corrisponde al modello di file. Se non è stato specificato alcun modello di file, tutti i file vengono caricati nell'array. Il modello di file può contenere gli stessi caratteri generali del comando LIKE.

Esempio:

```
PUBLIC SUB Form_Open ()  
File DIM come stringa  
PER OGNI file IN Dir (system.home, "")  
    Stampa file  
    IL PROSSIMO  
FINE
```

In questo esempio tutti i file nella directory home dell'utente corrente vengono visualizzati nella finestra diretta. Tuttavia, anche le sottodirectory vengono gestite come file.

esempio

```
'Stampa una directory  
SUB PrintDirectory (Directory AS String)  
    File AS String  
    PER OGNI file IN Dir (directory, "*.*")  
        Stampa file  
    IL PROSSIMO  
FINE
```

Variante di esempio: è necessaria una forma con un pulsante sulla forma. Aggiungi una directory esistente sul tuo PC al programma invece di / home / rho / gambastext /.

```
"File di classe Gambas  
Pulsante PUBLIC SUB1_Click ()  
    PrintDirectory ("/ home / rho / gambastext /")  
FINE  
SUB PrintDirectory (Directory AS String)  
    File DIM come stringa  
    PER OGNI file IN Dir (directory, "*.*")  
        Stampa file  
    IL PROSSIMO  
FINE
```

6.63 Crea nuova directory

È possibile creare una nuova directory con il comando MakeDir. Attenzione: funziona solo

- se hai inserito un percorso corretto,
- se hai il permesso di scrittura per questo percorso

- se non vuoi scrivere una nuova directory nella directory del progetto Gambas aperto. (vedi Application.Name)
- nessun file con lo stesso nome esiste in questo percorso.

Esempio:

È necessario un pulsante di comando per avviare il programma:

```
Pulsante PUBLIC SUB1_Click ()
  MKDIR system.home & "/ test"
  FINE
```

system.home indica la directory utente corrente / home / zbihrname.

Controlla con Konqueror se la directory è stata realmente creata.

È preferibile inserire una routine di errore nel caso in cui il percorso esista già o non sia disponibile l'accesso in scrittura.

6.64 Copia file

Il comando COPY copia un file da una directory a un'altra.

COPIA la directory / file di origine NELLA directory / file di destinazione

Il comando precedente copia un file dalla directory di origine alla directory di destinazione. Il file non deve necessariamente avere lo stesso nome del file di destinazione. Tuttavia, le sottodirectory non vengono copiate. (Notare che non è possibile copiare le directory in modo ricorsivo con questa funzione.)

Esempio:

```
' Esegue il backup del file di configurazione gambas nascosto
' La directory deve esistere
' e bisogna avervi accesso
COPY system.name & "./.gambas/gambas.conf" IN system.name &
"/gambas.conf.save"
```

System.name corrisponde alla tua directory home.

L'esempio produce un messaggio di errore se:

- La directory non esiste
- Non c'è accesso in scrittura alla directory
- Il file di destinazione esiste già.

Esempio2:

È necessario un pulsante di comando nel modulo per avviare l'esempio.

```
Pulsante PUBLIC SUB1_Click ()
  COPIA "/home/tuonome/.gambas/gambas.conf" A
```

```
"/home/tuonome/test/gambas.conf.save"
FINE
```

Se vuoi copiare più file contemporaneamente, il comando Dir ti aiuterà. Per favore, riferiscihttp:

//de.wikibooks.org/wiki/Gambas:_Datei

Come le directory possono essere copiate o spostate è qui http://de.wikibooks.org/wiki/Gambas:_Questions#Copy_directory¹⁷ http://descritto

6.65 Come si può interrogare la directory home dell'utente corrente?

Il comando Application.home non funziona più. Il comando Application.Name fornisce il percorso del programma attualmente in esecuzione, ma non la directory home.

La directory home dell'utente corrente può essere ottenuta con il comando system.home in Gambas 2, anche questo comando non funziona più. A questo scopo è ora necessario utilizzare la classe "Utente". Quindi il comando è: user.home

Esempio 1

```
PUBLIC SUB Form_Open ()
Sistema PRINT Home
FINE
```

Esempio2:

```
PUBLIC SUB Form_Open ()
ME.Text = "Informazioni di sistema"
textarea1.text = "Charset =" & system.Charset & Chr (13) & Chr (10) textarea1.text =
textarea1.text & "Domain =" & system.Domain & Chr (13) & Chr (10)

textarea1.text = textarea1.text & "Home directory =" & system.Home & Chr
(13) & Chr (10)
textarea1.text = textarea1.text & "Host =" & system.Host & Chr (13)
& Chr (10)
textarea1.text = textarea1.text & "Language =" & system.Language & Chr (13) & Chr (10)

textarea1.text = textarea1.text & "path =" & system.path & Chr (13)
& Chr (10)
textarea1.text = textarea1.text & "User =" & system.User
' Sistema PRINT
' Sistema PRINT Dominio
' Sistema PRINT Home
' Sistema di stampa Host
' Sistema PRINT Lingua
' Sistema PRINT Path
' Sistema PRINT Utente
FINE
Pulsante PUBLIC SUB1_Click ()
```

17 http://de.wikibooks.org/wiki/

```
ME.Chiudi  
FINE
```

Hai bisogno di un'area di testo e di un pulsante di comando per avviare il programma.

Esempio 3

```
PUBLIC SUB Form_Open ()  
PRINT user.Home  
FINE
```

6.66 Esiste un file?

L'esempio seguente controlla se la bash esiste o meno.

Esempio:

```
Pulsante PUBLIC SUB1_Click ()  
IF Exist (" / bin / bash ") = TRUE THEN  
Label1.Text = "Bash è disponibile"  
FINISCI SE  
IF Exist (" / bin / bash ") = FALSE THEN  
Label1.Text = "Errore: Bash non esiste"  
FINISCI SE  
FINE
```

È necessario un pulsante di comando e un campo etichetta per avviare il programma.

Funziona in modo simile:

```
GB.ExistFile (nome_file)
```

6.67 Carica e salva i file in un piccolo programma combinato

Un file test.txt viene caricato e salvato nel seguente programma.

Ecco come appare il programma finito:

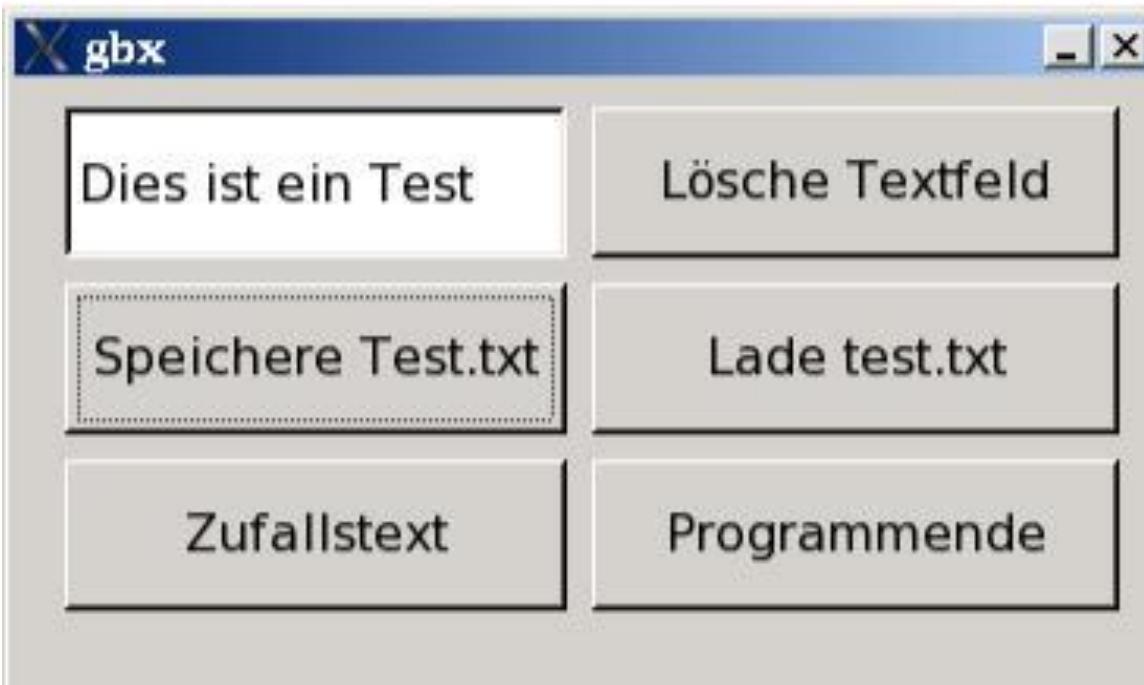


Fig.74

Quindi hai bisogno di:

- Un modulo
- Una casella di testo
- 5 pulsanti di comando

Il codice ha questo aspetto:

```
PUBLIC SUB Form_Open ()  
    Textbox1.Text = "Questo è un test"  
FINE  
Pulsante PUBLIC SUB1_Click ()  
'Viene salvato nella finestra di dialogo, cioè ci sono anche altri nomi di file  
possibile  
Dialog.SaveFile ()  
File.Save (Dialog.Path, TextBox1.Text)  
'File.Save ("test.txt", TextBox1.Text)  
'test.txt deve già esistere  
'altrimenti viene visualizzato un messaggio di errore.  
FINE  
PUBLIC SUB Button2_Click ()  
'Il caricamento avviene senza dialoghi  
IF Exist ("test.txt") THEN Textbox1.Text = File.Load ("test.txt")  
'test.txt deve essere presente!  
FINE  
Pulsante PUBLIC SUB 3_Click ()  
'La casella di testo viene eliminata  
    Textbox1.Text = ""  
FINE  
Pulsante PUBLIC SUB 4_Click ()  
«Fine del programma  
    ME.Chiudi  
FINE
```

```

Pulsante PUBLIC SUB 5_Click ()
'Genera testo casuale
DIM Z AS Inter
Lettera DIM come stringa
Rendi casuale ()
Textbox1.Text = ""
PER Z = DA 1 A 10
    Lettera = Chr $ (Int (Rnd (65.91)))
    TextBox1.Text = TextBox1.Text e lettera
IL PROSSIMO
FINE

```

6.68 Scrivere un file con WRITE

6.69 Lettura dei dati con READ

Il comando READ legge i dati da varie origini nel computer. Il modo più semplice per spiegarlo è caricare un file dal disco rigido nel display. Il seguente programma mostra come funziona.

Hai bisogno:

- un modulo
- un pulsante di comando

Il codice per farlo è il seguente:

```

PUBLIC SUB Form_Open ()
    Button1.Text = "Leggi test.txt"
    'Il pulsante di comando1 viene rinominato
    File.Save (application.path & "/test.txt", "This is a test")
    'Viene creato un file di prova nella directory corrente e
    salvato
    CATTURARE
    'Cogli eventuali errori
    SE ERRORE ALLORA RITORNO
    FINE
    Pulsante PUBLIC SUB1_Click ()
    'Il file di prova viene letto un po' alla volta e unito di nuovo hFile AS File

    s come stringa
    APRIRE "test.txt" PER LEGGERE COME #hFile
    MENTRE NON Eof (hFile)
        INGRESSO LINEA #hFile, s
        STAMPA s
        'Output nella finestra diretta
    WEND
    FINALMENTE 'Esegui sempre, anche in caso di errore
    CHIUDI #hFile
    CATTURARE
    'Esegui solo in caso di fallimento
        PRINT "Il file test.txt non può essere emesso."
    FINE

```

Questo programma salva un nome inserito in un file.

La prossima volta che avvierai il programma sarai accolto con questo nome.

Per fare questo, hai bisogno di:

- 1 forma
- 2 etichetta di testo
- 1 campo di immissione
- 1 pulsante di comando

```
' File di classe Gambas  
  
Pulsante PUBLIC SUB1_Click ()  
  
    settings ["Name"] = Campo di immissione.Testo  
    Impostazioni Salva  
  
    FINE  
  
PUBLIC SUB Form_Open ()  
  
    TextLabel2.Text = "Ciao" e impostazioni ["Nome", "Nessun nome"] & "!"  
  
    FINE
```

Ecco come appare il programma avviato:



Fig.75

6.70 Finestre di dialogo standard

Le finestre di dialogo standard semplificano la programmazione:

Il seguente programma di esempio mostra le finestre di dialogo standard per la scelta

- il carattere
- il colore
- del nome del file
- del percorso del file

Usato.

Sono necessari 1 area di testo e 3 pulsanti di comando per avviare il programma.

```
Pulsante PUBLIC SUB1_Click ()  
    IF Dialog.SelectFont () THEN RETURN  
    Textarea1.Font = Dialog.Font  
FINE  
PUBLIC SUB Button2_Click ()  
    IF Dialog.SelectColor () THEN RETURN  
    Textarea1.BackColor = Dialog.Color  
FINE  
Pulsante PUBLIC SUB 3_Click ()  
    Dialog.SaveFile ()  
    File.Save (Dialog.Path, TextArea1.Text)  
CATTURARE  
SE ERRORE ALLORA RITORNO  
FINE
```

La finestra di dialogo standard per la selezione della directory viene utilizzata nel seguente programma di esempio. La directory selezionata viene inserita nella casella di testo. Vengono utilizzati un pulsante e una casella di testo.

```
Pulsante PUBLIC SUB1_Click ()  
    Dialog.SelectDirectory ()  
    textbox1.Text = dialog.Path  
CATTURARE  
SE ERRORE ALLORA RITORNO  
FINE
```

6.71 teoria

Questa classe Dialog contiene metodi statici per chiamare le finestre di dialogo standard. È nel componente GB.QT.

caratteristiche

```
ColorFilterFontPathTitle
```

Metodi

```
OpenFileSaveFileSelectColorSelectDirectorySelectFont
```

6.72 Modifica delle dimensioni di una finestra di dialogo

In una finestra di dialogo non è possibile modificare semplicemente le dimensioni della finestra

```
dlgDialog.Height = nuovo valore ..
```

Per fare ciò, devi scegliere il seguente modo: Cambia Window.Border in Window.Resizable. Quindi puoi modificare le dimensioni. Quindi devi cambiare Window.Border di nuovo in Window.Fixed. Un po' imbarazzante, ma è così che funziona.

Naturalmente, è anche possibile programmare la guida a menu in Gambas, poiché l'inserimento dei comandi tramite i menu spesso rende un programma più chiaro e più facile da usare.

6.73 Primo menu

Per il nostro primo menu, inizieremo con una forma vuota. Nella finestra della forma facciamo clic con il tasto destro del mouse e selezioniamo la voce di menu editor ... dal menu contestuale che vi compare oppure premiamo la combinazione di tasti Ctrl + E.

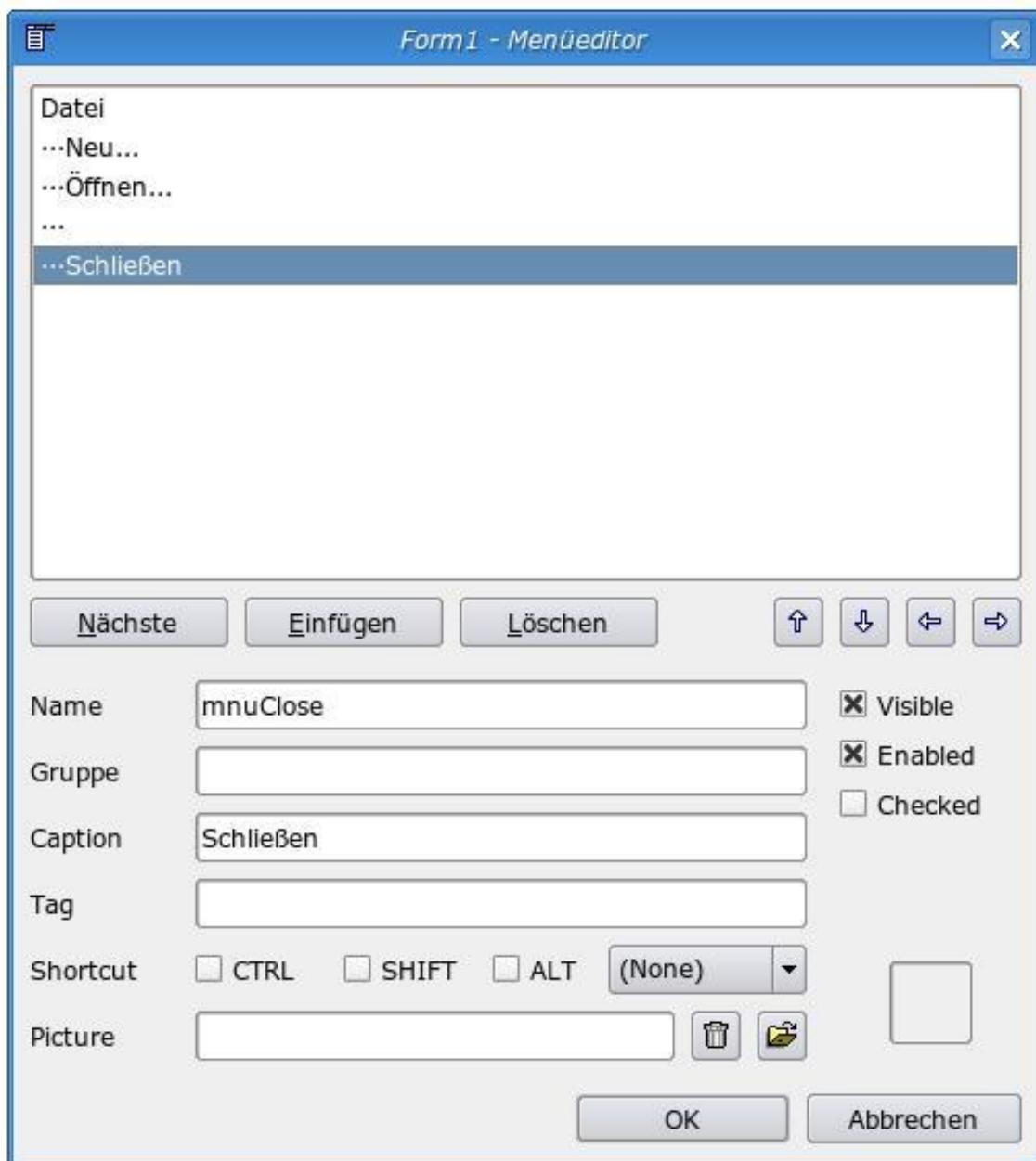


Fig.76 L'editor del menu Gambas

- Un clic sul pulsante Inserisci crea una nuova voce di menu
- L'editor del menu ora mostra diverse caselle di testo sotto il grande campo bianco della panoramica (nome, gruppo, didascalia, tag)
- Sotto Nome diamo alla voce di menu un nome: mnuFile sotto il quale possiamo indirizzarlo all'interno del programma
- e in Didascalia inseriamo un'etichetta: file

Con i campi di comando Avanti e Inserisci ora puoi creare nuove voci di menu.

Se lasci vuota la voce Didascalia, viene creata una linea di separazione invece di una voce di menu.

6.73.1 Sottomenu

Ora dobbiamo specificare che le altre voci di menu non sono i loro menu, ma piuttosto Sottomenu della prima voce. Per fare ciò, premiamo il pulsante con la freccia che punta a destra. Nella parte superiore del file

Nell'editor del menu, tre punti ... vengono visualizzati davanti al nome della voce.

Premendo più volte questo tasto, è possibile generare di conseguenza anche sottomenu, sottomenu, ecc.

Quando abbiamo creato tutte e cinque le voci di menu, possiamo uscire dall'editor di menu con OK.

6.73.2 Inserimento del codice

L'ultima cosa che dobbiamo fare è inserire il codice del programma che deve essere eseguito quando si fa clic su una voce di menu. Ma non è difficile, poiché una voce di menu cliccata si comporta come un pulsante cliccato.

Inseriamo semplicemente il seguente codice nella finestra del codice:

```
PUBLIC SUB mnuClose_Click ()  
    ME.Chiudi  
    FINE
```

Ora possiamo avviare il nostro programma con F5 e testare il menu. Quando si fa clic su Chiudi, il modulo viene chiuso.



Fig.77 Il nostro primo menu

6.74 Menu Opzioni

Un menu di opzioni è una voce di menu che può essere commutata avanti e indietro tra due stati come una casella di controllo. Per implementarlo nel nostro programma, utilizziamo la proprietà selezionata della voce di menu.

Supponiamo di avere un programma che emette una serie di dati e vogliamo dare all'utente la possibilità di ordinare anche l'output dei dati. Per fare ciò, creiamo prima una nuova voce di menu principale mnuData con l'etichetta Data e una voce di sottomenu mnuSorted con l'etichetta Sorted nell'editor di menu.

Infine abbiamo bisogno del seguente codice di programma:

```
PUBLIC SUB mnuSorted_Click ()  
    'Attiva / disattiva valore  
    mnuSorted.Checked = NOT mnuSorted.Checked  
  
    IF mnuSorted.Checked = TRUE THEN  
        PRINT "ordina su"  
    ALTRO  
        PRINT "smistamento"  
    FINISCI SE  
    FINE
```

Se viene richiamata questa voce di menu, il comando nella terza riga cambia il valore di Proprietà controllata ciascuno intorno. La quinta riga mostra come possiamo vedere lo stato corrente di Valuta la voce di menu nel resto del programma.



Fig.78 Un menu di opzioni

6.75 Menu dinamico

Se selezioni il menu File nella finestra del progetto Gambas e poi la voce Ultimi aperti, compare un menu con gli ultimi progetti sui quali hai lavorato. Per ovvie ragioni, non è possibile creare tali menu nell'editor di menu in fase di programmazione, ma devono essere creati dinamicamente mentre il programma è in esecuzione.

Per fare ciò, creiamo prima un nuovo sottomenu mnuRecent con l'etichetta Ultima apertura nell'editor del menu sotto la voce Apri Quindi aggiungiamo il seguente codice di programma al nostro programma:

```
$ aRecent AS NEW String []  
  
PUBLIC SUB Form_Open ()  
    'Dati di esempio  
    $ aRecent = ["File12", "File34", "File56", "File78", "File90"]  
  
    'Crea menu  
    UpdateMenu  
    FINE  
  
SUB UpdateMenu ()  
    DIM sFile AS String  
    DIM hMenuItem AS Object  
  
    'Elimina le vecchie voci  
    mnuRecent.Children.Clear  
  
    'Disattiva se non ci sono voci IF $ aRecent.Length = 0 THEN  
    mnuRecent.Enabled = FALSE  
  
    'Crea le singole voci  
    PER OGNI sFile IN $ aRecent  
        'Nuovo arrivato  
        hMenuItem = NUOVO menu (mnuRecent) AS "mnuRecentFile"  
  
        'Etichettatura  
        hMenuItem.Text = sFile  
    IL PROSSIMO  
    FINE  
  
PUBLIC SUB mnuRecentFile_Click ()  
    STAMPA ULTIMO Testo  
FINE
```

Le voci che creiamo hanno tutte lo stesso nome mnuRecentFile. Per poter distinguere quale delle voci è stata cliccata all'interno della funzione mnuRecentFile_Click (), utilizziamo l'oggetto LAST. LAST è un riferimento all'ultimo oggetto che ha attivato un evento (qui l'evento Click()). Con LAST.Text si accede alla proprietà text della voce che è stata cliccata per ultima.



Fig.79 Un menu dinamico

6.76 Menu contestuale

In molti programmi, viene visualizzato un menu contestuale dopo aver fatto clic con il pulsante destro del mouse. Di seguito vogliamo fornire un elemento di controllo con un semplice menu contestuale nel nostro programma Gambas.

Per fare ciò, posizioniamo un pulsante sul modulo e aggiungiamo il seguente codice di programma:

```
PUBLIC SUB Button1_MouseDown ()  
    Oggetto DIM hContext AS  
    DIM hMenuItem AS Object  
  
    SE Mouse.Right = VERO ALLORA  
        'Crea tu stesso il menu contestuale  
        hContext = NUOVO menu (ME) COME "mnuContext"  
  
        'Crea le singole voci di menu  
        hMenuItem = NUOVO menu (hContext) COME "mnuRot"  
        hMenuItem.Text = "Rosso"  
        hMenuItem = NUOVO menu (hContext) COME "mnuBlau"  
        hMenuItem.Text = "Blu"  
        hMenuItem = NUOVO menu (hContext) COME "mnuGruen"  
        hMenuItem.Text = "Verde"  
  
        'Visualizza il menu  
        hContext.Popup  
    FINISCI SE  
FINE  
  
PUBLIC SUB mnuRot_Click ()  
    Button1.Background = Color.Red  
FINE
```

```
PUBLIC SUB mnuBlau_Click ()  
    Button1.Background = Color.Blue  
FINE  
  
PUBLIC SUB mnuGruen_Click ()  
    Button1.Background = Color.Green  
FINE
```

Se si preme un pulsante del mouse su un elemento di controllo, viene chiamata la funzione `MouseDown ()` associata. All'interno di questi dobbiamo ancora verificare se è stato premuto il tasto destro del mouse. In tal caso, creiamo il menu e lo visualizziamo.

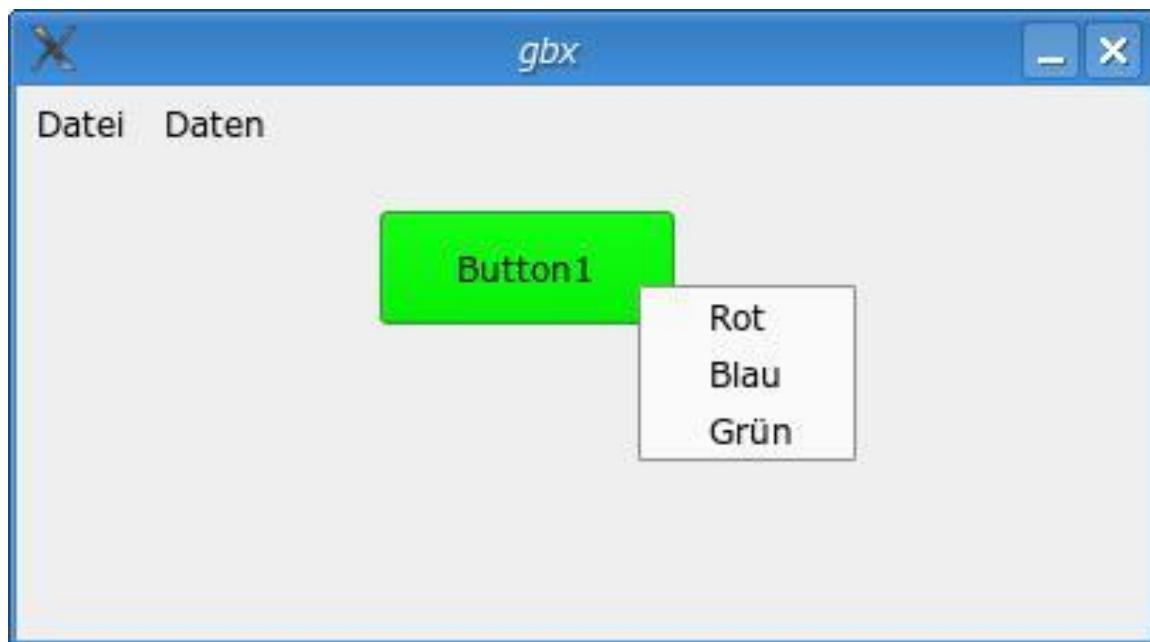


Fig. 80 Un menu contestuale

Attenzione! C'è ancora molto lavoro in corso in questa sezione !!

6.77 Programmazione database solo con Gambas

Non è necessario mysql per creare un semplice database di indirizzi e numeri di telefono.
Provalo passo dopo passo senza mysql.

Cominciamo con una tabella molto semplice

1 uno
2 due

Dovrebbe essere possibile salvarlo, modificarlo ed estenderlo. Altrimenti non abbiamo bisogno di ulteriori funzioni per il momento.

L'intera operazione può essere eseguita come un array di stringhe e visualizzata in un'area di testo.

Vedi Gambas: Arrays^{18 °}

Esempio:

Hai bisogno di un modulo, un'area di testo e 2 pulsanti di comando per avviare il programma.

```
' File di classe Gambas PUBLIC
SUB Form_Open () Textarea1.Text
= "1, uno"
    Textarea1.Text = Textarea1.Text & "2, Two"
FINE
PUBLIC SUB Button2_Click () Dialog.SaveFile ()
File.Save (Dialog.Path, TextArea1.Text) 'Immettere
db.txt come nome del file

CATTURARE
IF ERROR THEN
RETURN END
PUBLIC SUB Button1_Click () DIM
sLine AS String
sLine = File.Load ("db.txt") 'Il file
deve esistere
'altrimenti non può essere caricato.
textarea1.Text = sLine
FINE
```

Ora aggiungi alcune voci di tabella e salva di nuovo:

```
3 tre
4, quattro
5, cinque
```

Seleziona questi altri 3 record di dati e copiali negli appunti con Ctrl + C.

Con Ctrl + V li esegui di nuovo nella tua area di testo e salvi.

Elimina l'area di testo e ricarica il database. Ed ecco, il tuo mini-tavolo è ancora lì.

L'intera cosa sembra ancora un file di testo, ma anche una tabella corretta viene inizialmente salvata come file consecutivo. Nella nostra mini tabella, i singoli campi di dati sono separati da virgole, le singole righe (= record di dati) sono separate da un ritorno (= Chr (10)).

6.78 file CSV

Forse vuoi convertire la semplice tabella di numeri sopra in un formato di database CSV standard, come quello utilizzato da Excel o Openoffice come opzione di importazione.

18 Capitolo 6.157 a pagina 360

Il formato di file CSV viene spesso utilizzato per lo scambio di dati tra diversi programmi per computer (è un formato di scambio). L'uso del formato nei moduli utilizzati da Microsoft Excel si è affermato come uno pseudo-standard.

Il formato Excel CSV ha questo aspetto:

```
"1 uno"  
"2", "Due"
```

Ciò significa che, contrariamente alla nostra rappresentazione, le voci nei campi dati sono ancora fornite con virgolette. Devi solo fare attenzione che non ci siano virgolette o virgole nel campo dati stesso, perché questi 2 caratteri sono necessari per la struttura della tabella.

Un file CSV è un file di testo che contiene dati strutturati in una tabella e viene utilizzato principalmente per lo scambio di dati. L'abbreviazione CSV sta per valori separati da caratteri o valori separati da virgola, perché i singoli valori sono separati da un separatore speciale, in molti casi la virgola. I singoli record di dati sono solitamente separati da un'interruzione di riga (su Windows: CR LF = ritorno a capo, avanzamento riga; su Unix: solo LF; su Mac OS: solo CR). Il formato CSV è indipendente dalla codifica dei caratteri, dall'ordine dei byte e dai caratteri di interruzione di riga.

I file CSV hanno spesso l'estensione .txt anziché .csv e possono anche essere creati e modificati in qualsiasi editor di testo.

Proviamo a programmare la conversione CSV: per questo abbiamo bisogno di un nuovo pulsante di comando che chiamiamo CSV e del seguente codice:

```
Pulsante PUBLIC SUB 3_Click ()  
testo come stringa  
list AS String []  
posta AS String  
serie AS String  
text = textarea1.Text  
list = Split (text, Chr (10))  
textarea1.Text = ""  
PER OGNI voce IN lista  
Post IF = "" THEN  
riga = ""  
ALTRO  
riga = Chr (34) & post & Chr (34) & Chr (10)  
FINISCI SE  
textarea1.Text = textarea1.Text & row  
IL PROSSIMO
```

Sembra piuttosto carino. Soprattutto, la riga vuota alla fine non è fornita di virgolette. Ma sfortunatamente non ci sono virgolette intorno alle virgole. Quindi il programma deve ancora essere migliorato:

La stringa post deve essere cercata per virgole e quindi la virgola deve essere sostituita da ",". funziona meglio con il comando Sostituisci in Gambas. Utilizzare il tasto di aiuto F1 e l'elenco alfabetico dei comandi per vedere come viene utilizzato.

Per la nostra applicazione sostituiamo le virgolette con chr (34).

```

Pulsante PUBLIC SUB 3_Click ()
testo come stringa
list AS String []
posta AS String
serie AS String
text = textarea1.Text
list = Split (text, Chr (10))
textarea1.Text = ""
PER OGNI voce IN lista
Post IF = "" THEN
riga = ""
ALTRO
riga = Chr (34) e sostituisci $ (post, ", , Chr (34) & , & Chr (34))
& Chr (34) e Chr (10)
FINISCI SE
textarea1.Text = textarea1.Text & row
IL PROSSIMO

```

Provalo. Ad esempio, salva la tua mini-tabella come db1.csv e prova a importare la tabella in Openoffice come tabella nel foglio di calcolo. Funziona ??

Modificare la tabella in Open Office, salvarla di nuovo come file CSV Excel e provare a riportare il file nel programma. Funziona?

Sembra un espediente superfluo, ma chiunque abbia mai provato a salvare vecchi database sa quanto siano importanti i formati di dati standard e quanto siano importanti le opzioni di importazione ed esportazione.

6.79 Programmazione di database con Gambas e MySQL

6.79.1 Lavori preliminari Mysql

A differenza di MS Access, Mysql è il database standard nell'area Linux, ma non è così facile da usare come Access. Tuttavia, Mysql è molto buono perché è open source, molto diffuso e stabile. Inoltre, i formati dei file sembrano non essere modificati arbitrariamente, come spesso accadeva con MS Access senza motivo apparente.

Quindi dovresti familiarizzare un po' con Mysql prima di colpire Mysql con Gambas. Altrimenti, sarai rapidamente frustrato dagli errori di accesso negato.

Se non sai cos'è un database, dai un'occhiata alla pagina seguente:

<http://www.madeeasy.de/2/datbank.htm>

L'impostazione di mysql non è più un grosso problema sotto Linux, poiché tutte le attuali distribuzioni standard contengono i pacchetti mysql più importanti.

6.79.2 Impossibile connettersi al server MySQL locale tramite socket ...

Se vuoi che Mysql vada sotto Linux, il seguente messaggio di errore è sempre fastidioso.

Impossibile connettersi al server MySQL locale tramite socket ...

Qual è il motivo di questo messaggio di errore? Il più delle volte, il problema è già all'avvio di Linux. Il demone mysql non verrà avviato. Per correggere l'errore devi andare nel tuo editor runlevel. Lì si attiva Mysql in modo che il database venga avviato all'avvio.

Il modo migliore per avviare YAST sotto Suse è con la password di root. Quindi inserisci quanto segue:

YAST - SYSTEM / RUNLEVEL EDITOR / Servizi

e attiva mysql, se non è già attivo.

6.79.3 Mysql dalla console

Dopo l'installazione, avvia il tuo mysql dalla console con:

/etc/init.d/mysql start

Viene visualizzato il seguente messaggio: non riuscito (non riuscito) o

Avvio del servizio MySQL completato (riuscito)

Posso avviare mysql solo sul mio computer con diritti di root.

Puoi fermare mysql con:

/etc/init.d/mysql stop

Rapporto:

Chiusura del servizio mysqldone

Avvia di nuovo mysql e poi entra

mysql

Rapporto:

Benvenuto nel monitor MySQL I comandi terminano con; o \g.
Il tuo ID di connessione MySQL è 1 per la versione server: 4.0.18
Digita "help;" o "\ h" per chiedere aiuto. Digita "\ c" per cancellare il buffer

```
mysql>
```

Dai un'occhiata allo stato di mysql e inserisci lo stato:

```
mysql> stato
```

Rapporto:

```
-----
mysql Ver 12.22 Distrib 4.0.18, per suse-linux (i686)
ID connessione: 2
Database corrente:
Utente attuale: root @ localhost
SSL: Non in uso
Cercapersona corrente: Di meno
Utilizzando outfile:
Versione server: 4.0.18
Versione protocollo: 10
Connessione: Host locale tramite socket UNIX
Set di caratteri client: latin1
Set di caratteri del server: latin1
Presa UNIX: /var/lib/mysql/mysql.sock
Tempo di attività: 3 min 37 sec
Discussioni: 1 Domande: 3 Query lente: 0 Apre: 6 Flush tabelle: 1
Tabelle aperte: 0 query al secondo in media:
0.014
```

Il database corrente (non ancora caricato), l'utente corrente e la connessione sono importanti per Gambas.

Se hai già assegnato una password mysql, puoi accedere a mysql solo con

```
mysql -p
```

e quindi inserendovi la password. Questa password non ha nulla a che fare con la password di Linux ed è anche crittografata da mysql, quindi non è disponibile in testo normale. Prendine nota, ti servirà anche per gambas e mysql.

Quindi dai un'occhiata ai database esistenti, ad esempio il test del database di prova e inserisci Lei un:

```
USE test;
```

Rapporto:

```
mysql> USE test;
Database cambiato
```

Non lasciarti confondere dal messaggio: Database modificato.

Ora mostra le tabelle esistenti in mysql, ma non dimenticare il punto e virgola dopo la tua voce, altrimenti mysql sarà bitch.

```
MOSTRA TAVOLI;
```

Poiché alcune cose possono già essere trovate nel test sul mio computer, viene visualizzato il seguente messaggio:

```
+-----+
| Tables_in_test |
+-----+
| NewTable      |
| __RekallObjects |
| stagionatura   |
| t1            |
| t2            |
| test          |
+-----+
6 file in serie (0,00 sec)
```

Ora puoi dare un'occhiata alla singola tabella, ad esempio:

```
SELEZIONARE * DALLA spezia;
```

E ottengo una meravigliosa tabella delle spezie: mysql> SELECT * FROM spice;

```
+-----+-----+-----+-----+-----+
| no | Cognome    | prezzo | moneta | quantità | unità |
+-----+-----+-----+-----+-----+
| 1 | anice      | 4.6 | euro   | 6.8 | kg    |
| 2 | Pepe        | 5 | Euro   | 2 | 4 °   |
| 3 | zafferano   | 5 | e     | 3 | 3    |
| 4 ° | sale        | 4 ° | e     | 4 ° | 4 °   |
| 5 | curry        | 5 | e     | 5 | 5    |
| 6 ° | aglio       | 5 | e     | 5 | 5    |
+-----+-----+-----+-----+-----+
6 ° righe in serie (0,02 sec)
```

Prova a creare un tavolo delle spezie così carino per te stesso.

Per ulteriore sintassi mysql consiglio il seguente sito molto utile:

<http://www.selflinux.org/selflinux/html/mysql.html> - Sfortunatamente il collegamento è interrotto :-(È meglio seguire l'esempio del cliente lì.

6.79.4 Lavorare con il database manager Gambas

Il database manager di Gambas è adatto per visualizzare, creare e modificare nuovi database mysql in modo molto rapido e chiaro.

Puoi trovarlo nella finestra del progetto sotto la voce di menu Strumenti e Database Manager.

Quando lo avvii, ti chiede una password criptica che devi inserire, anche se stai ancora usando mysql come root e senza password.

Se inserisci semplicemente Invio qui, viene visualizzato un messaggio di errore: Inserisci almeno 8 caratteri
in tedesco: è necessario utilizzare una password con almeno 8 lettere.

Se si è superato con successo questo ostacolo relativo alla password, tutti i server di database vengono visualizzati in una visualizzazione ad albero.

Solo la voce root @ localhost (mysql) funziona per me . Se clicchi su questo, vengono visualizzati tutti i database che possono essere raggiunti, ad esempio test. Se fai clic su uno dei database, appariranno le tabelle lì. Se fai clic su una tabella, puoi ora modificare la struttura nella schermata di destra, impostare nuovi indici e inserire nuovi dati.

Prova tutto e salva le modifiche facendo clic sul simbolo del dischetto.

6.79.5 Diritti utente MySQL

6.79.6 MySQL, Apache e phpMyAdmin

I programmi Apache come server web e phpMyAdmin come interfaccia di accesso si sono dimostrati molto utili per visualizzare i database MySQL. Prova a installare questi due programmi sul tuo computer. Quindi familiarizza un po 'con phpMyAdmin.

Rende la risoluzione dei problemi molto più semplice se si dispone di un ulteriore accesso rapido ai database MySQL sotto forma di phpMyAdmin.

6.80 Programmazione database con Gambas e SQLite

Non è sempre necessario configurare un server completo per una piccola soluzione di database. SQLite offre la possibilità di utilizzare i driver necessari, disponibili per Gambas, per creare un DB in un file senza server.

Un database SQLite può essere creato utilizzando il programma di console sqlite, dopo aver installato i driver necessari, con Knoda o, ovviamente, con il database manager di Gambase.

Una descrizione più dettagliata del database può essere trovata sotto <http://de.wikipedia.org/ wiki / sqlite .>

Il database è integrato dalle variabili .Type e .Name della connessione, dove "sqlite" sta per .Type e il percorso assoluto del file di database per .Name. I diritti dell'utente si basano sui diritti del file.

6.81 Creazione di una connessione al database

Esempio di connessione al database. Il componente gb.db deve essere incluso nel progetto:

```
Connessione $ hConn AS PRIVATA
PUBLIC SUB GoConnect ()
    DIM sName AS String
    PROVA $ hConn.Close
    sName = "DatabaseName"
    CON $ hConn
        .Type = "mysql"
        .Host = "DatabaseServer"
        .Login = "ID utente database"
        .Password = "password utente database"
    FINISCI CON
    $ hConn.Name = sName
    $ hConn.Open
CATTURARE
Message.Error (Error.Text)
FINE
PUBLIC SUB Form_Open ()
    $ hConn = NUOVA connessione
    GoConnect
FINE
PUBLIC SUB Form_Close ()
    $ hConn.Close
FINE
```

6.82 Mostra database disponibili

Viene stabilita una connessione al database con il database standard (postgresql: 'template1'; mysql: 'mysql' ...), che è sempre presente sotto il rispettivo sistema di database, e il numero e interrogando la 'Collezione' 'Database' disponibili per l'utente connesso Database disponibili. A tale scopo, viene specificata una stringa vuota come nome del database. Il ".Type" deve essere impostato in base al tipo di database.

Ad esempio come procedura in un modulo:

```
Connessione database PRIVATA AS Connection
Nome database PRIVATO COME stringa
PROCEDURA lista database ()
    Variante DIM Element AS
    Connessione al database = NUOVA connessione
    CON connessione al database
        .Genere      = "postgresql"
        .Ospite      = "Nome computer"
        .Accesso     = "Nome utente"
        .Password   = "password"
        .Cognome    = ""
    FINISCI CON
    Connessione al database Apri
    PRINT "Numero di database:" & DB.Databases.Count
    PER OGNI elemento IN DB.Database
        Stampa "Nome database:" & Element.Name
    IL PROSSIMO
    Connessione al database Chiudi
FINE
```

6.83 Collegamenti

- <http://software.newsforge.com/software/04/11/30/1653243.shtml?tid=72&tid=132>
Esempio di un'applicazione mysql di Gambas
- <http://www.postgresql.de/> Sistema di database PostgreSQL
- <http://dev.mysql.com/doc/refman/5.1/de/index.html> Manuale del sistema di database MySQL

La stampa funziona sotto Gambas con l'oggetto stampante. Può essere trovato nel componente GB.QT.

Vedi Componenti Gambas.

Se sono necessarie ulteriori informazioni sull'oggetto stampante, selezionare la funzione di aiuto in Gambas con F1. Nel browser della Guida, è possibile immettere la parola chiave Stampante nel campo RICERCA.

6.84 Poche righe in stampa diretta

È possibile controllare la stampante senza finestra di dialogo. Il seguente programma mostra come funziona. Hai bisogno:

- un modulo
- un pulsante di comando
- un'area di testo

Ottieni questi controlli dalla casella degli strumenti con F6.

Quindi incolla il seguente codice:

```
PUBLIC SUB Form_Open ()
ME.Text = "Stampa programma con il comando stampante"
Larghezza ME = 400
ME.Height = 400
ME.Center
Button1.Text = "Stampa in formato orizzontale"
Button1.Width = 200
Pulsante1.X = 16
Pulsante1.Y = 24
Textarea1.Text = "Questo è un test per la stampante!"
Textarea1.X = 16
Textarea1.Y = 64
Textarea1.Width = 312
Textarea1.Height = 248
FINE
Pulsante PUBLIC SUB1_Click ()
Printer.Colormode = 0
'Imposta la stampa in bianco e nero
Printer.Orientation = 0
'Ritratto
Draw.Begin (Printer) 'Inizializza l'espressione Draw.Text
(TextArea1.Text, 200, 200)
' Stampa il contenuto di TextArea nell'angolo in alto a sinistra Draw.End
'La pagina viene inviata alla stampante.
FINE
```

La prima parte del programma è superflua per la stampa. Semplifica solo l'impostazione dei controlli nel modulo e viene eseguito immediatamente all'avvio del programma. (Sottomodulo aperto) La stampa effettiva viene eseguita solo dopo aver fatto clic su Button1. Lì troverai i comandi della stampante.

6.85 Stampa con finestra di dialogo

È possibile controllare la stampante con una finestra di dialogo di selezione per selezionare la stampante desiderata prima di stampare. Per fare ciò, aggiungi il comando: Printer.Setup () all'esempio "Poche righe nella stampa diretta"

Il seguente programma mostra come funziona. Hai bisogno:

- un modulo
- un pulsante di comando
- un'area di testo

Ottieni questi controlli dalla casella degli strumenti con F6.

Quindi incolla il seguente codice:

```
PUBLIC SUB Form_Open ()  
ME.Text = "Stampa programma con il comando stampante"  
Larghezza ME = 400  
ME.Height = 400  
ME.Center  
Button1.Text = "Stampa in formato orizzontale"  
Button1.Width = 200  
Pulsante1.X = 16  
Pulsante1.Y = 24  
Textarea1.Text = "Questo è un test per la stampante!"  
Textarea1.X = 16  
Textarea1.Y = 64  
Textarea1.Width = 312  
Textarea1.Height = 248  
FINE  
Pulsante PUBLIC SUB1_Click ()  
Printer.Setup () 'Seleziona stampante  
Printer.Colormode = 0  
'Imposta la stampa in bianco e nero  
Printer.Orientation = 0  
'Ritratto  
Draw.Begin (Printer) 'Inizializza l'espressione Draw.Text  
(TextArea1.Text, 200, 200)  
' Stampa il contenuto di TextArea nell'angolo in alto a sinistra Draw.End  
'La pagina viene inviata alla stampante.  
FINE
```

La prima parte del programma è superflua per la stampa. Semplifica solo l'impostazione dei controlli nel modulo e viene eseguito immediatamente all'avvio del programma. (Sottomodulo aperto) La stampa effettiva viene eseguita solo dopo aver fatto clic su Button1. Lì troverai i comandi della stampante.

6.86 Stampa in formato orizzontale

A tale scopo è possibile adattare l'esempio sopra citato "Poche righe in stampa diretta". Modificare il comando Printer.Orientation = 0 in 1 Il valore per il formato verticale = 0 e per il formato orizzontale = 1

6.87 Stampa di un grafico

6.88 Stampa su un file PS Postscript

6.89 Annulla stampa

6.90 Teoria delle classi di stampanti

In Gambas, l'oggetto stampante è un dispositivo grafico, quindi usa i metodi della classe Draw per stampare con esso.

Esempio:

```
IF Printer.Setup () THEN RETURN  
'Attiva la finestra di dialogo
```

```
Draw.Begin (Printer) 'Inizializza l'estrazione  
Draw.Text (TextArea1.Text, 200, 200) 'Stampa il contenuto dell'area di testo nell'angolo  
superiore sinistro della stampante. Draw.Foreground = Color.Red  
  
Draw.Rect (500, 200, 32, 32) 'Stampa un piccolo rettangolo Draw.End' La  
pagina viene inviata alla stampante.
```

```
Printer.NewPage 'Si accede a una nuova pagina.
```

```
La classe della stampante è statica.
```

6.90.1 Proprietà della stampante

- ColorMode 'Bianco e

nero o Colore

- Copie

'Numero di copie

- File

'Stampa su un file PS

- FromPage
' Non iniziare a stampare fino a questa pagina

- Alla pagina
'Termina la stampa su questa pagina

- Altezza

'lato

lunghezza

- Larghezz

a 'larghezza

della pagina

- MaxPage

'Il numero di pagina più grande che puoi scegliere

- MinPage

'Il numero di pagina più piccolo che puoi scegliere

- Cognome

'Nome della stampante

- Orientamento

'Ritratto o paesaggio

- Risoluzione

'Risoluzione in

DPI

- Dimensione

" A3, A4 lettera ecc.

6.90.2 Metodi di stampa

- Aborto

'Annulla stampa

- Configurazione

'Avvia finestra di

dialogo di stampa

- Nuova pagina

'Avanzamento modulo Inizia una nuova pagina

6.90.3 Costanti della stampante

Black 'black and white

Espressione del colore del colore

296

Paesaggio del paesaggio

Formato verticale ritratto

6.91 Printer.Print

Il popolare comando VB Printer.Print purtroppo non è ancora disponibile in Gambas.
Forse puoi aggiungerlo.

6.92 Browser HTML

Puoi facilmente visualizzare pagine HTML con Gambas. Il seguente programma mostra come funziona.
Serve per richiamare il Gambaswikibook localmente sul computer.

Hai bisogno di un modulo con

- una casella di testo
- un campo etichetta
- un campo del browser web

Per ottenere il campo del browser web, devi prima avere i componenti Gambas

- gb.qt.kde e
- gb.qt.kde.html

al tuo progetto.

Come farlo è sotto Componenti Gambas¹⁹ spiegato.

Ecco come apparirà la tua forma in modalità progettazione.

19 Capitolo 12.18 a pagina 424

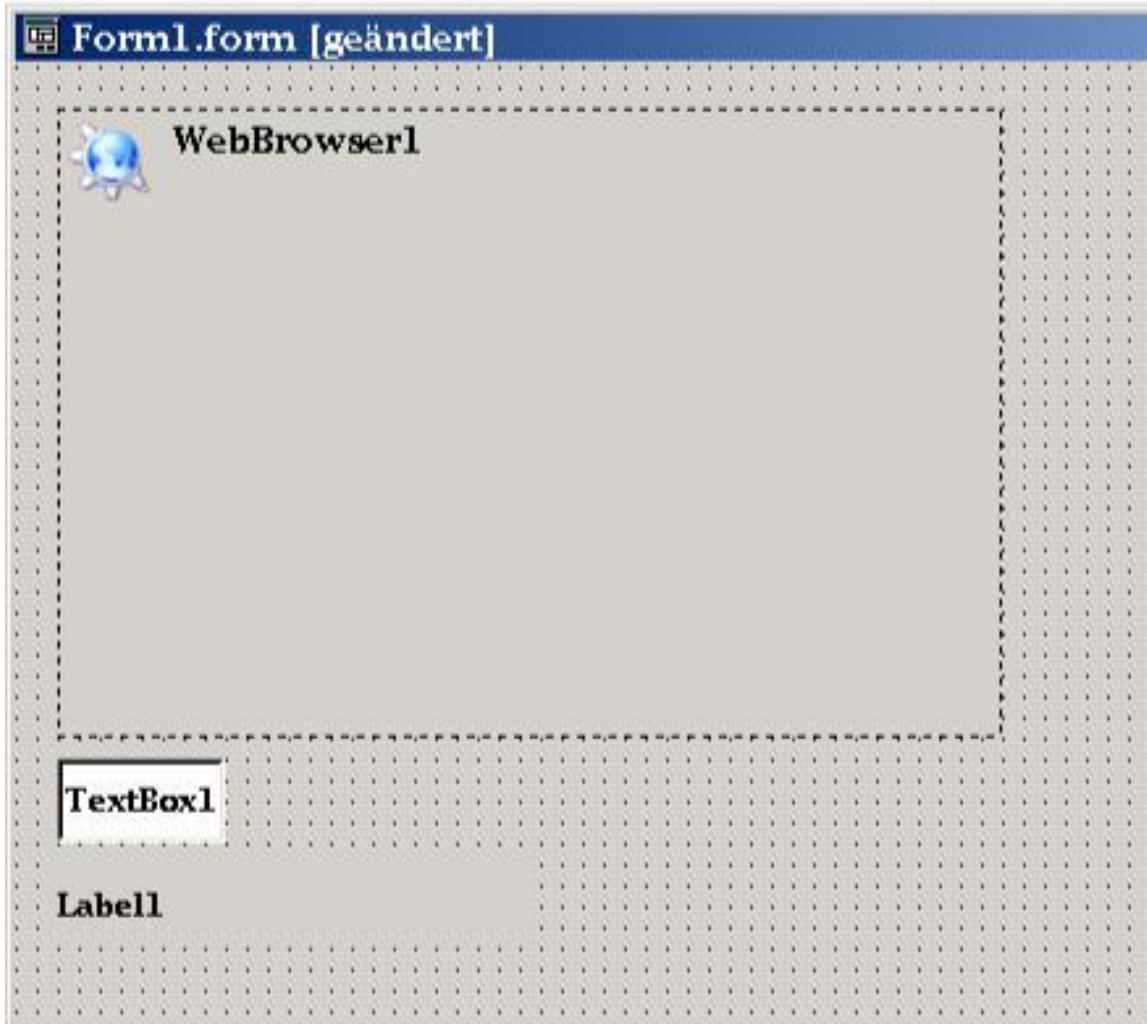


Fig.81

Aggiungi il seguente codice al tuo programma:

```
' File di classe Gambas
PUBLIC SUB Form_Resize ()
    textbox1.Move (0, 0, ME.ClientW)
    WebBrowser1.Move (0, textbox1.H, ME.ClientW, ME.ClientH - textbox1.H -
label1.H)
    label1.Move (0, WebBrowser1.Y + WebBrowser1.H, ME.ClientW)
FINE
PUBLIC SUB Form_Open ()
    textbox1.Text = "file: / home / USERS / wiki / Gambas"
    txtLink_Activate
FINE
PUBLIC SUB WebBrowser1_Change ()
    textbox1.Text = LAST.Path 'DEC
    Application.Busy
FINE
PUBLIC SUB WebBrowser1_Click ()
    textbox1.Text = "" & LAST.Link 'INC
    Application.Busy
```

```

FINE
PUBLIC SUB WebBrowser1_Link ()
    label1.Text = LAST.Link
FINE
PUBLIC SUB txtLink_Activate ()
    DIM sPath AS String
    sPath = textbox1.Text
    WebBrowser1.Path = Trim (sPath)
FINE

```

Affinché il programma possa funzionare devi impostare una copia locale di Gambas Wikibook sul tuo computer, vedi Configurazione di una copia locale^{20th}.

Quindi puoi configurare il programma come browser della guida di Gambas sul tuo computer e scaricare sempre le ultime pagine di Gambas dal GambasWikibook.

Solo lo scorrimento indietro non funziona ancora. Devi fare clic su Torna a Gambas nella riga superiore, quindi torni al menu principale.

Allo stesso modo, puoi configurare altri Wikibook sul tuo computer.

Ovviamente tutto funziona anche con Konqueror. È quindi anche possibile scorrere indietro in Konqueror.

6.93 Browser avanzato

Per risolvere il problema dell'impossibilità di tornare alla pagina, ora aggiungiamo tre pulsanti al form sopra: Button1, Button2 e Button3.

Infine, incorporeremo il seguente codice nel nostro programma: Wikibrowser2 Form1.class²¹

²⁰ Capitolo 0.52 a pagina 45
<http://de.wikibooks.org/wiki/Gambas%3A%20Html%3A%20Wikibrowser2%20Form1>.

²¹ classe



Fig.82 Browser Wiki 2

6.94 Browser Web Gambas

Un piccolo browser web è incluso con i programmi di esempio Gambas forniti. È molto pratico e facilmente espandibile.

6.95 Esempio di codice per l'interfaccia RS 232

```
PUBLIC sub Write_Serial (una stringa AS)
IF Sport.status = Net.Inactive THEN
    Messaggio ("apri prima la porta")
ALTRO
    chkrts.Value = TRUE
    ATTENDERE 0.01
    SCRIVI # Sport, a, Len (a)
    ATTENDERE 0.5
    chkrts.Value = FALSE
FINISCI SE
```



Il dispositivo seriale utilizza RTS UP per accettare i dati. Quando RTS è disattivato, questi dati vengono trasmessi.

Puoi indirizzare e utilizzare la shell Linux di Gambas. Di seguito viene mostrato come eseguire questa operazione

Esempi:

6.96 Ls sotto Gambas

Il seguente programma di esempio mostra come i file in una directory sul disco rigido possono essere visualizzati in un'area di testo.

Hai bisogno di un'area di testo e di un pulsante di comando per avviare il programma.

```
' File di classe Gambas
PUBLIC Content AS String
'Ottieni il contenuto di una directory
PUBLIC SUB Button1_Click ()
    SHELL "ls -la > /tmp/result" WAIT Content =
    File.Load ("/tmp/result") Textarea1.text =
    content
FINE
```

È possibile rimuovere la stringa con il nome Content dal programma se si riempie l'area di testo direttamente con il caricamento del file.

```
Textarea1.text = File.Load ("/tmp/risultato")
```

6.96.1 Un altro esempio: guarda le sottodirectory

Hai bisogno di un'area di testo e 2 pulsanti di comando per avviare il programma.

```
PUBLIC SUB Form_Open ()
    ME.Text = "Mostra sottodirectory - scegli un percorso"
    ME.X = 200
    ME.Y = 50
    Larghezza ME = 1200
    ME.Height = 600
    Button1.Text = "Mostra sottodirectory"
    Pulsante1.X = 16
    Button1.Y = 16
    Button1.Width = 300
    textare1.Text = ""
    textare1.x = 16
    textare1.y = 56
    textare1.width = 700
    textare1.height = 500
    Button2.Text = "Fine"
```

```
Button2.X = 340
FINE
Pulsante PUBLIC SUB1_Click ()
DIM t AS stringa
IF Dialog.SelectDirectory () THEN RETURN
t   = Dialog.Path
SHELL "ls -ld" & t & "*/. > / tmp / risultato "WAIT
Textarea1.text = File.Load ("/ tmp / risultato")
FINE
PUBLIC SUB Button2_Click ()
ME.Chiudi
FINE
```

Nel primo sottoprogramma Form_Open vengono effettuate solo alcune impostazioni per i 3 elementi di controllo. Il codice veramente importante è nel 2, sottoprogramma Button1_Click: prima viene dichiarata una stringa t. Quindi si apre una finestra di dialogo per la selezione della directory. La directory selezionata (Dialog.path) viene passata alla stringa t. Con il comando della shell tutte le sottodirectory di Dialogpath vengono salvate in un file temporaneo. Questo file viene caricato e visualizzato nell'area di testo.

6.97 wc = numero di parole tra i Gambas

Nel seguente programma il comando Linux wc, che conta le righe, le parole ed i caratteri di qualsiasi testo, viene utilizzato all'interno di un programma Gambas.

Per avviare il programma è necessario:

- 2 pulsanti di comando
- 1 area di testo (il testo da esaminare è inserito qui)
- 1 casella di testo (il risultato del conteggio parole viene visualizzato qui)

Il codice per questo:

```
PUBLIC t AS String
PUBLIC SUB Form_Open ()
    Textarea1.Text = "Questo è un test"
    Textbox1.Text = ""
FINE
Pulsante PUBLIC SUB1_Click ()
    File.Save (system.Home & "/test.txt", TextArea1.Text)
    CATTURARE
    SE ERRORE ALLORA RITORNO
FINE
PUBLIC SUB Button2_Click ()
    SHELL "wc" & system.Home & "/test.txt> / tmp / result" WAIT
    t = File.Load ("/ tmp / risultato")
    textbox1.Text = t
FINE
```

- Il comando 1 salva il contenuto dell'area di testo in un file test.txt.
- Con il comando 2, questo file test.txt è soggetto a un conteggio parole. il risultato viene salvato nel file / tmp / result.
- Questo file viene quindi ricaricato nella casella di testo e visualizzato.

Un po 'ingombrante ma funziona. La stringa t è effettivamente superflua e può essere rimossa dal programma se presente

```
textbox1.Text = File.Load ("/tmp / risultato")
```

scrive.

6.98 Caricamento del componente di compressione

Per creare un file compresso con Gambas, è necessario il componente Gambas gb.compress. Puoi attivarlo tramite la pagina del progetto Gambas. A tale scopo, fare clic sulla voce di menu Progetto e quindi su Proprietà. Seleziona Componenti dalle schede offerte lì. Trova gb.compress dall'elenco e seleziona la casella di fronte ad esso.

Vedi anche componenti²²

6.99 programma di esempio

Un file di testo viene generato nel seguente esempio. Questo viene salvato non compresso. Quindi viene compresso e salvato. Il file compresso viene caricato e generato. Quindi il file compresso viene nuovamente decompresso e visualizzato nella finestra diretta con PRINT.

Hai bisogno:

- 1 area di testo
- 7 campi di comando (pulsanti di comando)

per avviare il programma.

Il codice sorgente:

```
' File di classe Gambas File
PUBLICTxt AS String File
PUBLICZip AS String File
PUBLICUnzip AS String

PUBLIC SUB Form_Open ()
    Textarea1.Text = ""
    ME.Text = "Compressione con Gambas"
    FileTxt = System.Home & File.Separator & "ziptest.txt"
    FileZip = System.Home & File.Separator & "ziptest.txt.gz"
    FileUnzip = System.Home & File.Separator & "unziptest.txt"
FINE

Pulsante PUBLIC SUB1_Click ()
s come stringa
sp AS stringa
n AS intero
s = "Questo è un test"
```

22 Capitolo 12.18 a pagina 424

```
PER n = 1 A 10
    s = s & Str (n)
    sp = sp & s
    IL PROSSIMO

    PRINT sp
    TextArea1.Text = sp
FINE

PUBLIC SUB Button2_Click ()
    'Salva file di prova
    File.Save (FileTxt, TextArea1.Text)
    CATTURARE
    SE ERRORE ALLORA RITORNO
FINE

Pulsante PUBLIC SUB 3_Click ()
    DIM Cp AS Compress
    'DIM Cp COME NUOVO Compress
    Cp = NEW Compress
    Cp.Type = "zlib"
    Cp.File (FileTxt, FileZip, Cp.Max)
FINE

Pulsante PUBLIC SUB 4_Click ()
    DIM sLine AS String
    sLine = File.Load (FileZip)
    'Il file deve esistere
    'altrimenti non può essere caricato.
    PRINT sLine
    'TextArea1.Text = sline
FINE

Pulsante PUBLIC SUB 5_Click ()
    DIM UCp AS Uncompress
    UCp = NUOVO Uncompress
    UCp.Type = "zlib"
    UCp.File (FileZip, FileUnzip)
FINE

Pulsante PUBLIC SUB 6_Click ()
    DIM sLine AS String
    sLine = File.Load (FileUnzip)
    'Il file deve esistere
    'altrimenti non può essere caricato.
    PRINT sLine
    'TextArea1.Text = sline
FINE

Pulsante PUBLIC SUB 7_Click ()
    ME.Chiudi
FINE
```

6.100 Ci sono diversi giochi negli esempi forniti

- Spazio profondo
- RobotFindsKitten
- Serpente
- Solitario

6.101 Le torri di Hanoi

Prossimamente, guarda anche: <http://www.vb-kurs.cagaertig.de/hanoi.htm>

6.102 Sprint un mini-gioco per 2

Un semplice gioco per 2 giocatori. Chi ha messo l'ultima pietra ha vinto.

Hai bisogno:

- 40 caselle di testo (gemito)
- 2 aree di testo
- 4 pulsanti di comando

Codice sorgente (veloce e sporco):

- Sprint Form1.class²³
- Sprint Form1.form²⁴

²³ <http://de.wikibooks.org/wiki/Gambas%3A%20Spiele%3A%20Sprint%20Form1.class>

²⁴ <http://de.wikibooks.org/wiki/Gambas%3A%20Spiele%3A%20Sprint%20Form1.form>

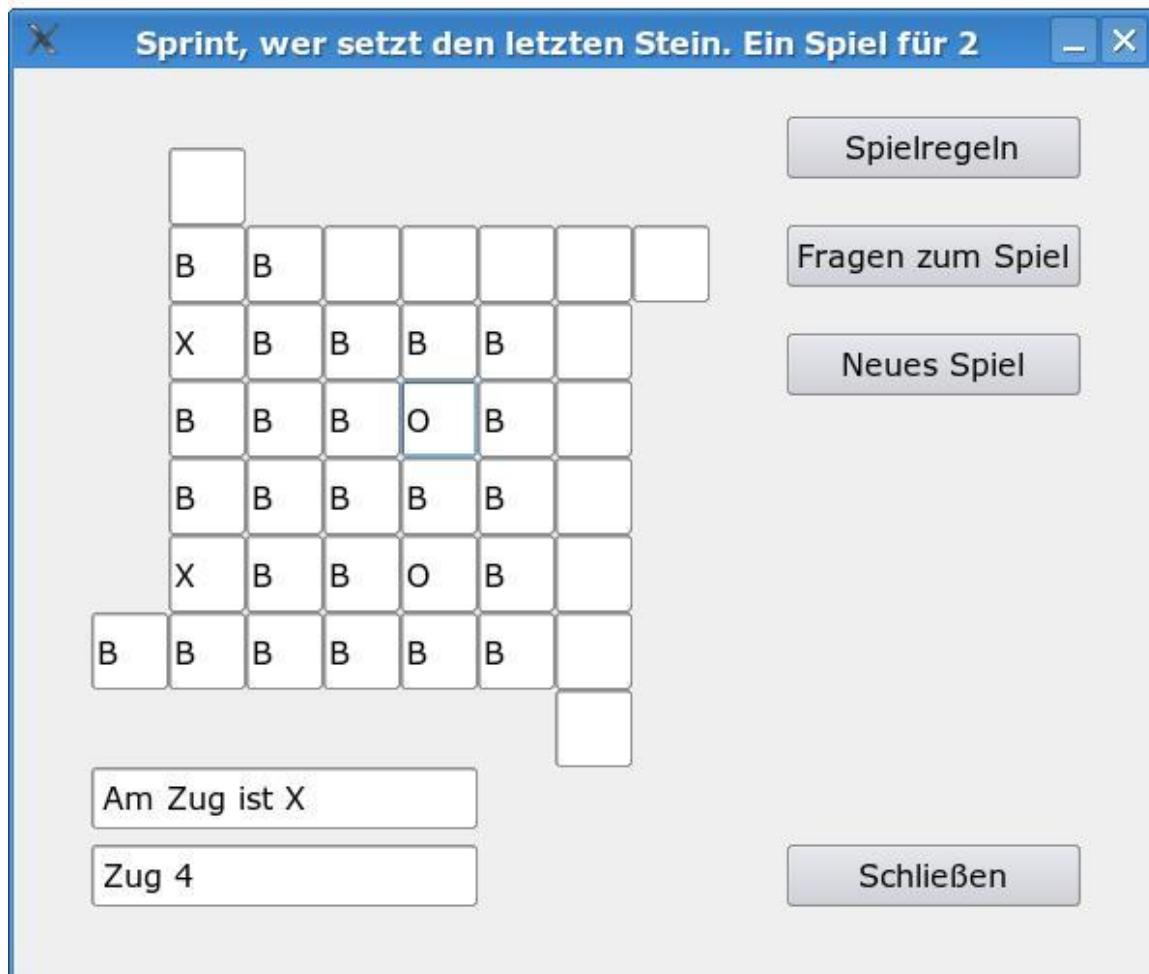


Fig.83

Questa programmazione necessita di un grande miglioramento. C'è una tipica soluzione per principianti. Tuttavia, puoi lasciarlo lì per ora, perché funziona. Ora i compiti possono essere migliorati.

6.102.1 Miglioramenti

A causa dell'elevato numero di caselle di testo, è particolarmente utile posizionare le caselle di testo in una matrice, il che semplifica notevolmente il codice del programma. Tutto ciò di cui hai bisogno ora:

- 2 caselle di testo
- 4 pulsanti di comando

Codice sorgente: Sprint2 Form1.class^{25th}

²⁵ <http://de.wikibooks.org/wiki/Gambas%3A%20Spiele%3A%20Sprint2%20Form1.class>



Fig.84

6.103 TicTacToe

Il seguente programma mostra una versione molto semplice del gioco per 2 giocatori.

Sono necessari i seguenti controlli sul modulo

- 9 caselle di testo
- 2 aree di testo
- 4 pulsanti di comando

Codice sorgente:

- TicTacToe Form1.class^{26th}
- TicTacToe Form1.form²⁷

^{26th} <http://de.wikibooks.org/wiki/Gambas%3A%20Spiele%3A%20TicTacToe%20Form1>.

²⁷ classe <http://de.wikibooks.org/wiki/Gambas%3A%20Spiele%3A%20TicTacToe%20Form1>.

²⁷ forma



Fig.85

La struttura necessita ancora di miglioramenti. Il programma funziona con due subroutine change e win. L'intera programmazione può sicuramente essere migliorata molto. Ad esempio, potresti aggiungere un array per i campi di testo. Il programma di cui sopra può essere utilizzato come prima soluzione eseguibile e dovrebbe essere lasciato lì per motivi didattici. Le abbreviazioni significative sono difficili da capire per i principianti e dovrebbero essere fatte passo dopo passo di seguito.

6.103.1 Miglioramenti

Se ora, come notato sopra, mettiamo le caselle di testo in un array, il file Semplifica il codice del programma. Tutto ciò di cui hai bisogno ora:

- 2 caselle di testo
- 4 pulsanti di comando

Codice sorgente: TicTacToe2 Form1.class²⁸

²⁸ <http://de.wikibooks.org/wiki/Gambas%3A%20Spiele%3A%20TicTacToe2%20Form1>. classe



Fig.86

6.104 collegamenti

- <http://gamebas.tuxfamily.org/download/> (non lavora più)

Convertitore 6.105 (testo in codice HTML)

Questo programma converte il testo in codice di caratteri HTML.

Hai bisogno:

- 1 forma
- 1 pulsante
- 1 etichetta di testo
- 2 caselle di testo

Risultato come stringa

```
Pulsante PUBLIC SUB1_Click ()
    i AS intero
    Risultato = ""
```

```
"Conversione dal set di caratteri" utf-8 "al set di caratteri" ISO-8859-1 "Textbox2.Text
= Conv $ (Textbox1.Text," utf-8 "," ISO-8859-1 ")
```

```
FOR i = 1 TO Len (Textbox2.Text)
  "Len" restituisce la lunghezza della stringa passata

  Risultato = Risultato & Chr $ (38) & "#" & Asc (Textbox2.Text, i) &
  ","
  "'Asc" restituisce il codice ASCII del carattere nella posizione specificata della stringa

  'Chr $ (38) restituisce il carattere &, usato qui solo per evitare possibili' problemi di
  visualizzazione nel browser

IL PROSSIMO

  Textbox2.Text = Risultato
FINE

PUBLIC SUB TextBox1_Enter ()
  Button1_Click
FINE
```

Ecco come appare:



Fig.87 Testo in codice HTML

6.106 Youtube Downloader

Ti consente di scaricare i video di YouTube utilizzando un collegamento YouTube. I componenti gb.net, gb.net.curl, gb.desktop, gb.gui, gb.form sono necessari.

Ecco la forma

```
# File di modulo Gambas 2.0

{Form Form MoveScaled
(0,0,38.5714,8.5714) Text =
("YoutubeDownloader") Border =
Window.Fixed
{TextBox1 TextBox
MoveScaled (1,1,21,3)
ToolTip = ("Qui va il collegamento")
Testo = ("")
}
{PictureBox1 PictureBox MoveScaled
(23,1,14.1429,7.1429)
}
Pulsante {Button1
```

```

        MoveScaled (1,5,21,3)
        ToolTip = ("fai clic qui se sei pronto per il download")
        Text = ("Download")
    }
}

```

Ecco il codice sorgente

```

"File di classe Gambas
CONST PICTURENAME AS String = "youtube.jpg"
CONST YOUTUBE AS String = "http://www.youtube.com"

PUBLIC SUB Form_Open ()
    IMMAGINE DIM COME
        Immagine

    pic = Picture.Load (Application.Path & "/" & PICTURENAME)
    immagine Ridimensiona (100, 50)
    PictureBox1.Picture = pic
    Application.MainWindow.Icon = pic
FINE

Pulsante PUBLIC SUB1_Click ()
    getYoutubeFile (TextBox1.Text)
FINE

PUBLIC SUB PictureBox1_MouseDown ()
    Desktop.Open (YOUTUBE)
FINE

' Apre il browser standard con il video di
' Youtubeurl viene passato come parametro
PUBLIC SUB getYoutubeFile (url AS String)
    Contenuto DIM AS String

    url = parseYoutube (url)
    IF url = NULL THEN
        Message.Error ("Errore: problemi con l'analisi della pagina!")
        RITORNO
    FINISCI SE

    Desktop.Open (url)
FINE

' ha analizzato una pagina di YouTube per trovare la posizione effettiva del video

FUNZIONE PUBBLICA parseYoutube (url AS String) AS String
    Contenuto DIM AS String
    DIM tmp AS stringa
    DIM tmp2 AS stringa
    DIM tval AS stringa
    ID DIM AS String

    'carica pagina
    contenuto = getFile (url)
    IF contenuto = NULL THEN
        RITORNO ZERO
    FINISCI SE

    'tval
    tmp = String.InStr (contenuto, "\" t \": \" ", 0) + String.Len (" \\" t \": "
    \ "")
    tmp2 = String.InStr (contenuto, "\", ", tmp)
    tval = String.Mid (contenuto, tmp, tmp2 - tmp)

    'id
    tmp = String.InStr (url, "v=", 0) + String.Len ("v =")

```

```
id = String.Mid (url, tmp)

RETURN "http://www.youtube.com/get_video?video_id=" & id & "& t =" & tval

FINE

' Scarica una pagina e la restituisce come stringa FUNZIONE PUBBLICA
getFile (url AS String) AS String
    DIM h AS HttpClient DIM
    buffer AS String

    buffer = NULL
    h = NUOVO HttpClient AS "h"
    h.URL = url
    h.Get ()

    DO WHILE h.Status <> 0
        ATTENDERE 0.01
    CICLO CONTINUO

    SE h. Stato >= 0 ALLORA
        SE Lof (h) ALLORA
            LEGGI #h, buffer, Lof (h)
        FINISCI SE
    FINISCI SE

    Buffer di RITORNO
FINE
```

6.107 Tutorial sull'aggiunta (conteggio in su)

Con questo piccolo tutorial puoi imparare ad aggiungere

Hai bisogno di un modulo grafico per il programma e

- 6 caselle di testo
- 3 pulsanti di comando

Il programma genera 2 numeri casuali a e b. Questi vengono sommati insieme al risultato c = a + b. L'utente inserisce il risultato d in una casella di testo (qui: casella di testo5). Ciò viene verificato dal comando TESTEN con il risultato vero c.

È necessario etichettare i pulsanti di comando come segue:

- Button1 = fine
- Button2 = Nuova attività
- Button3 = test

Modificare di conseguenza il testo dei pulsanti di comando nella finestra delle proprietà.

Il programma confronta d con il risultato corretto c in una valutazione di subroutine e visualizza il risultato in una delle caselle di testo. Il numero totale di attività, il numero di attività corrette e il numero di attività sbagliate vengono visualizzati in 3 caselle di testo aggiuntive.

Il codice per questo:

```

' File classe Gambas
PUBLIC a AS Intero PUBLIC
b AS Intero PUBLIC c AS
Intero PUBLIC SUB
Form_Open () z AS Intero

'numero di attività r AS
intero
'numero di attività corrette per AS
intero
'Numero di domande sbagliate z = 0

r = 0 f = 0 TextBox1.Text = ""
TextBox2.Text = Str (z)
TextBox3.Text = Str (r)
TextBox4.Text = Str (f)
Textbox5.Text = ""
Textbox6.Text = ""

FINE
PUBLIC SUB Button1_Click ()
ME.Close
FINE
PUBLIC SUB Button2_Click () z
AS Integer
r AS Intero f AS
Intero d AS
Intero
z = Val (textbox2.text) z = z +
1 Randomize ()

    ' Rem Counter per il numero di task a = Int (Rnd (1)
    * 10)
    'Rem ottiene un numero casuale
    b = Int (Rnd (1) * 10)
    'Rem ottiene il secondo numero
    casuale c = a + b
    STAMPA c
    textbox1.Text = "Aggiungi" & Str $ (a) & "e" & Str $ (b) &
controlla il risultato "
    textbox2.Text = Str (z)
FINE
Pulsante PUBLIC SUB 3_Click ()
d AS intero
d = Val (Textbox5.text)
valutazione
CATTURARE
IF ERROR THEN message.Info ("Inserisci il tuo risultato nella casella di testo
accanto al pulsante TESTING!") END

Classificazione SUB ()
r AS intero
d AS intero
f AS intero
r = Val (textbox3.text)
d = Val (textbox5.text)
f = Val (textbox4.text)
IF d = c THEN textbox6.Text = "Correct:" & Str $ (a) & "+"
&
Str $ (b) & "=" & Str (c)
SE d = c ALLORA r = r + 1
textbox3.Text = Str (r)
SE d <> c ALLORA textbox6.Text = "Sbagliato !! sarebbe stato corretto"
SE d <> c ALLORA textbox6.Text = textbox6.Text & Str $ (a) & "+" &
Str $ (b) & "=" & Str $ (c)
SE d <> c ALLORA f = f + 1

```

```
textbox4.Text = Str (f)
FINE
```

Hai avviato il programma? Quello che manca qui è un'immagine panoramica della forma. Questo verrà consegnato più tardi. Se si dichiarano pubbliche le variabili d, r, f, z prima delle subroutine, è possibile semplificare notevolmente il codice.

Ora prova a variare il programma:

- Crea un nuovo programma con numeri casuali fino a 100.
- Crea un nuovo programma con la moltiplicazione come risultato.

6.108 Esercitazione Sottrai (Sottrai)

Puoi imparare a sottrarre (detrarre) con questo piccolo tutorial

Hai bisogno di un modulo grafico per il programma e

- 3 caselle di testo
- 3 pulsanti di comando

Il programma genera 2 numeri casuali a e c. Questi vengono sottratti l'uno dall'altro con il risultato $c = a - b$. L'utente inserisce il risultato d in una casella di testo (qui: casella di testo5). Ciò viene verificato dal comando TESTEN con il risultato vero c.

È necessario etichettare i pulsanti di comando come segue:

- Button1 = fine
- Button2 = Nuova attività
- Button3 = test

Modificare di conseguenza il testo dei pulsanti di comando nella finestra delle proprietà.

Il programma confronta d con il risultato corretto c in una valutazione di subroutine e visualizza il risultato in una delle caselle di testo.

A differenza del tutorial sull'aggiunta, non viene eseguito il conteggio delle attività.

Il codice per questo:

```
' File di classe Gambas
PUBLIC a AS Intero PUBLIC
b AS Intero PUBLIC c AS
Intero PUBLIC d AS Intero
PUBLIC SUB Form_Open ()
    Textbox1.Text = ""
    Textbox5.Text = ""
    Textbox6.Text = ""

    FINE
    PUBLIC SUB Button1_Click ()
        ME.Close
    FINE
    PUBLIC SUB Button2_Click ()
        Rendi casuale ()
        a = Int (Rnd (1) * 10)
```

```

'Rem ottenere un numero casuale
c = Int (Rnd (1) * 10)
'Rem ottiene il secondo numero casuale
b = a - c
    textbox1.Text = "Sotrai" & Str $ (a) & "meno" & Str $ (b) &
    " e controlla il risultato "
FINE
PUBLIC SUB Button3_Click () d =
    Val (Textbox5.text) valutazione

CATTURARE
    IF ERROR THEN message.Info ("Inserisci il tuo risultato nella casella di testo
accanto al pulsante TESTING!")
FINE
Classificazione SUB ()
    IF d = c THEN textbox6.Text = "Corretto:" & Str $ (a) & "-" & Str $ (b) & "=" & Str (c)

    IF d <> c THEN textbox6.Text = "Sbagliato !! Sarebbe stato corretto" IF d <> c THEN
        textbox6.Text = textbox6.Text & Str $ (a) & " - " & Str $ (b) & " = " & Str $ (c)

FINE

```

Perché il numero c è stato definito come un numero casuale e non b? Come puoi cambiare il programma in modo che nessun numero negativo possa uscire per b? Qui manca un ciclo, quindi c <a. Allora anche b sarà sempre un numero positivo.

6.109 Tutorial sulla moltiplicazione

Con questo piccolo tutorial puoi imparare a moltiplicare (prendere tempo)

Hai bisogno di un modulo grafico per il programma e

- 6 caselle di testo
- 3 pulsanti di comando

Il programma genera 2 numeri casuali a e b. Questi vengono moltiplicati per il risultato c = a * b. Nel linguaggio del computer, l'asterisco è il simbolo della moltiplicazione. L'utente inserisce il risultato d in una casella di testo (qui: casella di testo5). Questo viene confrontato con il vero risultato c dal comando TESTEN.

È necessario etichettare i pulsanti di comando come segue:

- Button1 = fine
- Button2 = Nuova attività
- Button3 = test

Modificare di conseguenza il testo dei pulsanti di comando nella finestra delle proprietà.

Il programma confronta d con il risultato corretto c in una valutazione di subroutine e visualizza il risultato in una delle caselle di testo. Il numero totale di attività, il numero di attività corrette e il numero di attività sbagliate vengono visualizzati in 3 caselle di testo aggiuntive.

Il programma può essere utilizzato anche senza mouse. Pertanto sono state aggiunte le subroutine Keypress. Il tasto backspace nella casella di testo per l'immissione dei risultati funziona un po 'confuso. Pertanto qui è stato programmato un diversivo.

Il codice per questo:

```
' File classe Gambas
PUBLIC a AS Intero PUBLIC
b AS Intero PUBLIC c AS
Intero PUBLIC r AS Intero
PUBLIC d AS Intero PUBLIC
f AS Intero PUBLIC z AS
Intero PUBLIC SUB
Form_Open () z = 0

r = 0 f = 0 TextBox1.Text = ""
TextBox2.Text = Str (z)
TextBox3.Text = Str (r)
TextBox4.Text = Str (f)
TextBox5.Text = ""
TextBox6.Text = "" Button2.
Valore predefinito = TRUE
END

PUBLIC SUB Button1_Click ()
    ME.Close
FINE
PUBLIC SUB Button2_Click ()
    textBox5.Text = ""
    textBox5.SetFocus
    z = Val (textBox2.text) z = z +
    1
    Rendi casuale ()
        a = Int (Rnd (1) * 10) 'Rem
        ottiene un numero casuale b =
        Int (Rnd (1) * 10)
        'Rem ottiene il secondo numero
        casuale c = a * b
        'STAMPA c
        textBox1.Text = "Moltiplica" & Str $ (a) & "e" & Str $ (b) & "e controlla il risultato"
        textBox2.Text = Str (z)
FINE
PUBLIC SUB Button3_Click () d
    AS Integer
    d = Val (TextBox5.text)
    Button2.Default = TRUE
    valutazione
    CATTURARE
    IF ERROR THEN message.Info ("Inserisci il tuo risultato nella casella di testo
accanto al pulsante TESTING!")
FINE
Classificazione SUB ()
r = Val (textBox3.text) d = Val
(textBox5.text) f = Val
(textBox4.text)
IF d = c THEN textBox6.Text = "Correct:" & Str $ (a) & "mal" & Str $ (b) & "=" & Str (c)

SE d = c ALLORA r = r + 1
casella di testo 3 Testo = Str
(r)
IF d <> c THEN textBox6.Text = "Sbagliato !! Sarebbe stato corretto" IF d <> c THEN
textBox6.Text = textBox6.Text & Str $ (a) & "mal" & Str $ (b) & "=" & Str $ (c)

SE d <> c ALLORA f = f + 1
casella di testo 4 Testo = Str (f)
FINE
PUBLIC SUB TextBox5_KeyPress ()
```

```

t AS stringa
I AS intero
IF key.Code> = 48 AND key.Code <= 57 THEN
    Textbox1.text = Textbox1.text
ELSE IF key.Code = key.Return THEN
    Button3.setfocus
    Button3.Default = TRUE
ELSE IF key.Code = key.Backspace THEN
    t = textbox1.Text
    I = Len (t)
    t = Sinistra $ (t, I)
    textbox1.Text = t
    'funzione alquanto assurda del backspace
ELSE IF key.Code = key.Delete THEN
    Textbox1.Text = ""
ALTRO
STOP EVENTO
FINISCI SE
FINE
PUBLIC SUB Button3_KeyPress ()
    IF key.Code = key.Return THEN Button2.setfocus
FINE

```

6.110 Tutorial per i caratteri di confronto > <ed è lo stesso

Con questo piccolo tutorial puoi esercitarti sul significato dei segni di confronto

- <minore di
- > maggiore di
- = è lo stesso

Hai bisogno di un modulo grafico per il programma e

- 5 caselle di testo
- 5 pulsanti di comando

La forma è stata chiamata F. Puoi anche mantenere il nome Form1.

Il programma genera 2 numeri casuali. Questi vengono confrontati e devi scegliere il simbolo corretto> o <o =. È necessario etichettare i pulsanti di comando come segue:

- Button1 = fine
- Button2 = Nuova attività
- Button3 =>
- Button4 = "="
- Button5 = <

Modificare di conseguenza il testo dei pulsanti di comando nella finestra delle proprietà.

Il programma confronta l'immissione tramite i tasti>, <e = con il risultato corretto in una valutazione del sottoprogramma e visualizza il risultato in una delle caselle di testo. Questa casella di testo del risultato è stata chiamata valutazione. Devi cambiare il nome della casella di testo di conseguenza con F4 nelle proprietà.

Il codice per questo:

```
' File classe Gambas
PUBLIC a AS Intero PUBLIC
b AS Intero PUBLIC c AS
Intero PUBLIC d AS Intero
PUBLIC z AS Intero PUBLIC
r AS Intero PUBLIC f AS
Intero PUBLIC SUB
Form_Open () z = 0

r = 0 f
= 0
TextBox2.Text = Str (z)
TextBox4.Text = Str (f)
TextBox3.Text = Str (r)
FINE
PUBLIC SUB Button1_Click ()
ME.Close
FINE
PUBLIC SUB Button2_Click () z =
z + 1
' Contatore per il numero di attività
Rendi casuale ()
a = Int (Rnd (1) * 10) 'Rem
ottiene un numero casuale b =
Int (Rnd (1) * 10)
'Rem ottiene il secondo numero
casuale SE a> b ALLORA c = 62
' Rem Ascii codice per maggiore di IF a
= b THEN c = 61
SE a <b ALLORA c = 60
textbox1.Text = "Confronta" & Str $ (a) & "e" & Str $ (b) &
e inserisci il carattere corretto>, <o =
textbox2.Text = Str (z)
FINE
Pulsante PUBLIC SUB 3_Click ()
d = 62
'Codice Ascii per>
valutazione
FINE
Pulsante PUBLIC SUB 5_Click ()
d = 60
valutazione
FINE
Pulsante PUBLIC SUB 4_Click ()
d = 61
valutazione
FINE
Classificazione SUB ()
IF d = c THEN valut.Text = "Correct:" & Str $ (a) & "" & Chr $ (c)
& "" & Str $ (b)
SE d = c ALLORA r = r + 1
textbox3.Text = Str (r)
IF d <> c THEN assess.Text = "Wrong !! Correct would be"
IF d <> c THEN rated text = rated text & Str $ (a) & "" & Chr $ (c) & "" & Str $ (b)

SE d <> c ALLORA f = f + 1
textbox4.Text = Str (f)
FINE
```

6.111 L'istruttore di matematica di Harry

ATTENZIONE: programma ancora in corso !!

Con questo programma si dovrebbe essere in grado di addestrare i tipi base di aritmetica.

6.111.1 Forma e controlli

Hai bisogno di una nuova forma e

- 2 caselle di testo
- 2 campi etichetta di testo
- 2 pulsanti di comando

6.111.2 Codice

```
' Classe Gambas
numero file1 AS numero
intero2 AS intero risultato
AS intero risposta AS
intero
PUBLIC SUB StartAdditions_Click ()
    Rendi casuale ()
    Numero1 = int (10*Rnd ()) + 1 numero2 =
    Int (10*Rnd ()) + 1 risultato = numero1 +
    numero2 answer = answerTextBox.Text
    OutputRechnungLabel.Text = Number1 & "+" & Number2 & "=" 'Visualizzazione della
fattura

    TextLabelFeedback.Text = Risultato 'Output solo a scopo di controllo
    ResponseTextBox.SetFocus
FINE
PUBLIC SUB OkButton_Click () IF answer =
    risultato THEN TextLabelFeedback.Text =
    "Corretto"
    ALTRO
    TextLabelFeedback.Text = "False"
    ENDIF
END
RISPOSTA PUBLIC SUB TextBox_KeyPress ()
    IF key.Code = key.Enter THEN
        TextLabelFeedback.Text = Risultato 'Output solo a scopo di controllo
    ENDIF
END
```

6.112 Scarica un file

Per scaricare un file si usa Http, per Gambas c'è di conseguenza la classe HttpClient. Ciò consente di richiedere siti Web utilizzando i due metodi tipici Post e Get. Di seguito è riportato un esempio che scarica un sito Web e restituisce il codice sorgente della pagina (richiede l'integrazione dei componenti²⁹ gb.net e gb.net.curl):

29 Capitolo 12.18 a pagina 424

```
PUBLIC SUB Main ()  
    Contenuto DIM AS String  
    contenuto = getFile ("http://gambas.sourceforge.net/")  
    IF contenuto = NULL THEN  
        PRINT "Errore durante il download!"  
    ALTRO  
        PRINT contenuto  
    FINISCI SE  
FINE  
  
' Scarica una pagina e la restituisce come stringa FUNZIONE PUBBLICA  
getFile (url AS String) AS String  
DIM h AS HttpClient DIM  
buffer AS String  
  
buffer = NULL  
h = NUOVO HttpClient AS "h"  
h.URL = url  
h.Get () 'Usa il metodo Get per richiedere il sito web  
  
DO WHILE h.Status <> 0  
    ATTENDERE 0.01  
CICLO CONTINUO  
  
' Se la risposta è OK, continuare SE h.Status> =  
0 THEN  
    ' Se il file contiene qualcosa ...  
    SE Lof (h) ALLORA  
        ' scaricalo -> nel buffer delle stringhe LEGGI #h,  
        buffer, Lof (h)  
    FINISCI SE  
FINISCI SE  
  
Buffer di RITORNO  
FINE
```

6.113 Calcolatrice estesa

modulo

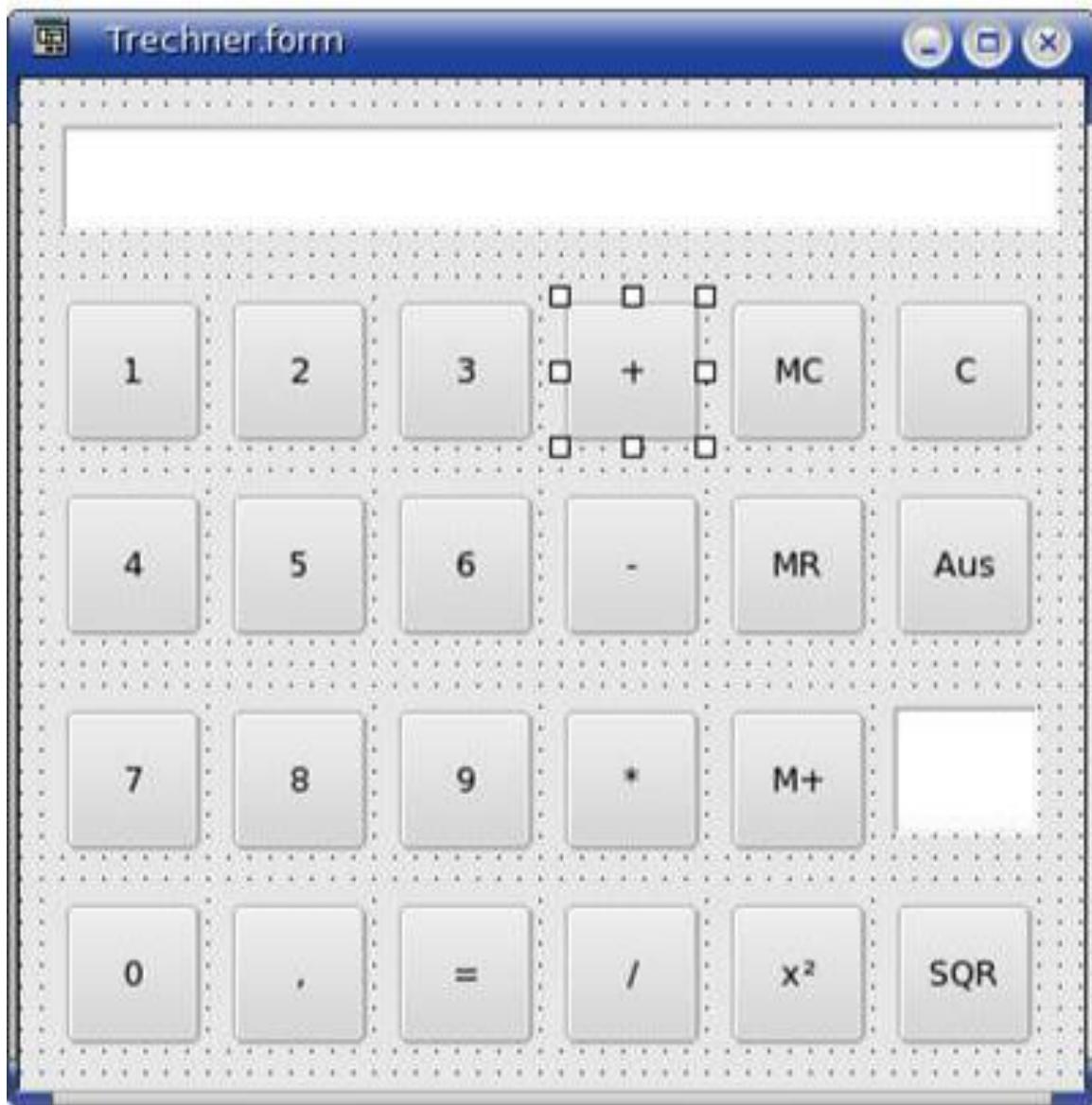


Fig. 88

```
' File classe Gambas
RISULTATO PUBBLICO AS
Variant Memoria PUBBLICA AS
Variante PUBLIC rechz AS Byte
```

```
Pulsante PUBLIC SUB1_Click ()
    Textbox1.Text = textbox1.Text & "1"
FINE
PUBLIC SUB Button2_Click ()
    Textbox1.Text = textbox1.Text & "2"
FINE
Pulsante PUBLIC SUB 3_Click ()
    Textbox1.Text = textbox1.Text & "3"
FINE
Pulsante PUBLIC SUB 7_Click ()
```

```
Textbox1.Text = textbox1.Text & "4"
FINE
Pulsante PUBLIC SUB 8_Click ()
    Textbox1.Text = textbox1.Text & "5"
FINE
Tasto PUBLIC SUB 9_Click ()
    Textbox1.Text = textbox1.Text & "6"
FINE
Tasto PUBLIC SUB 13_Click ()
    Textbox1.Text = textbox1.Text & "7"
FINE
Pulsante PUBLIC SUB 14_Click ()
    Textbox1.Text = textbox1.Text & "8"
FINE
Pulsante PUBLIC SUB15_Click ()
    Textbox1.Text = textbox1.Text & "9"
FINE
Pulsante PUBLIC SUB18_Click ()
    Textbox1.Text = textbox1.Text & "0"
FINE
```

```
Pulsante PUBLIC SUB 19_Click ()
    Textbox1.Text = textbox1.Text & ","
FINE
```

```
Pulsante PUBLIC SUB 4_Click ()
    rechz = 1
    risultato = Val (Textbox1.Text)
    Textbox1.Text = ""
FINE
```

```
Pulsante PUBLIC SUB 10_Click ()
    rechz = 2
    risultato = Val (Textbox1.Text)
    Textbox1.Text = ""
FINE
```

```
Pulsante PUBLIC SUB 16_Click ()
    rechz = 3
    risultato = Val (Textbox1.Text)
    Textbox1.Text = ""
FINE
```

```
Pulsante SUB PUBBLICO 21_Clic ()
    rechz = 4
    risultato = Val (Textbox1.Text)
    Textbox1.Text = ""
FINE
```

```
Pulsante PUBLIC SUB22_Click ()
    rechz = 5
    risultato = Val (Textbox1.Text)
    Textbox1.Text = ""
FINE
```

```
Pulsante PUBLIC SUB20_Click ()
SE rechz = 1 ALLORA
    risultato = risultato + Val (Textbox1.Text)
    textbox1.Text = risultato
ALTRIMENTI SE rechz = 2 ALLORA
```

```
risultato = risultato - Val (Textbox1.Text)
textbox1.Text = risultato
ALTRIMENTI SE rechz = 3 ALLORA
    risultato = risultato * Val (Textbox1.Text)
    textbox1.Text = risultato
ALTRIMENTI SE rechz = 4 ALLORA
    risultato = risultato / Val (Textbox1.Text)
    textbox1.Text = risultato
ALTRIMENTI SE rechz = 5 ALLORA
    risultato = risultato ^ Val (Textbox1.Text)
    textbox1.Text = risultato
CATTURARE
message.Error ("Per favore, non dividere per 0!")
FINISCI SE
FINE
```

```
Pulsante PUBLIC SUB 12_Click ()
ME.Chiudi
FINE
```

```
Pulsante PUBLIC SUB 5_Click ()
memory = ""
textbox2.BackColor = & HD9ECEB &
FINE
```

```
Pulsante PUBLIC SUB 11_Click ()
textbox1.Text = salva
FINE
```

```
Pulsante PUBLIC SUB 6_Click ()
rechz = 0
risultato = ""
textbox1.Text = ""
FINE
```

6.114 Calcolo del canone di locazione

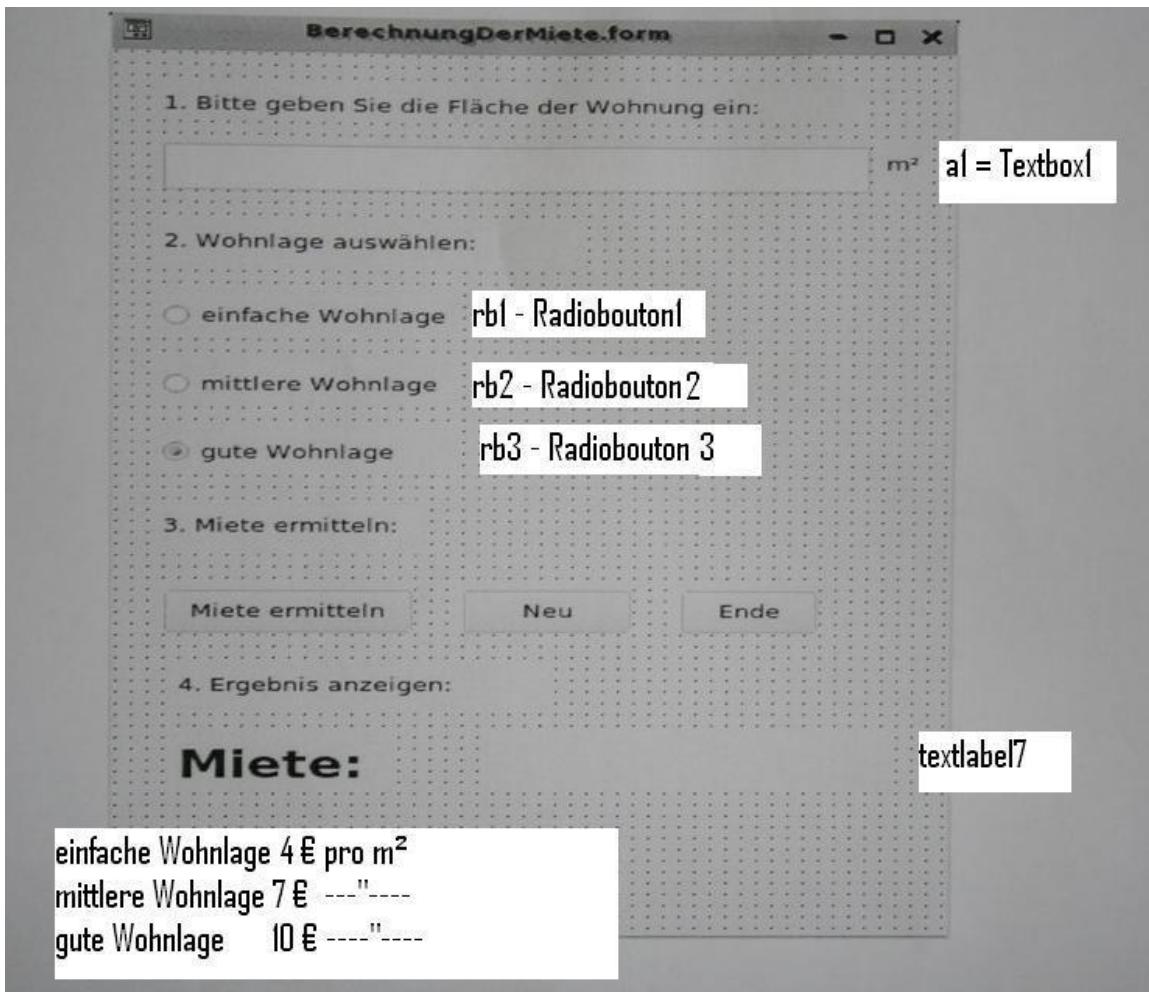


Fig.89

```
"File di classe Gambas
Pulsante PUBLIC SUB1_Click ()
DIM a1 AS Integer
a1 = Val (textbox1.Text)
SE rb1.Value = TRUE THEN
    a1 = a1*4 °
ALTRIMENTI SE rb2.Value = TRUE THEN
    a1 = a1*7 °
ALTRIMENTI SE rb3.Value = TRUE THEN
    a1 = a1*10
FINISCI SE
textlabel7.Text = a1
FINE
Pulsante PUBLIC SUB 3_Click ()
    ME.Chiudi
FINE
PUBLIC SUB Button2_Click ()
    Textlabel7.Text = ""
    textbox1.Text = ""
    rb3.Value = TRUE
```

```

FINE
PUBLIC SUB Form_Open ()
    ME.Caption = "Calcolo del canone di locazione" --- Intestazione della finestra
FINE

```

Convertitore Fahrenheit e Celsius

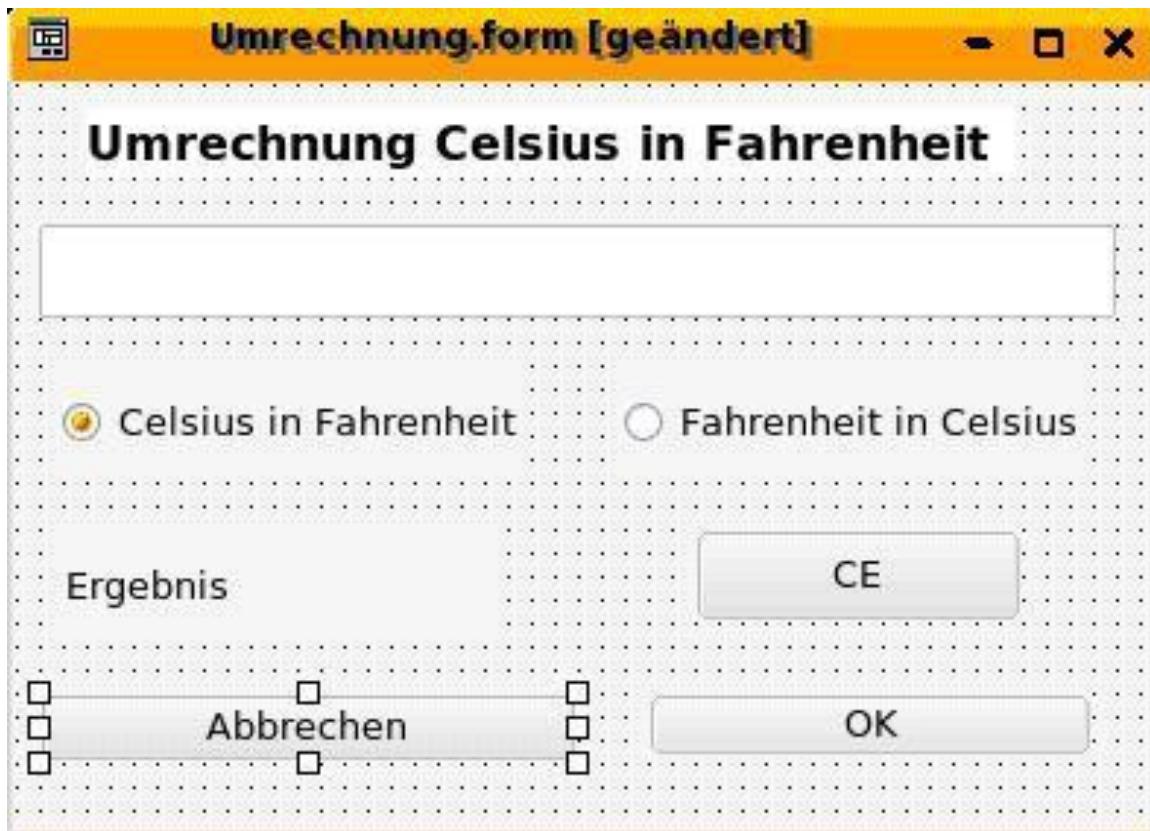


Fig.90

```

' File di classe Gambas
' Convertitore Fahrenheit e Celsius risultato
PUBBLICO AS Variant PUBLIC SUB
Button1_Click () ME.Close

FINE

```

```

PUBLIC SUB Button2_Click ()
IF RadioButton1.Value = TRUE THEN risultato = Val (textbox1.Text)*9/5 + 32

IF RadioButton2.Value = TRUE THEN risultato =
(Val (textbox1.Text) -32) / 9*5
Label1.Text = risultato
FINE

```

```

Pulsante PUBLIC SUB 3_Click ()
textbox1.Text = ""
Label1.Text = ""
FINE

```

Questo programma è un semplice calcolatore di potenza. Utilizza il simbolo matematico \wedge per l'estrapolazione. Esempio: $2^3 = 2 * 2 * 2 = 8$

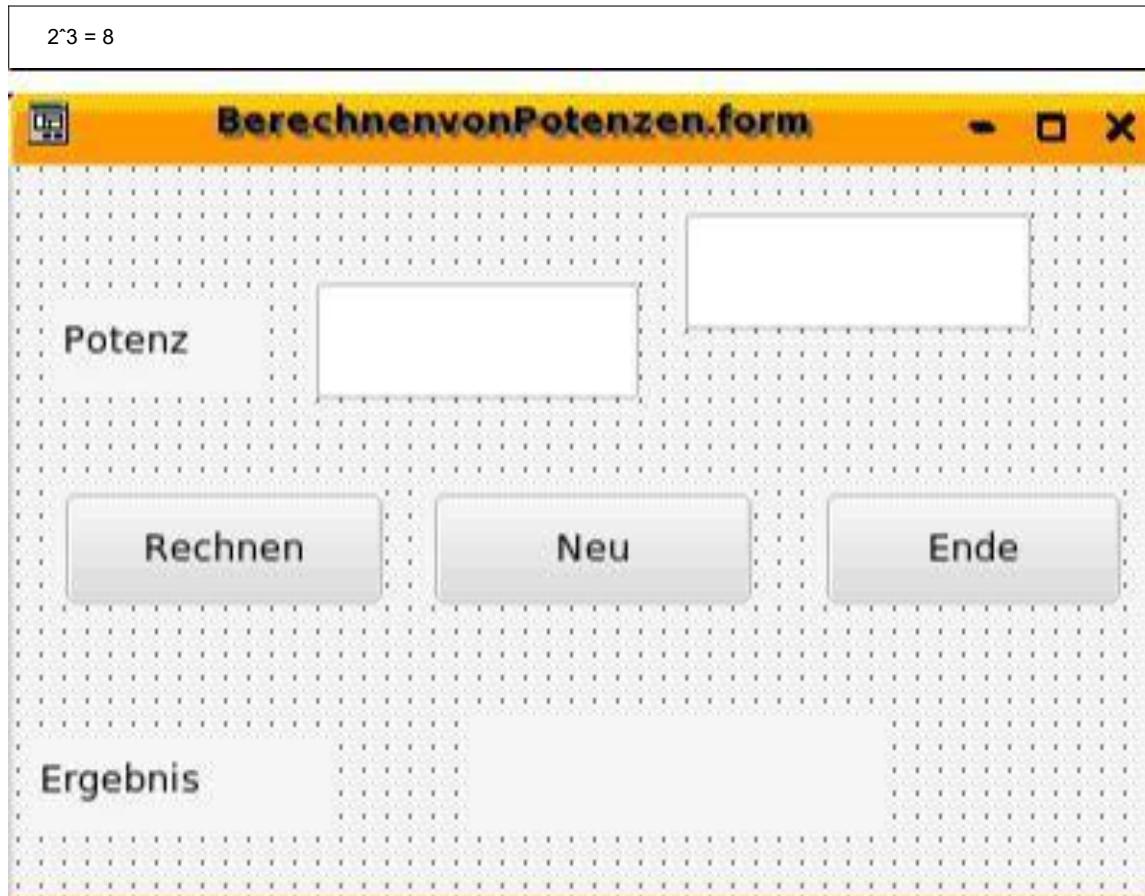


Fig.91

```
"File di classe Gambas
Pulsante PUBLIC SUB1_Click ()
DIM Pota1 AS Integer
DIM Pota2 AS Integer
DIM Pota3 AS Integer

IF textbox2.Text = 0 THEN
    label1.Text = 1
ALTRO
    Pota1 = Textbox1.Text
    Pota2 = Textbox2.Text
    Pota3 = Pota1^Pota2
    label1.Text = Pota3
FINISCI SE
FINE
PUBLIC SUB Button2_Click ()
    Textbox1.Text = ""
    Textbox2.Text = ""
    label1.Text = ""
FINE
Pulsante PUBLIC SUB 3_Click ()
    ME.Chiudi
FINE
```

Ci sono due esempi di programmazione in loop basata sul calcolo dell'interesse.

6.115 Loop semplice

```
' File classe Gambas Capitale
PUBBLICO AS Flottante Interesse
PUBBLICO AS Flottante Durata
PUBBLICA AS Intero PUBLIC SUB
Button3_Click () ME.Close

FINE
PUBLIC SUB Button2_Click ()
    Textbox1.text = "" Textbox2.text =
    "" Textbox3.text = ""

FINE
PUBLIC SUB Button1_Click () DIM
    X AS Integer capital =
    textbox1.Text interest =
    textbox2.Text duration =
    textbox3.Text
    PER x = 1 TO durata FASE 1
        capitale = capitale + (capitale*interesse / 100)
    IL PROSSIMO
    label1.Text = "Capitale finale" & Round (capitale, -2) & "Euro"
FINE
PUBLIC SUB Label1_MouseDown ()
FINE
PUBLIC SUB Form_Open ()
FINE
```

6.116 Loop esteso

```
PUBBLICO euro AS flottante
PERCENTUALE PUBBLICO AS fluttuante
Pulsante PUBLIC SUB 3_Click ()
    ME.Chiudi
FINE
PUBLIC SUB Button2_Click ()
    Textbox1.Text = ""
    Textbox2.Text = ""
FINE
Pulsante PUBLIC SUB1_Click ()
    DIM anni AS Integer
    euro = Val (Textbox1.Text)
    percentuale = Val (Textbox2.Text)
    FARE MENTRE euro <1.000.000
        euro = euro + (euro*percentuale / 100)
        Anni = anni + 1
    CICLO CONTINUO
    textlabel5.Text = "Dopo" & anni & "anni sono milionari e hanno un capitale di" & euro &
    "euro." FINE

PUBLIC SUB TextBox1_KeyPress ()
FINE
```

6.117 emogasanalisi

per favore, riferisci <http://www.madeeasy.de/7/ph.htm>

L'emogasanalisi è rapida e importante in medicina. Come puoi interpretare i valori misurati? Ciò che manca ancora è il valore del lattato e il valore dell'emoglobina. Il codice del programma

```
"File di classe Gambas
Pulsante PUBLIC SUB 7_Click ()
ME.Chiudi
FINE
PUBLIC SUB Button2_Click ()
textbox1.text = 7
textbox2.text = 30
textbox3.text = -10
textbox4.text = 90
FINE
Pulsante PUBLIC SUB 3_Click ()
textbox1.text = "7.6"
textbox2.text = 40
textbox3.text = 10
textbox4.text = 90
FINE
Pulsante PUBLIC SUB 6_Click ()
textbox1.text = "7.4"
textbox2.text = 40
textbox3.text = 0
textbox4.text = 90
FINE
Pulsante PUBLIC SUB 4_Click ()
textbox1.text = "7.4"
textbox2.text = 30
textbox3.text = 0
textbox4.text = 50
FINE
Pulsante PUBLIC SUB 5_Click ()
textbox1.text = "7.1"
textbox2.text = 65
textbox3.text = 0
textbox4.text = 50
FINE
PUBLIC SUB TextBox1_KeyPress ()
IF key.Code = key.Return THEN textbox2.setfocus
FINE
PUBLIC SUB TextBox2_KeyPress ()
IF key.Code = key.Return THEN textbox3.setfocus
FINE
PUBLIC SUB TextBox3_KeyPress ()
IF key.Code = key.Return THEN textbox4.setfocus
FINE
PUBLIC SUB TextBox4_KeyPress ()
IF key.Code = key.Return THEN button1.Default = TRUE IF key.Code
= key.Return THEN button1.setfocus END

Pulsante PUBLIC SUB1_Click ()
' befehl3.SetFocus DIM
pc AS Float DIM po AS
Float DIM be AS Float
DIM ph AS Float DIM
risultato AS String DIM b
AS String

IF textbox1.text = "" THEN textbox1.text = 0 IF
textbox2.text = "" THEN textbox2.text = 0 IF textbox3.text =
"" THEN textbox3.text = 0 IF textbox4.text = "" THEN
textbox4.text = 0 ph = Val (textbox1.text)
```

```

pc = Val (textbox2.text)
be = Val (textbox3.text)
po = Val (textbox4.text)
' STAMPA ph, pc, be, po ergeb
= metabol (ph, pc, be) b =
ossigeno (po, pc)
risultato = risultato & Chr $ (13) & Chr $ (10) & b
textarea1.text = risultato
' Risultato di STAMPA
FINE
FUNZIONE PUBBLICA metabol (ph AS Float, pc AS Float, be AS Float) AS
Corda
Rem *****
: Rem * DICHIARAZIONE 1 PH, BE, PCO2 *
DIM a1 AS String DIM
a2 AS String DIM a3
AS String DIM a4 AS
String DIM a5 AS
String DIM a6 AS
String DIM a7 AS
String DIM a8 AS
String DIM a9 AS
String DIM a10 AS
String DIM a11 AS
String DIM a12 AS
String DIM e Come
stringa

a1 = "valori non compatibili con la vita"
a2 = "Acidosi metabolica" & Chr $ (13) & Chr $ (10) & "Iperventilazione respiratoria
compensata!"
a3 = "Acidosi metabolica non compensata"
a4 = "Acidosi respiratoria non compensata"
a5 = "Acidosi respiratoria + metabolica"
a6 = "ipercapnia + alcalosi compensatoria"
a7 = "Equilibrio acido-base equilibrato"
a8 = "alcalosi metabolica"
a9 = "iperventilazione"
a10 = "Valori non plausibili"
a11 = "ipercapnia"
a12 = "acidosi metabolica"
' Rem * CONDIZIONI PH, BE, PCO2 *
SE essere> 50 ALLORA
e = a1
RITORNO e
ALTRIMENTI SE essere <-50 ALLORA
e = a1
RITORNO e
ALTRIMENTI SE ph> 7,9 ALLORA
e = a1
RITORNO e
ALTRIMENTI SE ph <6,5 ALLORA
e = a1
RITORNO e
ALTRIMENTI SE pc <1 ALLORA
e = a1
RITORNO e
ALTRIMENTI SE pc> 120 ALLORA
e = a1
RITORNO e
ALTRIMENTI SE ph <7,35 AND be> -4 AND pc <35 THEN
e = a10
RITORNO e
ALTRIMENTI SE ph <7,45 AND be <-3 AND pc <35 THEN
e = a2
RITORNO e

```

```
ALTRIMENTI SE ph <7.35 AND be <-3 AND pc> 35 THEN
e = a3
RITORNO e
ALTRIMENTI SE ph <7,35 AND be <4 AND pc> 45 THEN
e = a4
RITORNO e
ALTRIMENTI SE ph <7.35 AND be <-3 AND pc> 45 THEN
e = a5
RITORNO e
ALTRIMENTI SE ph> 7.34 AND be> 3 AND pc> 44 THEN
e = a6
RITORNO e
ALTRIMENTI SE ph <7,45 AND ph> 7,35 AND be> -5 AND be <5 AND pc> 35
E pc <45 POI
e = a7
RITORNO e
ALTRIMENTI SE ph> 7,45 ED essere> 3 ALLORA
e = a8
RITORNO e
ALTRIMENTI SE ph> 7.45 AND be <-5 THEN
e = a10
RITORNO e
ALTRIMENTI SE ph <7,35 ED essere <-5 ALLORA
e = a12
RITORNO e
ALTRIMENTI SE pc <36 ALLORA
e = a9
RITORNO e
ALTRIMENTI SE pc> 44 ALLORA
e = a11
RITORNO e
FINISCI SE
FINE
FUNZIONE PUBBLICA ossigeno (po AS Float, pc AS Float) AS String DIM b AS
String
Rem ****
' Rem * PO2 *
' Rem ****
SE po> 700 ALLORA
b = "Valori non compatibili con la vita" RITORNO b

ALTRIMENTI SE pc> 300 ALLORA
b = "Valori non compatibili con la vita" RITORNO b

ALTRIMENTI SE po <1 E pc <1 ALLORA
b = "Valori non compatibili con la vita" RITORNO b

ALTRIMENTI SE po <1 ALLORA
b = "Valori non compatibili con la vita" RITORNO b

ALTRIMENTI SE pc <1 ALLORA
b = "Valori non compatibili con la vita" RITORNO b

ALTRIMENTI SE po> 100 ALLORA
b = "iperventilazione tramite ventilazione o apporto di ossigeno ??" RITORNO b

ALTRIMENTI SE po <55 E pc <35 ALLORA
b = "Ipossia e iperventilazione embolia polmonare?" RITORNO b

ALTRIMENTI SE po <55 ALLORA
b = "L'ipossia richiede ventilazione?"
RITORNO b
ALTRIMENTI SE po <75
ALLORA b = "Ipossia"
RITORNO b

ALTRIMENTI SE pc> 50 ALLORA
```

```

b = "ipercapnia"
RITORNO b
FINISCI SE
FINE
PUBLIC SUB ToggleButton1_Click ()
DIM Help AS String
IF TextArea2.Visible = FALSE THEN
    ToggleButton1.Text = "Elimina informazioni"
    Help = "Autore:" & Chr (13) & Chr (10)
    Help = "Rho = Reiner Hoffmann" e Chr (13) e Chr (10)
    Help = Help & "www.madeeasy.de" & Chr (13) & Chr (10)
    TextArea2.Visible = TRUE
    TextArea2.text = Aiuto
    ALTRO
    TextArea2.Visible = FALSE
    ToggleButton1.Text = "Visualizza informazioni"
    FINISCI SE
    FINE
PUBLIC SUB Form_Open ()
    textbox1.setfocus
FINE

```

6.118 referto ECG

In medicina devi mettere insieme gli stessi moduli di testo più e più volte per formare un risultato. Questo programma è stato scritto per accelerare questo processo e per ottenere un risultato chiaramente leggibile.

Hai bisogno:

- Una classe come classe iniziale con il nome t
- 4 forme

Codice per la classe t:

```

' File di classe Gambas STATIC
PUBLIC t AS String STATIC
PUBLIC SUB Main () form1.show

FINE

```

Codice per il modulo1:

```

PUBLIC SUB Form_Open ()
'Appunti. Cancella
' Elimina      appunti.
Textarea2.Visible = FALSE
Textarea2.Text = ""
ME.Text = "ECG - diagnosi"
' Quando esci, il testo è negli appunti, purtroppo non textarea1.Text = ""
textarea1.Text = dd
FINE
PUBLIC SUB Button1_Click ()
Clipboard.Copy (Textarea1.Text)
ME.Close

```

```
FINE
PUBLIC SUB Button2_Click ()
textarea1.text = dd
Appunti.Copia (Textarea1.Text)
FINE
Pulsante PUBLIC SUB24_Click ()
t.t = tt & "ECG a riposo altrimenti normale"
textarea1.Text = tt Appunti.Copia (tt)

FINE
Pulsante PUBLIC SUB 3_Click ()
tt = tt & "ritmo sinusale" & Chr (10)
FINE
Pulsante PUBLIC SUB 10_Click ()
tt = tt & "frequenza cardiaca 50 - 60 / min" & Chr (10)
FINE
Pulsante PUBLIC SUB 11_Click ()
tt = tt & "frequenza cardiaca 40 - 50 / min" & Chr (10)
FINE
Pulsante PUBLIC SUB 12_Click ()
t.t = tt & "=> tachiaritmia" & Chr (10)
FINE
Tasto PUBLIC SUB 13_Click ()
t.t = tt & "=> Bradiaritmia" & Chr (10)
FINE
Pulsante PUBLIC SUB 14_Click ()
t.t = tt & "Tipo di postura normale, ECG normale a riposo" & Chr (10)
textarea1.Text = tt
Appunti.Copia (tt)
FINE
Pulsante PUBLIC SUB15_Click ()
tt = tt & "=> tachicardia> 120 / min" & Chr (10)
FINE
Pulsante PUBLIC SUB 16_Click ()
t.t = tt & "=> Bradicardia <30 / min !!" & Chr (10)
FINE
Pulsante PUBLIC SUB 17_Clic ()
t.t = tt & "=> Bradicardia <40 / min" & Chr (10)
FINE
Pulsante PUBLIC SUB 4_Click ()
t.t = tt & "frequenza cardiaca 60 - 80 / min" & Chr (10)
FINE
Pulsante PUBLIC SUB 5_Click ()
t.t = tt & "frequenza cardiaca 80 - 100 / min" & Chr (10)
FINE
Pulsante PUBLIC SUB 6_Click ()
t.t = tt & "=> frequenza cardiaca 100 - 120 / min !!" & Chr (10)
FINE
Pulsante PUBLIC SUB 7_Click ()
t.t = tt & "Aritmia assoluta" & Chr (10)
FINE
Pulsante PUBLIC SUB 8_Click ()
t.t = tt & "fibrillazione atriale" & Chr (10)
FINE
Tasto PUBLIC SUB 9_Click ()
t.t = tt & "flutter atriale" & Chr (10)
FINE
Pulsante PUBLIC SUB 19_Click ()
t.t = tt & "Tipo posizione sinistra, ECG normale a riposo" & Chr (10)
textarea1.Text = tt
Appunti.Copia (tt)
FINE
Pulsante PUBLIC SUB18_Click ()
t.t = tt & "Tipo di pendenza ripida, ECG normale a riposo" & Chr (10)
textarea1.Text = tt
Appunti.Copia (tt)
```

```

FINE
Pulsante PUBLIC SUB20_Click ()
    tt = tt & "==> tachicardia> 150 / min" e Chr (10)
FINE
Pulsante SUB PUBBLICO 21_Clic ()
t.t = tt & "colpi di stanza extra individuali" & Chr (10)
FINE
Pulsante PUBLIC SUB22_Click ()
tt = tt & "battiti atriali extra individuali" & Chr (10)
FINE
PUBLIC SUB Button23_Click () 'per
la forma del ritmo
FINE
PUBLIC SUB Button25_Click ()
    Textarea1.Text = ""
FINE
PUBLIC SUB Button27_Click () tt =
"""
FINE
PUBLIC SUB ToggleButton1_Click ()
    DIM Help AS String
    IF TextArea2.Visible = FALSE          POI
        ToggleButton1.Text = "Elimina informazioni"
        Help = "Rho = Reiner Hoffmann" e Chr (13) e Chr (10)
        Help = Help & "www.madeeasy.de" & Chr (13) & Chr (10)
        Aiuto = Aiuto & " rho54@gmx.de "
        TextArea2.Visible = TRUE
        TextArea2.text = Aiuto
        ALTRO
        TextArea2.Visible = FALSE
        ToggleButton1.Text = "Visualizza informazioni"
    FINISCI SE
    FINE
Pulsante PUBLIC SUB 28_Click ()
    Textarea1.text = Textarea1.text & "Questo programma è stato creato con Gambas" & Chr
(10)

    Textarea1.text = Textarea1.text & "Viene utilizzato per la diagnosi dell'ECG." & Chr (10)

    Textarea1.text = Textarea1.text & "Le abbreviazioni cliccate sono incollate
insieme." & Chr (10)
    Textarea1.text = Textarea1.text & "L'intero testo è memorizzato negli appunti."
& Chr (10)
    Textarea1.text = Textarea1.text & "Con i tasti CTRL e V il testo" & Chr (10)

    Textarea1.text = Textarea1.text & "può essere inserito ovunque in qualsiasi altro
programma." FINE

Pulsante PUBLIC SUB 26_Click ()
Form1.Close
Form2.Show
FINE

```

6.119 Esame radiografico del torace

6.120 Diagnosi ECG a lungo termine

6.121 Rapporto di endoscopia

6.122 Diagnosi ecografica

6.123 Risultati di ammissione

6.124 Breve lettera di dimissione del medico

6.125 Diagnosi TC

6.126 Tempo QT

Conversione frequenza cardiaca -

6.127 distanza RR

6.128 Gambas e SVG

Gambas dovrebbe avere il supporto per le icone e la grafica SVG. Finora non riesco a trovare nulla.

Secondo il changelog di Gambas, il controllo dell'immagine può caricare file SVG.

La classe Picture può ora gestire i file SVG! La sua interfaccia ha acquisito una proprietà di tipo e si comporta in modo leggermente diverso.

Puoi verificarlo con Picturebox. Non funziona, solo per i seguenti formati di immagine: JPEG, PNG, BMP, GIF e XPM. Vedi PictureLoad (carica immagine) Gambas: `_ - Bilder # Laden_Sie_ein_Bild_mit_Dialog_in_eine_Picturebox`^{30th}

Finora, un file SVG è stato caricato solo in un'area di testo come file di testo.

Forse devi aggiungere i componenti XML di Gambas:

- `gb.xml.libxml` - Strumenti XML basati su libxml
- `gb.xml.libxml.rpc` - client XML-RPC basato su libxml e libcurl
- `gb.xml.libxml.xslt` - Strumenti XSLT basati su libxslt

Pubblicherei anche la domanda qui:

- <http://sourceforge.net/mailarchive/forum.php?forum=gambas-user>

30 Capitolo 6.9 a pagina 219

6.128.1 Fonti

Scalable Vector Graphics, o SVG, è il nuovo standard grafico basato su XML del W3C che consentirà ai documenti Web di essere più piccoli, più veloci e più interattivi.

Per favore, riferisci

- http://de.wikipedia.org/wiki/Scalable_Vector_Graphics
- <http://de.wikipedia.org/wiki/Sodipodi>
- <http://www.adobe.com/svg/viewer/install/main.html>
- <http://www.oreilly.com/catalog/svgess/chapter/ch03.html>
- <http://www.w3.org/TR/SVG/intro.html>
- http://entwickler.com/itr/online_artikel/show.php3?nodeid=97&id= 325

Un certo numero di programmi di esempio vengono installati con Gambas. Puoi guardarli nel codice sorgente e imparare molto sulla programmazione. Tuttavia, questi programmi sono generalmente troppo difficili per i principianti. Questi programmi di esempio sono suddivisi in mini programmi e commentati di seguito.

Guarda anche: <http://www.madeeasy.de/7/zbgambas.htm>

6.129 piaghe

Blights è una barra lampeggiante con diverse piccole luci che lampeggiano in modo casuale.

Cosa puoi imparare dal programma?

6.129.1 Come si integra un grafico?

Un semplice esempio basato su Blights.

Si avvia un nuovo progetto Gambas blights0 e si inserisce una casella delle immagini e 2 pulsanti di comando sul modulo.

Quindi inserisci il seguente codice:

```
Pulsante PUBLIC SUB1_Click ()
    PictureBox1.Picture = Immagine ["bloff.xpm"]
FINE

PUBLIC SUB Button2_Click ()
    PictureBox1.Picture = Immagine ["blon.xpm"]
FINE
```

Se avvii il programma, non succede nulla perché mancano i 2 grafici bloff.xpm e blon.xpm. Li rubi dai guai e li copi nella directory del tuo progetto.

Avvia il tuo guscio. Il bottone con la conchiglia.

Vai alla directory del progetto sopra, ad esempio con

```
cd gambas / blights0
```

Quindi prova prima quanto segue:

```
individuare bloff.xpm
```

Se questo non funziona perché non hai installato il programma di localizzazione, inserisci invece quanto segue:

```
trova / -name bloff.xpm 2> / dev / null
```

(Dovrai quindi essere paziente.) L'output che otterrai è qualcosa del genere:

```
/opt/gambas/share/gambas/examples/Basic/Blights/bloff.xpm  
/usr/share/doc/packages/gambas/examples/Basic/Blights/bloff.xpm
```

Ora contrassegna una delle directory con il mouse e puoi copiare i file. Gli appunti ti aiuteranno in questo³¹ e incollare con il pulsante centrale del mouse o il tasto di scorrimento del mouse.

```
cp / opt / gambas / share / gambas / examples / Basic / Blights / bl* .
```

Puoi creare le tue immagini con un programma di pittura. Anche gli appunti sarebbero un trucco. È possibile caricare la visualizzazione delle macchie con Ctrl e il pulsante come screenshot negli appunti e quindi ritagliare le immagini in un programma di pittura.

6.129.2 Come si accende e si spegne un grafico?

6.130 Snake

6.131 browser web

6.132 connessione al database

6.133 Apostrofo come segno di commento

Un buon programma contiene commenti esplicativi. Questi sono molto utili per gli altri per capire un programma. I commenti vengono utilizzati quando si interpreta o si compila il

³¹ <http://de.wikipedia.org/wiki/Zwischenablage>

i grammi non vengono presi in considerazione. Quindi puoi scrivere tutto quello che vuoi nei commenti in testo normale.

Se vuoi inserire una riga in Gambas come commento, scrivi un apostrofo all'inizio della riga.

6.133.1 »

Troverai l'apostrofo a sinistra del tasto Invio. È necessario premere il tasto cancelletto # e il tasto su per creare il carattere apostrofo. Il codice chiave per l'apostrofo è Chr (39).

In Wikibooks e Wikipedia, il carattere apostrofo viene utilizzato per le lettere in grassetto o corsivo.

Durante la programmazione in Gambas, a differenza di VB, è necessario osservare quanto segue:

- L'apostrofo funziona
- Rem non funziona

Esempio:

```
'Questo è un commento e non verrà preso in considerazione quando il  
programma è in esecuzione.'
```

6.134 test di programma con commenti

Commentare e decommentare i comandi è un buon modo per testare i programmi ed eseguirne il debug.

Ad esempio, hai scritto una riga di programma del tipo seguente. (Formula per calcolare l'area circolare.)

```
F = Pi * r * r
```

Poiché il tuo programma sta ancora commettendo un errore qui, commenta la riga e prova a trovare l'errore. Tuttavia, il codice viene conservato e non è necessario immetterlo di nuovo in seguito.

Nella finestra del codice puoi anche commentare con il tasto destro del mouse e lavorare con le abbreviazioni Ctrl + K e Ctrl + U.

Programma di esempio con il calcolo del cerchio. Per questo è necessario sul modulo:

- 1 pulsante di comando
- 2 caselle di testo
- 1 campo etichetta

La prima e la nona riga sono commentate.

```
"File di classe Gambas

PUBLIC SUB Form_Open ()
    Textbox1.Text = ""
    Textbox2.Text = ""
    Button1.Text = "Calcolo del cerchio con raggio:"
    Label1.Text = "Area:"
    Label1.Alignment = 66
    'Allineamento del testo a destra
FINE

Pulsante PUBLIC SUB1_Click ()
    DIM r AS Float
    DIM F AS galleggiante

    IF Textbox1.Text = "" THEN
        r = 0
        Textbox1.Text = Str (0)
    ALTRO
        r = Val (textbox1.text)
    FINISCI SE

    F = Pi*(r^2)
    Textbox2.Text = Str (Round (F, -2))
FINE
```

Modifica il programma in modo da avere la linea

```
F = Pi*(r^2)
```

commentare e per questo

```
F = (Pi*r) ^2
```

accedere.

Quale versione è corretta? Puoi fare a meno della staffa nella linea o no?

```
F = Pi*r^2
```

Cosa significa il comando Round nel programma? Cerca con F1 nella directory dei comandi alfabetici di Gambas. Il comando arrotonda un numero di punto per difetto al numero di posizioni desiderato.

6.135 Reportistica semplice

Un messaggio è molto facile da programmare. Immettere il codice seguente in un modulo vuoto e avviare il programma.

```
PUBLIC SUB Form_Open ()  
    Messaggio ("Questo è un messaggio", "ButtonText")  
    FINE
```

In alternativa, puoi utilizzare il seguente codice:

```
PUBLIC SUB Form_Open ()  
    Message.Info ("Ciao, questo è il messaggio informativo", "OK")  
    FINE
```

6.136 Messaggio con diverse risposte possibili

Oltre al semplice Message.Info, ci sono altri quattro tipi di messaggi:

- Message.Delete (elimina)
- Message.Error (messaggio di errore)
- Message.Question (domanda)
- Message.Warning

Sono possibili fino a tre risposte per queste finestre di messaggio. C'è anche un'icona Schermo:

- Delete = "bidone della spazzatura",
- Errore = "Punto rosso con X",
- Domanda = "?" Punto interrogativo,
- Avviso = "Triangolo giallo con punto esclamativo"

6.136.1 Esempio

L'esempio seguente mostra come funzionano queste finestre di messaggio. Hai bisogno di un pulsante e di un controllo dell'etichetta sul modulo per avviare il programma. Puoi trovarli entrambi nella casella degli strumenti con F6.

6.136.2 Layout

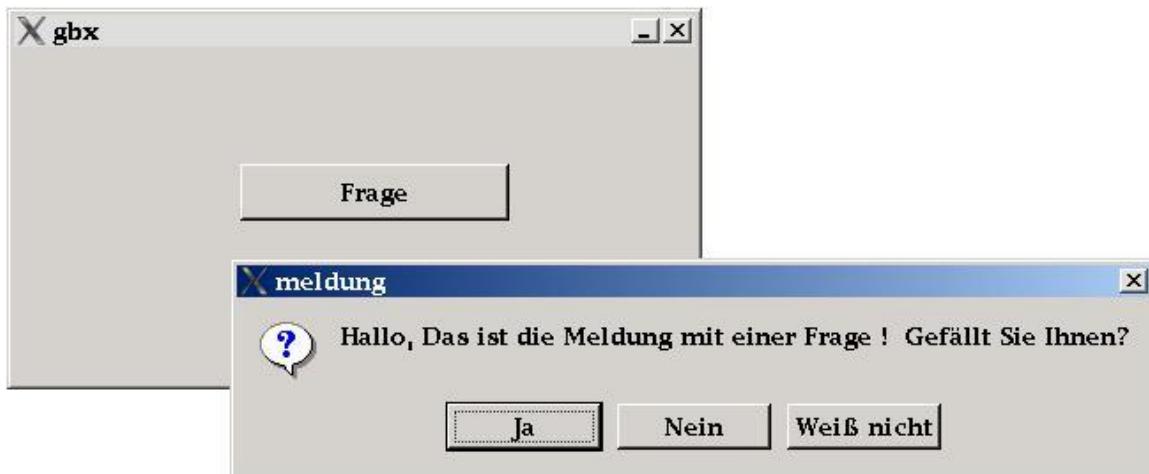


Fig.92

Codice 6.136.3

```
Pulsante PUBLIC SUB1_Click ()  
    SELECT Message.Question ("Ciao, questo è il messaggio con un  
    Domanda! Ti piace? ", "Sì ", "No ", "Non so ")  
        CASO 1  
            Label1.Text = "Piacere!"  
        CASO 2  
            Label1.Text = "Deplorevole!"  
        CASO 3  
            Label1.Text = "Ti piaceranno presto!"  
    FINE SELEZIONA  
FINE
```

Il segno di uguale in Basic viene utilizzato in modo diverso rispetto alla matematica.
Viene utilizzato come assegnazione per una variabile.

Quando un segno di uguale appare nel codice del programma, dovrà sempre dire nella tua mente:

È equiparato. Oppure si dice riempire lo spazio di archiviazione della variabile con invece di è uguale.

Quindi eviti conclusioni illogiche.

Ad esempio, il codice seguente è corretto in Basic. Provalo. Qual'è il risultato?

```
a = 5  
a = a * 5
```

L'esempio seguente mostra come funziona in un programma:

```
Pulsante PUBLIC SUB1_Click ()  
    DIM un numero intero AS  
    a = 5
```

```
a = a * 5
STAMPA a
FINE
```

I capelli di un matematico si drizzerebbero. Sarebbe più logico utilizzare qui 2 variabili diverse.

```
Pulsante PUBLIC SUB1_Click ()
DIM un numero intero AS
DIM b come numero intero
a = 5
b = a * 5
STAMPA b
FINE
```

6.137 Teoria dell'assegnazione

Il codice seguente

```
Variabile = espressione
```

assegna il valore di un'espressione a uno dei seguenti:

- Una variabile locale
- Un parametro di funzione
- Una variabile globale (classe)
- Uno slot per array
- Una variabile pubblica oggetto
- Una proprietà dell'oggetto

Esempi:

```
iVal = 1972
```

```
Nome = "Gambas"
```

```
hObject.Property = iVal
```

```
cCollection [sKey] = nome
```

6.138 Tipi di dati semplici

La tabella seguente elenca i tipi di dati semplici di Gambas. I più importanti sono Boolean, Integer, Float e String.

Cognome	descrizione	Requisiti di memoria	Valore preimpostato
Booleano	Vero o falso	1 byte	Falso
byte	0 ... 255	1 byte	0
Corto	-32768 ... +32767	2 byte	0
Numero intero	-2147483648 ... +2147483647	4 byte	0
Galleggiante	Come il doppio tipo di dati in C.	8 byte	0.0
Data	Data e ora salvate come integrità	8 byte	zero
Corda	Qualsiasi sequenza di Lettere.	4 byte	zero
Variante	Qualsiasi tipo di tipo di dati	12 byte	zero
Oggetto	Un riferimento anonimo a un oggetto	4 byte	zero

6.139 Tipi di dati composti

6.139.1 classi

6.139.2 Oggetti

6.139.3 Arrays

Vedi Gambas: Arrays³²

6.139.4 Collezioni

6.140 Conversione dei tipi di dati

6.141 Molto semplice

Poiché la maggior parte dei programmati ha notevoli problemi con l'OOP quando vengono a contatto per la prima volta con loro, dovrebbe essere scelto un esempio molto semplice per la comprensione:

Durante la programmazione del controllo³³ Per un'interfaccia utente grafica, ad esempio, i pulsanti (pulsanti di comando) vengono gestiti come oggetti di una classe di pulsanti generale.

- Ottimo pulsante
 - Oggetti
 - Button1

³² Capitolo 6.157 a pagina 360

³³ Capitolo 0.60 a pagina 59

- Button2
- Eccetera

Quindi per lo più stai già lavorando con classi e oggetti senza che te ne accorga.

6.142 collegamenti

6.142.1 Gambas

- <http://www.madeeasy.de/7/prgmini.htm#klasse>

6.142.2 Generale

- <http://sepwww.stanford.edu/sep/josman/oop/oop1.htm>

6.143 variabili

6.143.1 Cosa sono le variabili?

Una variabile è una quantità che può avere valori diversi. Quindi è di dimensioni variabili. Le variabili sono anche chiamate segnaposto o incognite. Appaiono in formule e termini.

L'oppuesto di una variabile è la costante, cioè un valore fisso e immutabile. Ad esempio, il numero del cerchio $\pi = 3,14 \dots$ non è una variabile, ma una costante.

Esempi di variabili:

- x, y, z nel sistema di coordinate cartesiane
- $y = x^2$
- Temperatura t
- Frase B

Per il programmatore, una variabile è un'area di memoria a cui può accedere (tenendo conto del tipo di dati) utilizzando un identificatore univoco (nome). Il concetto di variabili è al centro della programmazione imperativa. Una variabile deve essere dichiarata esplicitamente in Gambas.

6.143.2 Dichiarazione (dimensionamento, assegnazione del tipo di dati)

Esistono due tipi principali di informazioni che possono essere memorizzate in una variabile: numeri e testo. Prima di poter utilizzare una variabile, è necessario crearla. Questo viene fatto con il comando DIM AS.

Esempi per la dichiarazione di variabili:

La stella marina

```
DIM x AS intero  
DIM y AS float  
DIM z AS stringa
```

Non è possibile utilizzare la stessa designazione per 2 variabili diverse in una subroutine.

Puoi anche semplicemente tralasciare Dim:

```
x AS intero  
y AS float  
z AS stringa
```

Finora non c'è modo di dichiarare più di una variabile in una riga:

```
intA, intB, intC AS Integer
```

Purtroppo non è possibile. Inoltre, le dichiarazioni vengono sempre fatte all'inizio di un sottoprogramma e non da qualche parte nel codice. Benoit è molto severo al riguardo. La corretta dichiarazione delle variabili è mostrata nell'applet seguente

```
Pulsante PUBLIC SUB1_Click ()  
'Dichiarazione  
x AS intero  
y AS float  
z AS stringa  
'Assegnazione di valori  
x = 2  
y = 2.378  
z = "Questa è una dichiarazione corretta"  
'Uso delle variabili  
stampa x, y, z  
FINE
```

Il seguente programma, invece, produce un messaggio di errore: x già dichiarato

```
Pulsante PUBLIC SUB1_Click ()  
DIM x AS intero  
DIM x AS float  
DIM x AS stringa  
x = 2  
x = 2.378  
y = "Questa non è una dichiarazione corretta"  
stampare x, x, x  
FINE
```

Ci sono problemi anche nel seguente programma, poiché non viene fatta alcuna distinzione tra lettere minuscole e maiuscole:

```
Pulsante PUBLIC SUB1_Click ()
DIM x AS intero
DIM X AS galleggiante
DIM x $ AS stringa
x = 2
X = 2.378
x $ = "Questa è una dichiarazione corretta"
stampa x, X, x $
FINE
```

La dichiarazione di x \$ è corretta.

6.143.3 Assegnazione

Una volta creata una nuova variabile, è possibile assegnarle un valore da salvare. Per fare ciò, viene utilizzato il comando =. L'assegnazione del valore è chiamata assegnazione in inglese.

L'esempio assegna un valore a una variabile chiamata peso.

Esempio:

```
Peso = 80
```

Quando si assegnano i valori, è necessario attenersi a una certa sequenza. Prima vengono dichiarate le variabili, quindi è possibile assegnare loro un valore:

Le variabili devono essere specificate all'inizio di una classe, metodo o funzione.

Esempio corretto:

```
PUBLIC SUB blah ()
peso AS intero
size AS Integer
peso = 75
altezza = 176
FINE
```

Esempio FALSO !! (ordine sbagliato)

```
PUBLIC SUB blah ()
peso AS intero
peso = 75
size AS Integer
altezza = 176
FINE
```

L'ultimo esempio risulta nel messaggio di errore: Identificatore sconosciuto, riga grande: xx in Form1.class

6.143.4 Come si possono limitare le variabili?

Pubblico, privato, statico

Non dovrebbero esserci variabili globali. Facoltativo viene utilizzato solo nella dichiarazione del metodo.

6.143.5 Pubblico - Privato

Entra nel seguente programmino e guarda cosa succede quando fai clic sul file
Fare clic sul pulsante1:

```
PRIVATO SUB Button1_Click ()  
    Numero DIM AS intero  
    'Crea una variabile locale privata  
    Numero = numero + 1  
    Textbox1.Text = Textbox1.Text e numero  
FINE
```

Sorprendentemente, nessun valore viene emesso nella casella di testo. Ciò è dovuto all'identificazione del sottoprogramma con PRIVATO. Scambia il termine PRIVATO con PUBBLICO e prova di nuovo la fortuna:

```
PUBBLICO SUB Button1_Click ()  
    Numero DIM AS intero  
    'Crea una variabile locale pubblica  
    Numero = numero + 1  
    Textbox1.Text = Textbox1.Text e numero  
FINE
```

Ora viene emesso un 1 nella casella di testo. Se si fa clic più volte sul pulsante, l'1 viene visualizzato più volte.

6.143.6 Definizione di variabili in generale

Le variabili possono anche essere dichiarate al di fuori delle funzioni, ecco un esempio:

```
Messaggio PUBBLICO AS String  
PUBLIC SUB Form_Open ()  
    message = "Questo è un test"  
END SUB  
Pulsante PUBLIC SUB1_Click ()  
    Message.Info (messaggio)  
FINE
```

È necessario un pulsante di comando nel modulo per avviare l'esempio.

Form_Open è un evento come Button1_Click. Il codice per questo evento è già chiamato all'avvio del programma, quando il form viene caricato e aperto. Qui la variabile è dichiarata con Public al di fuori di una subroutine. Quindi questa variabile è disponibile fintanto che il modulo è caricato.

Sostituisci il termine PUBBLICO con il termine PRIVATO e prova il risultato.

```

Messaggio PRIVATO AS String
PRIVATE SUB Form_Open ()
    message = "Questo è un test"
END SUB
PRIVATE SUB Button1_Click ()
    Message.Info (messaggio)
FINE

```

6.143.7 Statico (variabile statica)

A volte il valore di una variabile dovrebbe essere preservato. Se si richiama di nuovo la variabile, è possibile continuare a lavorare con il valore memorizzato della variabile. Il comando di dichiarazione statica viene utilizzato per abilitarlo.

L'esempio seguente mostra come funziona. È necessario un pulsante di comando sul modulo per farlo funzionare.

```

Statico Pubblico x Come numero intero
Public Sub Button1_Click ()
    x = x + 1
    stampa x
Fine

```

Se si preme il pulsante di comando, x viene prima definito come 0. Quindi 1 viene aggiunto allo 0 nella riga x = x + 1. Con Print x l'intera cosa viene visualizzata nella finestra diretta. Se si preme più volte il pulsante di comando, è possibile vedere che il programma ha memorizzato il valore di x.

Output nella finestra diretta:

```

1
2
3
4 °
5

```

La parola statica assicura che il valore di una variabile non scompaia semplicemente. Se viene richiamato una seconda volta, il vecchio valore è di nuovo automaticamente disponibile.

Prova la seguente variante:

```

Public Sub Button1_Click ()
    x Come numero intero
    x = x + 1
    stampa x
Fine

```

Premere più volte il pulsante di comando. Qual è l'output?


```
1  
1  
1  
1
```

La sottofunzione azzerà sempre il valore della variabile. Notare che, a differenza di Visual Basic, la variabile statica deve essere definita all'esterno della subroutine.

Saresti deluso se tutto funzionasse anche senza Statico e Pubblico? Prova il codice seguente:

```
x Come numero intero  
Public Sub Button1_Click ()  
    x = x + 1  
    stampa x  
Fine
```

6.143.8 Variabili globali in Gambas

A Benoit Minsini non piacciono. Ma sono possibili e, secondo me, anche molto pratici. Vedi Gambas: `_Form # Globale_Variablen_in_Gambas`³⁴

I seguenti esempi mostrano un altro modo di trattare le variabili globali:

Diverse forme e moduli sono utilizzati in un grande progetto Gambas. Le variabili globali che si applicano all'intero progetto non esistono in Gambas. Tuttavia, ha senso poter accedere alle variabili di altri moduli o moduli da moduli o moduli. Come funziona è mostrato di seguito. Abbiamo bisogno di una forma con un pulsante di comando e una casella di testo per visualizzare i risultati. Abbiamo anche bisogno di un modulo in cui le variabili del modulo vengono utilizzate per il calcolo.

Lettura di variabili pubbliche in un Form1 da una procedura in un modulo.

Codice programma Form1:

```
PUBBLICO a AS Float  
PUBBLICO b AS Float  
' Le variabili sono pubbliche e sono valide in tutte le procedure e funzioni del form1.  
PUBLIC SUB Button1_Click () DIM  
    c AS Float  
    ME.Text = "Esempi" a =  
    3.2  
    b = 7.4  
    c = Module1.RechnenVariable ()  
    textbox1.Text = Str (c)
```

³⁴ Capitolo 0.97 a pagina 91

```
FINE
```

Codice di programma Module1:

```
FUNZIONE PUBBLICA RechnenVariable () AS Float
  DIM c AS float
  c = form1.a + form1.b
  RITORNO c
FINE
```

Quando si preme il pulsante di comando, i valori vengono prima assegnati alle variabili a e b. Chiamando la funzione CalculateVariable in Module1, i valori delle variabili aeb di Form1 vengono letti, quindi aggiunti e restituiti alla procedura chiamante Sub Button1 come valore restituito dalla funzione. Il risultato viene emesso come variabile float in Textbox1.

Assegnare valori per le variabili aeb di Form1 tramite una procedura da Module1 è facile come leggere le variabili.

Codice programma Form1:

```
PUBBLICO a AS Float
PUBBLICO b AS Float
Pulsante PUBLIC SUB1_Click ()
  ME.Text = "Esempi"
  Module1.CalculateVariable ()
  textbox1.Text = Str (a + b)
FINE
```

Codice di programma di Module1

```
PUBLIC SUB CalculateVariable ()
  form1.a = 10.2
  form1.b = 2.5
FINE
```

Leggere le variabili da un array è un po 'più complicato, poiché indirizzare la variabile array dal modulo con il codice c = form1.a [5] non funziona. Di conseguenza, una procedura pubblica è programmata in Form1 che può accedere alla matrice. Questa procedura pubblica è accessibile dal modulo per leggere l'array. Ecco il codice per il Form1:

```
d [10] AS float
' l'array è dichiarato senza PUBLIC. Con Public d [10] AS Float, Gambas emette un
messaggio di errore.
' un array può essere dichiarato solo nell'intestazione dei moduli. Quando si dichiarano
i moduli in testa, Gambas emette anche un messaggio di errore.
' Il conteggio per il contenuto dell'array inizia da 0 (d [0] = ..), mentre il conteggio per la
dichiarazione inizia da 1
' La matrice è pubblica ed è valida in tutte le procedure e le funzioni di Form1.
```

La stella marina

```
PUBLIC SUB Form_Open ()  
DIM i AS intero  
ME.Text = "Esempi"  
PER i = da 0 a 9  
    d [i] = i * 1.2      'La matrice è piena di valori arbitrari.  
    IL PROSSIMO  
FINE  
FUNZIONE PUBBLICA d_aus (x AS Integer) AS Float  
DIM vale come galleggiante  
valore = d [x]  
RITORNO ne vale la pena  
FINE  
Pulsante PUBLIC SUB1_Click ()  
DIM c AS float  
c = Module1.RechnenArray ()  
textbox1.Text = Str (c)  
FINE
```

Codice di programma Module1:

```
FUNZIONE PUBBLICA RechnenArray () AS Float  
DIM c AS float  
c = form1.d_from (2) + form1.d_from (5)  
RITORNO c  
FINE
```

Innanzitutto, l'array viene riempito con i valori nella procedura Open (). Quando la funzione RechenArray viene chiamata in Module1, la funzione d_aus viene chiamata due volte nella forma 1 e il contenuto di d [2] od [5] viene determinato e restituito alla matrice aritmetica della funzione. Lì vengono quindi aggiunti e restituiti alla procedura chiamante Sub Button1 come valore di ritorno della matrice aritmetica della funzione. Il risultato viene emesso come variabile float in Textbox1.

L'assegnazione dei valori per l'array è simile. Al posto delle funzioni, in questo esempio sono necessarie procedure perché i valori vengono trasferiti e non letti. Il codice richiesto per Form1 è:

```
d [10] AS float  
PUBLIC SUB Form_Open ()  
DIM i AS intero  
ME.Text = "Esempi"  
PER i = da 0 a 9  
    d [i] = i * 1.2 'L'array è riempito con valori arbitrari.  
    IL PROSSIMO  
FINE  
PUBLIC SUB d_ein (valore AS Float, x AS Integer)  
    d [x] = valore  
FINE  
Pulsante PUBLIC SUB1_Click ()  
DIM c AS float  
Module1.CalculateArray ()  
textbox1.Text = Str (d [5])  
FINE
```

Codice di programma Module1:

```
PUBBLICO SUBRechnenArray ()
  DIM vale come galleggiante
  valore = 12,7
  form1.d_ein (value, 5) '5 è scelto arbitrariamente
FINE
```

Nel caso di un array bidimensionale, il codice di Form1 per la lettura dei valori è:

```
d [10.5] AS float
PUBLIC SUB Form_Open ()
  DIM i AS intero
  DIM j AS Numero intero
  ME.Text = "Esempi"
  PER i = da 0 a 9
    FORj = da 0 a 4
      d [i, j] = i * L'array 1.2 + j 'è riempito con valori arbitrari.
      IL PROSSIMO
      IL PROSSIMO
    FINE
  FUNZIONE PUBBLICA d_aus (x AS Integer, y AS Integer) AS Float Valore
    DIM AS Float
    valore = d [x, y]
    RITORNO ne vale la pena
  FINE
  Pulsante PUBLIC SUB1_Click ()
    DIM c AS float
    c = Module1.RechnenArray ()
    textbox1.Text = Str (c)
  FINE
```

Codice di programma Module1:

```
FUNZIONE PUBBLICA RechnenArray () AS Float
  DIM c AS float
  c = form1.d_from (5.2) + form1.d_from (8.4)
  RITORNO c
FINE
```

Nel caso di un array bidimensionale, il codice form1 per il trasferimento dei valori è:

```
d [10.5] AS float
PUBLIC SUB Form_Open ()
  DIM i AS intero
  DIM j AS Numero intero
  ME.Text = "Esempi"
  PER i = da 0 a 9
    FORj = da 0 a 4
      d [i, j] = i * L'array 1.2 + j 'è riempito con valori arbitrari.
      IL PROSSIMO
      IL PROSSIMO
    FINE
  PUBLIC SUB d_ein (valore AS Float, x AS Integer, y AS Integer)
    d [x, y] = valore
  FINE
  Pulsante PUBLIC SUB1_Click ()
    Module1.CalculateArray ()
```

```
textbox1.Text = Str (d [5,2])
FINE
```

Codice di programma Module1:

```
PUBBLICO SUBRechnenArray ()
DIM vale come galleggiante
valore = 12,7
form1.d_ein (value, 5,2) '5 e 2 sono scelti arbitrariamente
FINE
```

I numeri in virgola mobile sono stati sperimentati in tutti gli esempi. Apportando piccole modifiche alle procedure e alle funzioni, è ovviamente possibile lavorare anche con variabili intere, stringhe, variabili di dati o anche variabili booleane. Inoltre, non è complicato accedere al contenuto delle caselle di testo, riempire le caselle di testo con il contenuto o leggere o modificare i testi delle etichette.

Di seguito vengono descritti alcuni comandi che possono essere utilizzati per convertire le variabili da un formato a un altro. I comandi Val, Str, Chr, Asc e Int sono particolarmente importanti.

6.144 val

Come convertire stringhe (stringhe di testo) con Val () .

Il comando Val () converte una stringa (= sequenza di caratteri ascetici) in un tipo di dati che corrisponde al contenuto della stringa.

Se la stringa sembra un numero in virgola mobile, Val () la trasforma in un numero in virgola mobile. Val () usa la lingua locale della tua macchina (qui tedesco), che è stata specificata in "locales". Puoi controllare le tue impostazioni con il seguente comando della console:

```
"locales" o "echo $ LANG".
```

Su una macchina tedesca ottieni: de_DE @ euro Attenzione: con numeri decimali e l' impostazione del paese !! Se è presente 6.6 con l'impostazione del paese tedesco in una casella di testo1, consegna

```
val (textbox1.text)
```

una variabile float con il valore 6.6

Se la casella di testo dice 6.6, allora val (textbox1.text) restituisce una data 06.06.04 Se vuoi riconvertire il numero in un testo tedesco, usa str (numero) per convertirlo.

Il programma

Se si immette una stringa, il programma converte la stringa con val e viene verificato il tipo di variabile convertita.

Il risultato è l'output.

Per avviare il programma è necessario nel modulo:

- 1 pulsante di comando
- 1 campo etichetta di testo
- 1 casella di testo
- 1 campo etichetta

Prendi questo dalla casella degli strumenti con F6 e posizionalo sulla forma.

Il codice:

```
PUBLIC SUB _new ()
TextLabel1.Text = "Digita qualcosa che assomigli a un numero o un punto decimale o
una variabile booleana"

FINE
Pulsante PUBLIC SUB1_Click ()
x variante AS
x = Val (TextBox1.Text)
SE IsBoolean (x) ALLORA
    Label1.Text = "Questa è una variabile booleana!"
FINISCI SE
IF IsInteger (x) THEN
    Label1.Text = "Questo è un numero!"
FINISCI SE
SE IsFloat (x) ALLORA
    Label1.Text = "Questo è un numero di punto!"
FINISCI SE
SE IsString (x) THEN
    Label1.Text = "Questa è una stringa!"
FINISCI SE
FINE
```

Teoria Espressione = Val (String) Converte una stringa in un valore booleano, un numero o una data, a seconda del contenuto della stringa La localizzazione corrente viene utilizzata per convertire numeri e date. L'algoritmo di conversione è il seguente: se la stringa può essere interpretata come data e ora (con separatori di data o ora), vengono restituite data e ora. Altrimenti, se la stringa può essere interpretata come un numero in virgola mobile, viene restituito questo numero in virgola mobile. Altrimenti, se la stringa può essere interpretata come un numero intero, viene restituito questo numero intero. Altrimenti, se la stringa è TRUE o FALSE, viene restituito il valore booleano corrispondente. In caso contrario, viene restituito NULL.

esempio

STAMPA Val ("09/06/72 01:00")

=> 09/06/72 01:00:00

PRINT Val ("3.1415")

=> 3,1415 o 31415 a seconda dell'impostazione del paese

PRINT Val ("6.6")

=> 6.6 o 06.06.04 a seconda dell'impostazione del paese !!

```
STAMPA Val ("- 25")
```

=> -25

```
PRINT Val ("True")
```

=> Vero

```
PRINT IsNull (Val ("Gambas"))
```

=> Vero

6.145 str \$

Str è l'esatto opposto del comando Val. Quindi Str converte tutto in una stringa. Ad esempio, se hai calcolato un numero, ad esempio la radice di 3 e desideri visualizzarlo in una casella di testo, puoi utilizzare il comando Str:

```
Textbox1.text = str $ (sqr (3))
```

Attenzione: il comando Str utilizza nuovamente le impostazioni locali del PC, che possono essere la causa di errori.

Dai un'occhiata al seguente semplice esempio. Hai bisogno di una casella di testo e una Pulsante di comando sul modulo per farlo funzionare:

```
Pulsante PUBLIC SUB1_Click ()  
    textbox1.Text = Str (Sqr (3))  
FINE
```

Il comando sqr estrae la radice quadrata di un numero. Invece del 3 tra parentesi dopo Sqr, puoi inserire qualsiasi altro numero positivo o 0. Se si immette un numero negativo, viene visualizzato un messaggio di errore: Errore matematico

Un altro esempio:

```
Pulsante PUBLIC SUB1_Click ()  
    textbox1.Text = Str (8 * 37)  
FINE
```

L'asterisco corrisponde al simbolo tedesco di Malhaben (moltiplicazione)

Oppure prova il codice seguente:

```
Pulsante PUBLIC SUB1_Click ()
    textbox1.Text = Str (Pi)
FINE
```

Negli esempi con gli istruttori di matematica, i comandi Val e Str vengono utilizzati più volte. Vedi Gambas: trainer aritmetico³⁵

L'esempio dalla funzione di aiuto di Gambas è abbastanza incomprensibile per i principianti:

```
' Emissione nello standard output o come messaggio PUBLIC
CONST ON_STDOUT AS Integer = 1
PUBLIC CONST ON_MESSAGE AS Integer = 2
SUB PrintOn (Where AS Integer, What AS Variant) IF Where
    = ON_STDOUT THEN
        STAMPA Cosa
    ALTRIMENTI SE Where =
        ON_MESSAGE THEN Message (Str $
        (What))
    ENDIF
END
```

6.146 asc

Asc restituisce il codice ASCII di una lettera o di un carattere. Se la posizione non è specificata, viene restituito il codice ASCII del primo carattere.

Sintassi = Asc (String [, Position])

Esempio:

```
PRINT Asc ("Gambas")
```

=> Edizione 71

```
PRINT Asc ("Gambas", 3)
```

=> Edizione 109

Esempio 2 È necessario un pulsante di comando per farlo funzionare. L'output avviene con la stampa nella finestra diretta.

```
Pulsante PUBLIC SUB1_Click ()
    x come numero intero
    Per x = da 1 a 6
    PRINT Asc ("coltello", x)
```

³⁵ Capitolo 6.106 a pagina 312

```
Il prossimo  
FINE
```

Un esempio in modalità terminale:

```
STATICO PUBBLICO SUB Principale ()  
test AS stringa  
x AS intero  
test = "1234567 Questo è un test"  
FOR x = 0 TO Len (test)  
    PRINT Asc (test, x)  
IL PROSSIMO  
FINE
```

6.147 chr \$

Questo comando restituisce il carattere ASCII che corrisponde al numero tra parentesi.

```
Carattere = Chr $ (numero)
```

Attenzione! Gambas utilizza internamente il set di caratteri UTF-8, in modo che i numeri > 128 restituiscano caratteri diversi da quelli previsti con il set di caratteri ISO8859-1.

Esempio:

```
STAMPA Chr $ (65)
```

==> A

Un altro esempio, è necessario un pulsante di comando per farlo funzionare. L'output avviene con la stampa nella finestra diretta.

```
Pulsante PUBLIC SUB1_Click ()  
X AS intero  
PER X = 32 A 127  
    STAMPA Chr $ (X);  
IL PROSSIMO  
FINE
```

Con un ciclo, tutti i caratteri compresi tra 32 e 127 vengono emessi uno dopo l'altro su una riga. Per favore, riferisci <http://www.madeeasy.de/7/prgansi.htm>

In modalità terminale puro puoi provare il seguente programma come esempio:

```
STATICO PUBBLICO SUB Principale ()  
x AS intero  
PER x = da 30 a 133  
    STAMPA x, Chr (x)
```

```

IL PROSSIMO
STAMPA
PRINT "Interruzione riga CHR (10)", Chr $ (10)
PRINT "Return CHR (13)", Chr $ (13)
PRINT "spazio Chr $ (32)", "a"; Chr $ (32); "a"
STAMPA "0 CHR48", Chr $ (48), Str (0)
STAMPA "9 CHR57", Chr $ (57), Str (9)
PRINT "@ CHR64", Chr $ (64), Str ("@")
STAMPA "A CHR65", Chr $ (65), Str ("A")
STAMPA "Z CHR90", Chr $ (90), Str ("Z")
PRINT "a CHR97", Chr $ (97), Str ("a")
PRINT "z CHR122", Chr $ (122), Str ("z")
STAMPA "A" + "A"
STAMPA "A" e "A"
FINE

```

6.148 int

Il comando Int (Integer = numero intero) restituisce il numero intero che si trova al di sotto di un punto decimale.

```
3.1 ==> 3
```

Confronta anche il comando Fix Il comando è spesso utilizzato per l'arrotondamento.
esempio

```
PRINT Int (Pi)
```

=> 3

```
PRINT Int (-Pi)
```

=> -4

Esempio: è necessario un pulsante di comando. L'output avviene nella finestra diretta

```

Pulsante PUBLIC SUB1_Click ()
x AS intero
un galleggiante AS
a = -3.456
PER x = da 1 a 10
  a = a + 1
  PRINT a, Int (a)
IL PROSSIMO
FINE

```

Notare che il punto decimale deve essere inserito qui nel codice in notazione inglese con un punto!

```
a = -3.456
```

Risultato:

```
-2.456 -3  
-1.456 -2  
-0.456 -1  
0,544 0  
1.544 1  
2.544 2  
3.544 3  
4.544 4  
5.544 5  
6.544 6
```

6.149 fix

Il comando fix restituisce la parte intera di un punto decimale. Tutto ciò che viene prima della virgola viene mantenuto, ciò che viene dopo la virgola viene eliminato. Confronta anche il comando Int.

Sintassi:

```
Valore = Fix (numero)
```

Esempi semplici:

```
PRINT Fix (Pi)
```

=> 3

```
PRINT Fix (-Pi)
```

=> -3

Esempio di programma: è necessario un pulsante di comando per avviarlo.

```
Pulsante PUBLIC SUB1_Click ()  
x AS intero  
un galleggiante AS  
a = -3.456  
PER x = da 1 a 10  
a = a + 1  
STAMPA a, correggi (a)  
IL PROSSIMO  
FINE
```

Output dei risultati nella finestra diretta:

```
-2.456 -2
-1.456 -1
-0,456 0
0,544 0
1.544 1
2.544 2
3.544 3
4.544 4
5.544 5
6.544 6
```

6.150 cosiddetti

La funzione Signum restituisce un valore che rappresenta il segno di un numero.

Sintassi Sgn (numero)

Note L'argomento numero può essere qualsiasi espressione numerica legale. Il suo segno determina il valore emesso dalla funzione Sgn:

- Se il numero > 0, Sgn (numero) restituisce il valore 1.
- Se il numero = 0, Sgn (numero) restituisce il valore 0.
- Se il numero è <0, Sgn (numero) restituisce il valore -1.

Esempio: tutto ciò di cui hai bisogno è una forma vuota. I risultati vengono visualizzati nella finestra diretta.

```
PUBLIC SUB Form_Open ()
x AS intero
y AS float
PER x = da 1 a 100
y = Rnd (-5, 5)
SELECT CASE Sgn (y) '... assess.
CASO 0 'Se il numero è zero.
    PRINT y & "" & Sgn (y) & "Signum = zero."
CASO 1 'Se il numero è positivo.
    PRINT y & "" & Sgn (y) & "Signum = numero positivo."
CASO -1 'Se il numero è negativo.
    PRINT y & "" & Sgn (y) & "Signum = numero negativo."
FINE SELEZIONA
IL PROSSIMO
FINE
```

Esempio 2

6.151 let

6.152 bool

Questo comando converte un'espressione in una variabile booleana (true, false).

Sintassi;

Boolean = CBool (espressione)

Il risultato sarà falso se l'espressione:

- una variabile booleana con il valore non è corretta.
- è uno zero.
- una stringa di testo con lunghezza zero è ad es. a = ""
- è un oggetto nullo.

In tutti gli altri casi l'espressione è vera. Uno spazio come stringa di testo restituisce anche il valore True quando convertito con CBool!

Esempio:

```
PRINT CBool (0); ""; CBool (1)
```

=> False True "" funziona solo come spaziatore per l'output del risultato.

```
PRINT CBool ("Gambas"); ""; CBool (""), "", CBool ("")
```

=> Vero Falso Vero

```
STAMPA CBool (NULL)
```

=> Falso

6.153 cbyte

6.154 cdate

6.155 cfloat

6.156 cstr

6.157 cint

7 semplici array

Guarda anche: <http://www.madeeasy.de/7/prgmini.htm#arr>

Gli array sono elenchi o tabelle in cui vengono salvati testi o valori dello stesso tipo di file che appartengono insieme. Ogni elemento dell'array può essere indirizzato individualmente.

Esempio di array: elenco di 5 nomi

- Anna
- Anton
- Toni
- Max
- Klaus

Per definire gli array, un semplice segno di uguale non è sufficiente, è necessaria una dichiarazione speciale: per ottenere un elenco di nomi come quello sopra in gambas, è meglio usare un array di stringhe (array di testo):

Esempio:

```
PUBLIC SUB Form_Open ()  
una stringa AS []  
x AS intero  
a = NUOVA stringa []  
PER x = da 0 a 4  
    a. Aggiungi (Str (x))  
IL PROSSIMO  
    a [0] = "Anna"  
    a [1] = "Anton"  
    a [2] = "Toni"  
    a [3] = "Max"  
    a [4] = "Klaus"  
PER x = da 0 a 4  
    STAMPA a [x]  
IL PROSSIMO  
FINE
```

Per avviare questo programma di esempio, tutto ciò di cui hai bisogno è un modulo vuoto. Il programma è diviso in 4 parti:

- 1. Viene dichiarato un array di stringhe.

```
una stringa AS []
```

- 2. La matrice di stringhe viene inizializzata.

```
DIM x AS intero  
a = NUOVA stringa []  
PER x = da 0 a 4
```

```
a. Aggiungi (Str (x))  
IL PROSSIMO
```

- 3. L'array di stringhe è riempito con elementi reali:

```
a [0] = "Anna"  
a [1] = "Anton"  
a [2] = "Toni"  
a [3] = "Max"  
a [4] = "Klaus"
```

- 4. Per verificare se gli elementi sono stati effettivamente salvati nell'array, vengono quindi emessi di nuovo con un ciclo e il comando di stampa nella finestra diretta.

```
PER x = da 0 a 4  
STAMPA a [x]  
IL PROSSIMO
```

Sembra piuttosto complicato, o cosa ne pensi? È anche un po 'più semplice se usi il comando array. Questo è descritto nel prossimo capitolo.

Sorprendentemente, il programma di cui sopra funziona solo in modalità grafica. Ma forse mi manca solo un trucco per abilitarlo in modalità terminale. Inoltre, la dichiarazione è piuttosto macchinosa. `a = NEW String [5]` non funziona.

È possibile indirizzare i singoli elementi tramite la loro posizione nell'array, dove il primo elemento ha la posizione 0:

Denominazione delle voci

```
a [0] = "Anna"  
a [1] = "Anton"  
a [2] = "Toni"  
a [3] = "Max"  
a [4] = "Klaus"
```

Per avere il nome Max output, bisogna poi scrivere:

```
stampa un [3]
```

Si noti che a non è un elemento di Gambas, ma è stato definito come un array nel programma di esempio sopra.

Le parentesi quadre non sono informazioni opzionali qui, come è normale con i testi del computer, ma devi davvero usare le parentesi quadre per gli array.

L'indice di un array inizia sempre da 0, quindi gli indici validi per un array di dimensione n sono: 0, 1, 2, ..., n-1.

La gestione degli array richiede molto tempo per abituarsi ai principianti. Tuttavia, vale la pena familiarizzare con i diversi tipi e opzioni di programmazione degli array, perché rendono i programmi più chiari e più brevi. Gli array sono indispensabili quando si programmano database o tabelle.

Oltre all'array di stringhe mostrato qui, puoi anche definire un array per qualsiasi altro tipo di file. Quindi ci sono array binari, array di testo, array di interi, array per numeri in virgola mobile, ecc. Puoi anche usare array per tipi di file compositi come gli oggetti.

Gli elementi di un array hanno tutti lo stesso tipo di file. Se vuoi anche variare il tipo di file nell'elenco, devi usare una raccolta invece di un array.

7.1 Riempimento più veloce con il comando array

Il comando array accorcia enormemente il programma precedente. Puoi usarlo per riempire un array molto più velocemente.

L'esempio seguente mostra come eseguire questa operazione. Hai solo bisogno di una forma vuota per farlo andare:

```
PUBLIC SUB Form_Open ()  
x AS intero  
una stringa AS []  
a = array ("Anna", "Anton", "Toni", "Max", "Klaus")  
PER x = da 0 a 4  
    stampa una [x]  
    'Output dell'array nella finestra diretta  
IL PROSSIMO  
FINE
```

Se non desideri reindirizzare l'output alla finestra diretta, aggiungi un'area di testo alla tua forma e modifica il codice come segue:

```
PUBLIC SUB Form_Open ()  
x AS intero  
una stringa AS []  
a = array ("Anna", "Anton", "Toni", "Max", "Klaus")  
Textarea1.Text = ""  
    PER x = da 0 a 4  
        Textarea1.Text = Textarea1.Text & a [x] & Chr (10)  
    IL PROSSIMO  
FINE
```

Il carattere chr (10) inserisce un'interruzione di riga tra i nomi. Con Textarea1.Text = "" la textarea viene cancellata all'inizio.

7.2 Routine di array semplici

7.2.1 Generazione

Deve essere generato un array intero con 20 elementi. A tale scopo, utilizzare la riga seguente:

```
campo [20] AS Integer
```

Anche questo è un po 'più complicato:

```
campo COME NUOVO Intero []
i AS intero
PER i = da 0 a 19
    field.Add (i)
IL PROSSIMO
```

Notare, tuttavia, che ogni array inizia con [0]. Quindi l'ultimo elemento è il campo [19] (da 0 a 19 = 20 elementi). Secondo il campo della dichiarazione [20] AS Integer, si presume che anche il campo dell'elemento [20] abbia un valore. Ma questo non è vero, è già fuori dalla dichiarazione (fuori limite).

7.2.2 Operazioni con gli elementi

Con gli elementi array, tutte le operazioni possono essere eseguite normalmente.
Esempio:

```
campo [2] = 77
campo [3] = campo [2] - 7
```

La prima riga assegna il numero 77 al 3 ° (terzo!) Elemento della matrice. La seconda riga assegna il numero $77 - 7 = 70$ al 4 ° (quarto!) Elemento della matrice.

7.2.3 Indice

È possibile specificare una variabile o una costante come indice:

```
n come numero intero
n = 1
campo [n] = 25
```

Questo codice assegna il valore 25 all'ennesimo elemento (il 2 °).

Come programma:

```
PUBLIC SUB Form_Open ()
campo [4] AS intero
n AS intero
campo [2] = 77
campo [3] = campo [2] - 7
n = 1
campo [n] = 25
PER n = 0 A 3
    Campo PRINT [n]
IL PROSSIMO
FINE
```

Output nella finestra diretta:

```

0
25th
77
70

```

Se si immette il campo di stampa [4], viene visualizzato il messaggio di errore: Fuori limite

Prova di nuovo la seguente variante, quindi potrebbe essere più chiara:

```

PUBLIC SUB Form_Open ()
campo [4] AS intero
'Dichiara il campo della matrice numerica [da 0 a 3] !!!
n AS intero
campo [2] = 77
campo [3] = campo [2] - 7
n = 1
campo [n] = 25
PER n = 0 A 3
    PRINT "field [" & n & "] =" & field [n]
IL PROSSIMO
'Il campo PRINT [4] restituisce un errore !!!
FINE

```

7.2.4 Dimensioni multiple

Inoltre, è possibile creare array con più di una dimensione.

```
scacchiera [10, 10] AS Integer
```

Genera un campo intero bidimensionale di dimensioni 10 x 10.

Come esempio di programma:

```

PUBLIC SUB Form_Open ()
i AS intero
n AS intero
scacchiera [10, 10] AS Integer
PER i = da 0 a 9
    PER n = 0 A 9
        STAMPA i, n
        scacchiera [i, n] = i*10 + n
        STAMPA scacchiera [i, n]
    IL PROSSIMO
    IL PROSSIMO
FINE

```

7.3 Errori con gli array

Esempio 2:

Matrici semplici

In questo esempio un array di stringhe è definito e riempito con numeri consecutivi all'inizio. Il rispettivo numero corrisponde alla posizione nella matrice. Questo non è più il caso del secondo riempimento della matrice. Ora l'array è stato riempito con numeri quadrati.

Per avviare il programma, sono necessari 2 pulsanti di comando e un campo di etichetta di testo.

```
ar AS String []
'L'array è definito come una stringa
Pulsante PUBLIC SUB1_Click ()
'L'array viene emesso
i AS intero
txt come stringa
ar = NEW String []
PER i = da 0 a 3
    ar. Aggiungi (Str (i))
IL PROSSIMO
PER i = da 0 a 3
    txt = txt & ar [i]
IL PROSSIMO
TextLabel1.Text = txt
FINE
PUBLIC SUB Button2_Click ()
'L'array verrà riempito
un numero intero AS
i AS intero
txt come stringa
ar = NEW String []
PER a = da 0 a 3
    ar. Aggiungi (Str (a*un))
IL PROSSIMO
PER i = da 0 a 3
    txt = txt & ar [i]
IL PROSSIMO
TextLabel1.Text = txt
FINE
```

Puoi anche accorciare questo programma con il comando array:

```
ar AS Integer []
'L'array è definito come un elenco di numeri
Pulsante PUBLIC SUB1_Click ()
'L'array viene riempito e visualizzato
i AS intero
txt come stringa
ar = matrice (0,1,2,3)
PER i = da 0 a 3
    txt = txt & ar [i]
IL PROSSIMO
TextLabel1.Text = txt
FINE
PUBLIC SUB Button2_Click ()
'La matrice viene riempita di nuovo con i numeri quadrati del primo riempimento

un numero intero AS
i AS intero
txt come stringa
ar = matrice (ar [0] ^2, ar [1] ^2, ar [2] ^2, ar [3] ^2)
PER i = da 0 a 3
    txt = txt & ar [i]
IL PROSSIMO
TextLabel1.Text = txt
FINE
```

Attenzione! Questo programma funziona correttamente solo una volta. Durante la seconda esecuzione del programma viene visualizzato un messaggio di errore.

Tipo mancata corrispondenza: Wanted Integer [], ottenuto invece float [].

L'annidamento dell'array è quindi soggetto a errori:

```
ar = matrice (ar [0] ^2, ar [1] ^2, ar [2] ^2, ar [3] ^2)
```

Come esercizio, puoi creare un array di tutte le lettere minuscole dell'alfabeto:

Esempio 3: un array tridimensionale

Per avviare il programma è necessario un modulo e un pulsante di comando.

```
Pulsante PUBLIC SUB1_Click ()  
DIM i AS intero  
DIM ii AS Integer  
DIM iii AS Integer  
Matrice DIM [3, 3, 3] AS Integer  
PER i = da 0 a 2  
    PER ii = da 0 a 2  
        PER iii = da 0 a 2  
            STAMPA i, ii, iii  
            Matrice [i, ii, iii] = i*9 + ii*3 + iii  
            Matrice PRINT [i, ii, iii]  
            IL PROSSIMO  
        IL PROSSIMO  
    IL PROSSIMO  
FINE
```

7.4 Split >> Divide un testo in un array di stringhe con molti elementi individuali

Il comando Dividi divide una stringa di testo in piccole parti di testo, separate da un separatore specifico.

La sintassi del comando è:

```
Matrice = Suddividi (testo [, separatore, non separatore])
```

Se si utilizza il comando semplicemente con la specifica del testo, la virgola viene utilizzata come separatore.

```
Array = Split (testo)
```

Ad esempio, se desideri separare una frase nelle sue parole, puoi utilizzare la seguente variante:

```
Array = Split (frase, "")
```

Se si desidera utilizzare diversi separatori diversi (ad es. Virgola e punto), è necessario passarli come secondo parametro combinato in un'unica stringa.

Se non si desidera separare determinate aree, è possibile specificare un non separatore. Tutto ciò che si trova tra due non separatori (ad esempio tra virgolette) non viene suddiviso.

Non esiste un'impostazione predefinita per il non separatore. Devi sempre inserirlo esplicitamente.

La funzione split restituisce un array di stringhe (= un elenco di parole), che è stato riempito con i singoli elementi split.

Esempio: è necessario un pulsante di comando per avviare l'esempio.

```
Pulsante PUBLIC SUB1_Click ()  
e AS String []  
s come stringa  
e = Split ("Gambas Almost Means BASIC!", "")  
PER CIASCUNO IN e  
STAMPA s  
IL PROSSIMO  
Fine
```

Produzione:

```
Gamberi  
Quasi  
Si intende  
DI BASE  
!
```

Prova lo stesso programma con il separatore a. L'output riceverai quindi:

G mb s Almost Me ns BASIC!

Il separatore non è quindi incluso negli elementi dell'array.

Il comando split è molto comodo e funziona bene in una varietà di programmi. Un esempio può essere trovato in Gambas: Statistics¹

7.5 Join Il comando per unirsi

Il comando riunisce di nuovo un elenco. È l'opposto del comando di divisione.

¹ Capitolo 0.131.2 a pagina 144

7.6 Per ciascuno

Con For each, tutte le voci di un array vengono assegnate a un'altra variabile, che può quindi essere emessa.

Esempio: Addy il totalizzatore è molto popolare. Usa anche un array di stringhe.

Hai bisogno di un'area di testo, una casella di testo e un pulsante di comando per avviare il programma.

```
Pulsante PUBLIC SUB1_Click ()
Testo DIM AS String
Somma DIM AS Float
Elenco DIM come stringa []
DIM post AS String
text = textarea1.Text
list = Split (text, Chr (10))
PER OGNI voce IN lista
    somma = somma + val (elemento)
IL PROSSIMO
textbox1.Text = sum
FINE
```

Inserisci alcuni numeri nell'area di testo con A capo in mezzo. Quindi premi il tasto di comando. I numeri vengono sommati. L'elenco nell'area di testo può essere corretto secondo necessità. Vengono elaborati anche i numeri negativi.

Attenzione: non inserire lettere o caratteri speciali. Addy non è ancora così robusto.

7.7 Routine di array Aggiungi, Rimuovi, Ridimensiona, Ripristina, Cancella, Ordina

Gambas offre una serie di routine molto utili per tutti i suoi tipi di array.

7.7.1 Routine degli array

- Inserisci
 - aggiungi un elemento alla fine o inseriscilo in una posizione
- Rimuovere
 - rimuovere un elemento alla fine o rimuoverlo in una posizione definita
- Ripristina
 - Inverti l'ordine
- Ridimensiona
 - Ridimensiona l'array
- Chiaro
 - Elimina tutti gli elementi
- Ordinare
 - Ordina in ordine crescente o decrescente

Il seguente programma mostra come funzionano le routine dell'array.

7.7.2 Hai bisogno

- 11 pulsanti
 - 4 caselle di testo
 - 1 area di testo
- per avviare il programma.

7.7.3 Layout

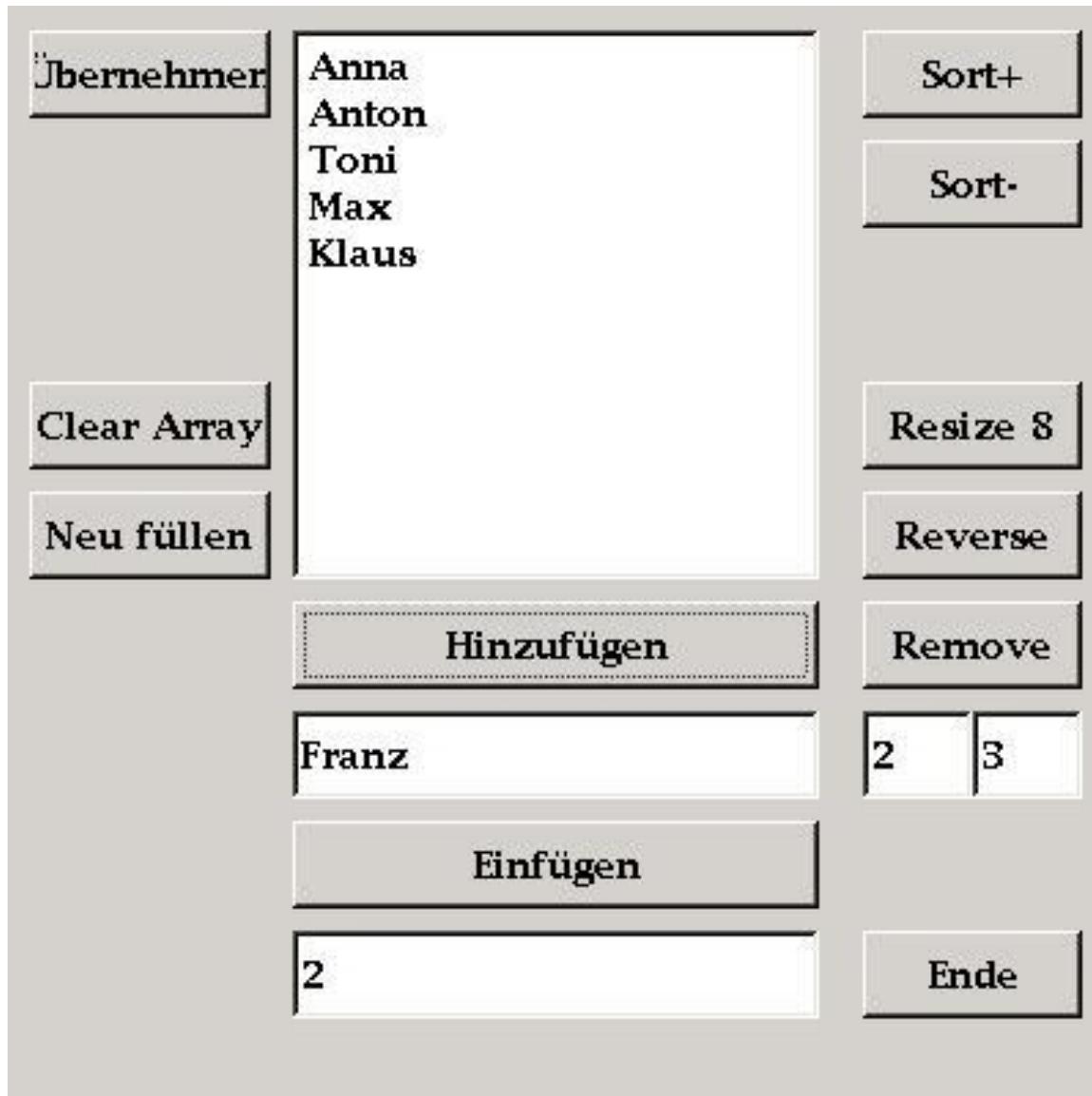


Fig.93

7.7.4 Codice

```
PUBLIC una stringa AS []
PUBLIC SUB Form_Open ()
x AS intero
a = array ("Anna", "Anton", "Toni", "Max", "Klaus")
Textarea1.Text = ""
FOR x = 0 TO (a. Lunghezza - 1)
    Textarea1.Text = Textarea1.Text & a [x] & Chr (10)
    IL PROSSIMO
Textbox1.Text = "Franz"
Button1.Text = "Aggiungi"
Button2.Text = "Inserisci"
Textbox2.Text = "2"
Textbox3.Text = "2"
Textbox4.Text = "3"
ME.Text = "Comandi array"
FINE
```

```
Pulsante PUBLIC SUB1_Click ()
x AS intero
a. Aggiungi (Textbox1.Text)
Textarea1.Text = ""
FOR x = 0 TO (a. Lunghezza - 1)
    'La lunghezza dell'array viene contata da 1, la posizione da 0 a Textarea1.Text =
        Textarea1.Text & a [x] & Chr (10)
    IL PROSSIMO
FINE
```

```
PUBLIC SUB Button2_Click ()
x AS intero
a. Aggiungi (Textbox1.Text, Val (Textbox2.Text))
Textarea1.Text = ""
FOR x = 0 TO (a. Lunghezza - 1)
    Textarea1.Text = Textarea1.Text & a [x] & Chr (10)
    IL PROSSIMO
FINE
```

```
Pulsante PUBLIC SUB 3_Click ()
x AS intero
a. Ordina (0)
Textarea1.Text = ""
FOR x = 0 TO (a. Lunghezza - 1)
    Textarea1.Text = Textarea1.Text & a [x] & Chr (10)
IL PROSSIMO
FINE
```

```
Pulsante PUBLIC SUB 4_Click ()
x AS intero
a. Ordina (16)
Textarea1.Text = ""
FOR x = 0 TO (a. Lunghezza - 1)
    Textarea1.Text = Textarea1.Text & a [x] & Chr (10)
    IL PROSSIMO
FINE
```

```
Pulsante PUBLIC SUB 5_Click ()
x AS intero
un chiaro
Textarea1.Text = ""
FOR x = 0 TO (a. Lunghezza - 1)
    Textarea1.Text = Textarea1.Text & a [x] & Chr (10)
```

Matrici semplici

```
IL PROSSIMO  
FINE
```

```
Pulsante PUBLIC SUB 6_Click ()  
x AS intero  
a. ridimensionare (8)  
Textarea1.Text = ""  
FOR x = 0 TO (a. Lunghezza - 1)  
    Textarea1.Text = Textarea1.Text & a [x] & Chr (10)  
IL PROSSIMO  
FINE
```

```
Pulsante PUBLIC SUB 7_Click ()  
x AS intero  
a. inverso  
Textarea1.Text = ""  
FOR x = 0 TO (a. Lunghezza - 1)  
    Textarea1.Text = Textarea1.Text & a [x] & Chr (10)  
IL PROSSIMO  
FINE
```

```
Pulsante PUBLIC SUB 8_Click ()  
x AS intero  
a.remove (Val (textbox3.text), Val (textbox4.text))  
Textarea1.Text = ""  
FOR x = 0 TO (a. Lunghezza - 1)  
    Textarea1.Text = Textarea1.Text & a [x] & Chr (10)  
IL PROSSIMO  
FINE
```

```
Tasto PUBLIC SUB 9_Click ()  
ME.Chiudi  
FINE
```

```
Pulsante PUBLIC SUB 10_Click ()  
x AS intero  
a = array ("Anna", "Anton", "Toni", "Max", "Klaus")  
Textarea1.Text = ""  
FOR x = 0 TO (a. Lunghezza - 1)  
    Textarea1.Text = Textarea1.Text & a [x] & Chr (10)  
IL PROSSIMO  
FINE
```

```
Pulsante PUBLIC SUB 11_Click ()  
testo come stringa  
posta AS String  
text = textarea1.Text  
a = split (text, Chr (10))  
PER OGNI post IN a  
IL PROSSIMO  
FINE
```

È possibile semplificare questo programma se si impacchetta l'output dell'array nell'area di testo in una subroutine, perché si verifica ancora e ancora.

```
x come numero intero  
Textarea1.Text = ""  
FOR x = 0 TO (a. Lunghezza - 1)
```

```
Textarea1.Text = Textarea1.Text & a [x] & Chr (10)
IL PROSSIMO
```

A proposito: l'ordinamento non funziona ancora per l'array Object [].

7.8 Riempire la matrice con numeri casuali

Puoi anche riempire un array con numeri casuali, ad esempio per testare algoritmi per l'ordinamento.

Il seguente mini-programma mostra come farlo. A tale scopo, sono necessari un modulo e un pulsante di comando. L'array viene creato e riempito quando il modulo viene caricato. Se si preme il pulsante di comando, l'array viene visualizzato nella finestra diretta.

6 numeri casuali compresi tra 1 e 10 vengono generati e letti nell'array.

Ci sono alcune incongruenze nella denominazione degli array (vedere i commenti del programma) Inoltre, ora puoi vedere il vantaggio che ci sono 2 metodi di riempimento degli array.

```
' File di classe Gambas
ar AS Float []
'perché è presente un messaggio di errore,
'se specifichi l'array con Integer []. PUBLIC SUB
Form_Open ()
    Rendi casuale ()
    ar = Array (Int (Rnd (1.10)), Int (Rnd (1.10)), Int (Rnd (1.10)), Int (Rnd (1.10)), Int
(Rnd (1, 10)), Int (Rnd (1, 10)))
    FINE
PUBLIC SUB Button1_Click ()
elemento AS Integer
    PER OGNI elemento IN
        elemento ar PRINT
    FINE
SUCC
ESSIV
A
```

Alternativa:

```
ar AS Integer []
'Sebbene l'array sia definito come un array intero,
"funziona di seguito come un array di stringhe. Perché?
PUBLIC SUB Form_Open ()
    x AS intero
    ar = NEW Integer []
    PER x = da 0 a 10
        ar. Aggiungi (Str (x))
    IL PROSSIMO
    FINE
Pulsante PUBLIC SUB1_Click ()
x AS intero
Rendi casuale ()
PER x = da 0 a 10
    ar [x] = Int (Rnd (1,10))
    IL PROSSIMO
    FINE
PUBLIC SUB Button2_Click ()
```



```
elemento AS Integer  
PER OGNI elemento IN ar  
    Elemento PRINT;  
IL PROSSIMO  
STAMPA  
FINE
```

Quanti elementi ha l'array? Perchè così tanti Prova il programma una volta senza il comando Randomize e ti renderai conto che è necessario.

7.9 numeri della lotteria (7 su 49)

Il seguente programma genera sette numeri della lotteria, elimina i doppi e restituisce i numeri ordinati: Viene utilizzato un array con 49 elementi.

Innanzitutto, il programma viene scomposto nelle sue singole parti e posizionato su 4 pulsanti di comando.

- Commandbutton1 >> Array con 7 numeri casuali
- Commandbutton2 >> Esegue l'output dell'array completo tramite Print
- Commandbutton3 >> Visualizza solo gli elementi che non sono uguali a zero
- Commandbutton4 >> Riempì di nuovo l'intero array con zeri.

```
ar AS Integer []  
PUBLIC SUB Form_Open ()  
'Quando il modulo viene aperto, la matrice viene riempita con tutti zeri.  
  
x AS intero  
ar = NEW Integer []  
PER x = da 0 a 48  
    ar.Add (Str (0))  
IL PROSSIMO  
FINE  
Pulsante PUBLIC SUB1_Click ()  
'quindi l'array viene riempito con 7 numeri casuali nelle posizioni che corrispondono al  
numero casuale x AS Integer  
  
r AS intero  
Rendi casuale ()  
PER x = da 0 a 6  
    r = Int (Rnd (1.49))  
    SE r = ar [r] ALLORA x = x - 1  
    'Se si verifica un doppelganger torna indietro di uno ar [r] = r  
  
IL PROSSIMO  
FINE  
PUBLIC SUB Button2_Click ()  
'L'intero array viene emesso a scopo di test  
elemento AS Integer  
    PER OGNI elemento IN ar  
        Elemento PRINT  
    IL PROSSIMO  
FINE  
Pulsante PUBLIC SUB 3_Click ()  
'Ora vengono emessi solo gli elementi che non sono uguali a 0  
  
elemento AS Integer  
    PER OGNI elemento IN ar  
        IF element <> 0 THEN PRINT element  
    IL PROSSIMO
```

```

FINE
Pulsante PUBLIC SUB 4_Click ()
'l'array è completamente azzerato
x AS intero
PER x = da 0 a 48
    ar [x] = 0
IL PROSSIMO
FINE

```

Dopo aver suddiviso il programma in singoli passaggi, è ora possibile riepilogarlo su un unico pulsante. E ovviamente manca ancora il numero aggiuntivo.

```

' File di classe Gambas Numeri Lotto ar
AS Integer []
PUBLIC SUB Form_Open ()
'Quando il modulo viene aperto, la matrice viene riempita con tutti zeri.

x AS intero
ar = NEW Integer []
FOR x = 0 TO 48
    ar.Add (Str (0))
FINE
SUC
CES
SIVA
PUBLIC SUB Button1_Click () x
AS Integer
r AS Integer elemento
AS Integer
'l'array è completamente impostato su zero FOR x = 0
TO 48
    ar [x] = 0
IL PROSSIMO
'quindi la matrice viene riempita con 7 numeri casuali nei punti che corrispondono al numero
casuale
Randomize ()
FOR x = da 0 a 6
    r = Int (Rnd (1.49))
    SE r = ar [r] ALLORA x = x - 1
    'Se si verifica un doppelganger torna indietro di uno ar [r] = r

IL PROSSIMO
'vengono emessi solo i numeri <> 0 PER OGNI
elemento IN ar
    IF element <> 0 THEN PRINT element
FINE
SUC
CES
SIVA

```

Pensa al motivo per cui gli elementi dell'array sono stati azzerati due volte. Se non lo sai, perché non lanciare un ciclo e provare il programma alcune volte. Quindi riconoscerai la necessità quando premi più volte il pulsante Befhel.

7.10 Lanciare doppelganger fuori dall'array

A tale scopo, viene creato un array temporaneo e vengono copiati gli elementi che non esistono ancora nel nuovo array temporaneo.

Esempio come funzione:


```
FUNZIONE PUBBLICA RemoveDuplicates (sArray AS String []) AS String []
  DIM s AS String
  DIM tempArray AS NEW String []
  PER CIASCUNO IN sArray
    IF tempArray.Find (s) = -1 THEN tempArray.Add (s)
    IL PROSSIMO
  RETURN tempArray
FINE
```

Chiama con:

```
...
RemoveDuplicates (array)
...
```

7.11 teoria

.Array Questa classe virtuale è la classe genitore di ogni altro array Gambas.

Questa classe è virtuale. Non puoi usarlo come tipo di dati. Non può essere creato (creatable).).

Caratteristiche:

- Count (identico a Length)
- Lunghezza

Metodi

- Inserisci
- Chiaro
- Rimuovere
- Ridimensiona
- Inversione
- Ordinare

8 tipi di array

Gli array in Gambas di solito possono essere riconosciuti dalle parentesi quadre: []

Con il metodo rapido che utilizza il comando array, tuttavia, possono anche essere definiti con parentesi tonde ().

Se non sai che tipo di array hai definito, puoi visualizzare il tipo di array nella finestra immediata. Quando hai riempito l'array con il comando array, il tipo dell'array è definito dal primo elemento. Gli altri elementi vengono convertiti automaticamente nello stesso tipo.

Con Join puoi separare gli elementi dell'array in uscita

Esempio:

```
PUBLIC SUB Form_Open ()  
ar AS Integer []  
ar = matrice (3,3,3)  
PRINT Object.Type (ar)  
FINE
```

Output: numero intero []

```
PRINT Object.Type (Array ("2.4", 3, 3.2))
```

Output: String []

```
PRINT ["A", "B", "C"] .Join ("/")
```

Edizione A / B / C

9 Creare una matrice bidimensionale

Vedi Gumbas: tabella¹

¹ Capitolo 0.105 a pagina 101

10 routine di array

- Crea campi bidimensionali
- Un array booleano
- Determina il numero di dimensioni in un array
- Espandi dinamicamente gli array
- FAQ - Domande frequenti sugli array
- Elemento di ricerca nell'array
- Inserimento e cancellazione rapidi
- Mescola rapidamente i contenuti dell'array
- Ordinamento veloce con QuickSort e MinSort

10.1 Raccolta di materiali e domande sugli array

Se vuoi separare le singole voci con un carattere, di solito puoi farlo con il metodo di unione.

Aggiunta di voci

Come posso aggiungere voci a un array?

Conteggio delle voci

Come vengono contate le iscrizioni? (Proprietà Length)

STAMPA a.Count

Cerca voci

Se vuoi cercare una voce in un array, funziona in questo modo:

Array.Find (SearchValue)

Ordina le voci

Come puoi ordinare gli array?

Array.Sort

Come puoi ordinare in base ai nomi dei valori? Come puoi ribaltare gli elementi nel loro ordine?

Array.Reverse

All'inizio, non lasciare che i molti comandi diversi ti confondano. All'inizio, scegli solo i più importanti e lavora con loro.

AS, Print, =, &, FOR TO, NEXT, ME.Chiudi, SE ALLORA, VERO, FALSO

La struttura per una subroutine viene specificata automaticamente con PUBLIC SUB e END alla fine di Gambas quando si fa clic su un pulsante di comando, ad esempio.

Ogni volta che ti imbatti in un nuovo comando, usa semplicemente F1 per scoprire cosa significa nella guida.

10.2 Indice alfabetico dei comandi

Su gambasdoc.org¹ troverai una panoramica sempre aggiornata di tutti i comandi di Gambas.

Segni matematici in gambas

Numero + numero	Somma due numeri, somma
- Numero	Calcola il numero negativo di un numero. Il numero negativo di zero è zero. N = 5 R = -8 PRINT N; ""; R; ", "; -N; ""; -R ==> 5-8, -5 8
Numero - numero	Sottrai, sottrae un numero da un altro
Numero * numero	Moltiplica due numeri, moltiplica
Numero / numero	Forma una pausa. Un numero è diviso per un altro. Un div- Quando il divisore viene visualizzato l'errore sione per zero (= divisione per zero) è uguale a zero. Il risultato non è sempre un numero intero. Perciò- gen, il risultato dovrebbe essere dichiarato come una variabile float.
Numero ^ potenza	Un numero viene elevato al potere. STAMPA 4^3 ==> 64
Numero \ numero	Calcola il quoziente di 2 numeri. Una divisione per zero (= divi- da sione a zero) si verifica quando il numero a destra della barra rovesciata (Slash) è zero. A \ B è uguale a INT (A / B). STAMPA 9 \ 4; ","; 9 MOD 4 ==> 2, 1
Numero Numero MOD	Calcola il resto di una frazione di due numeri. Una divisione per zero (= Divisione per zero) L'errore si verifica quando il numero si trova a destra del Il comando MOD è zero. STAMPA 9 \ 4; ","; 9 MOD 4

[] ==> 2, 1]

1 http://gambasdoc.org/help/lang

Numero E numero	Calcola l'AND matematico dei valori binari di due numeri. STAMPA 5 E 3 ==> 1
Numero O numero	Calcola l'OR matematico (o) dei valori binari di due numeri len. STAMPA 5 O 3 ==> 7
Numero Numero XOR	Calcola l'ExOR matematico (o o) dei valori binari due numeri. STAMPA 5 XOR 3 ==> 6

10.3 E

Di solito il comando AND non viene utilizzato molto spesso.

Ha 2 funzioni:

1. Combinazione logica di due affermazioni e verifica del loro valore di verità
2. Confronto di 2 numeri nel sistema numerico binario e output delle corrispondenze

Esempio per la prima funzione:

STAMPA VERO E FALSO

Uscita: falso

STAMPA VERO E VERO

Risultato: vero

Un riepilogo può essere trovato nel seguente programma:

```
PUBLIC SUB Form_Open ()
    DIM a AS Boolean
    DIM b AS Boolean
    PER a = da 0 a 1
        PER b = da 0 a 1
            PRINT Str $ (a) & "AND" & Str $ (b) & "==" & Str $ (a AND b)
    IL PROSSIMO
    IL PROSSIMO
FINE
```

Tabella della verità:

UN.	B.	A AND B
Falso	Falso	Falso

UN.	B.	A AND B
Falso	Vero	Falso
Vero	Falso	Falso
Vero	Vero	Vero

Circuito elettrico: il collegamento AND corrisponde a un circuito elettrico in serie o in serie.

10.4 OR (Or)

Il comando logico OR collega due istruzioni e fornisce un nuovo valore di verità per il collegamento.

OR = Or = Uno o l'altro o entrambi.

È sufficiente che una delle due affermazioni sia vera. Anche allora, il collegamento OR è già vero.

Un riepilogo può essere trovato nel seguente programma:

```
PUBLIC SUB Form_Open ()
    DIM a AS Boolean
    DIM b AS Boolean
    PER a = da 0 a 1
        PER b = da 0 a 1
            STAMPA Str $ (a) & "OR" & Str $ (b) & "=>" & Str $ (a OR b)
        IL PROSSIMO
    IL PROSSIMO
FINE
```

Tabella della verità:

UN.	B.	A OR B
Falso	Falso	Falso
Falso	Vero	Vero
Vero	Falso	Vero
Vero	Vero	Vero

Circuito elettrico: il collegamento OR corrisponde a un circuito elettrico in parallelo.

10.5 XOR

XOR = Esclusivo OR = Uno o l'altro

La funzione mostra il seguente programma

```

STATICO PUBBLICO SUB Principale ()
DIM a AS Boolean
DIM b AS Boolean
PER a = da 0 a 1
    PER b = da 0 a 1
        PRINT Str $ (a) & "XOR" & Str $ (b) & "==>" & Str $ (a XOR b)
    IL PROSSIMO
IL PROSSIMO
FINE

```

Tabella della verità:

UN.	B.	A XOR B
Falso	Falso	Falso
Falso	Vero	Vero
Vero	Falso	Vero
Vero	Vero	Falso

10.6 NON

NON inverte il valore di verità.

La funzione mostra il seguente programma

```

STATICO PUBBLICO SUB Principale ()
DIM a AS Boolean
PER a = da 0 a 1
    PRINT Str $ (a) & "==>" & Str $ (NON a)
    IL PROSSIMO
FINE

```

Tabella della verità:

UN.	NON A
Vero	Falso
Falso	Vero

10.7 Operatori bit per bit

10.7.1 AND bit per bit

Per fare ciò, considera il seguente programma:

```

PUBLIC SUB Form_Open ()
DIM x AS intero
DIM y AS intero
PER x = da 1 a 15
    PER y = da 1 a 15
        STAMPA Bin $ (x, 4), Bin $ (y, 4), Bin $ (x AND y, 4)
    IL PROSSIMO
IL PROSSIMO
FINE

```

```
IL PROSSIMO  
IL PROSSIMO  
FINE
```

10.7.2 OR bit per bit

Per fare ciò, considera il seguente programma:

```
PUBLIC SUB Form_Open ()  
DIM x AS intero  
DIM y AS intero  
PER x = da 1 a 15  
    PER y = da 1 a 15  
        STAMPA Raccoglitore $ (x, 4), Raccoglitore $ (y, 4), Raccoglitore $ (x OR y, 4)  
    IL PROSSIMO  
IL PROSSIMO  
FINE
```

10.7.3 XOR bit per bit

Per fare ciò, considera il seguente programma:

```
PUBLIC SUB Form_Open ()  
DIM x AS intero  
DIM y AS intero  
PER x = da 1 a 15  
    PER y = da 1 a 15  
        STAMPA Raccoglitore $ (x, 4), Raccoglitore $ (y, 4), Raccoglitore $ (x XOR y, 4)  
    IL PROSSIMO  
IL PROSSIMO  
FINE
```

10.8 Panoramica delle strutture dei gamberi

Come la maggior parte dei linguaggi di programmazione, Gambas ha varie opzioni per il controllo del processo:

- sequenza
- selezione
- Ripetizione
- Sottoprogrammi

10.8.1 Ordine

Normalmente, una riga di comando dopo l'altra viene elaborata in un programma. Esistono tuttavia delle eccezioni a questo principio: le variabili vengono sempre definite con il comando DIM all'inizio del programma o all'inizio di una subroutine. Inoltre, il

Comando GOTO Possibilità di saltare l'ordine del programma. Nel caso dei sottoprogrammi, l'ordine nella finestra del codice è irrilevante, poiché vi si accede con il loro nome. Tuttavia, dovresti abituarti a un certo ordine. Ad esempio, la subroutine PUBLIC SUB Form_Open () appartiene all'inizio del proprio codice.

```
<istruzione>
<istruzione>
<istruzione>
<...>
```

10.8.2 Selezione

I rami del programma richiedono condizioni e decisioni. Ci sono vari comandi per abilitarlo. Se una condizione non si applica, la riga viene semplicemente saltata e il programma continua con la riga dopo la query della condizione.

Selezione unilaterale

```
SE <condizione> ALLORA <istruzione>
```

Selezione bilaterale

```
SE <condizione> ALLORA
  <istruzione>
ALTRO
  <istruzione>
FINISCI SE
```

Distinzione dei casi

```
SELEZIONA CASE TRUE
CASE <condizione1>
  <istruzione>
CASO <condizione2>
  <istruzione>
CASO <condizione3>
  <istruzione>
CASO ALTRO <istruzione>
FINE SELEZIONA
```

10.8.3 loop

Un comando o una sequenza di comandi viene ripetuta più volte in loop.

Ciclo di conteggio

```
FOR <variabile di esecuzione> = <valore iniziale> TO <valore finale>
  <istruzione>
IL PROSSIMO
```

Fiocco repellente

```
WHILE <condizione di ripetizione>
  DO <istruzione>
```

Fiocco non repellente

```
RIPETERE
  <istruzione>
  FINO A <condizione di terminazione>
```

Ciclo di conteggio per array

```
PER OGNI <Elemento> IN <elenco>
  <istruzione>
  IL PROSSIMO
```

Le strutture elencate possono essere annidate l'una nell'altra. Tuttavia, questo spesso porta a confondere il codice e dovrebbe forse essere sostituito da funzioni nelle subroutine.

10.9 rami

10.9.1 se - allora

La ramificazione più semplice di un programma viene eseguita con il comando if - then

- Se significa CADUTE in tedesco
- quindi significa DANN in tedesco

Il comando If Then rende possibile programmare rami di programma se una certa condizione è soddisfatta. In inglese il comando si chiama If la condizione è vera, eseguire le seguenti operazioni.

Questo si comprende meglio con l'aiuto di un esempio: Emette il giorno della settimana di oggi. Il programma funziona in modalità Terminal:

```
STATICO PUBBLICO SUB
Principale ()
PRINT WeekDay (ora)
STAMPA "Oggi è":
SE WeekDay (ora) = 1 ALLORA STAMPA "Lunedì"
SE WeekDay (ora) = 2 ALLORA STAMPA "Martedì"
SE WeekDay (ora) = 3 ALLORA STAMPA "Mercoledì"
SE WeekDay (ora) = 4 ALLORA STAMPA "Giovedì"
SE WeekDay (ora) = 5 ALLORA STAMPA "Venerdì"
SE WeekDay (ora) = 6 ALLORA STAMPA "Sabato"
SE WeekDay (ora) = 7 ALLORA STAMPA "Domenica"
FINE
```

Il programma utilizza il comando Weekday e il comando Now di Gambas

Tra If e Then puoi anche elencare diverse condizioni e anche con le istruzioni dopo il comando Then puoi mettere in sequenza diverse istruzioni, che poi terminerai con Endif.

```
Se
Condizione 1
Condizione2
Poi
Istruzione 1
Istruzione 2
Finisci se
```

10.9.2 in caso contrario

If then Else consente la programmazione di un'alternativa alla condizione If. Questo è meglio illustrato da un esempio

```
nome fioco come stringa
se nome = maschio allora
    Stampa "A Mr." e nome
altrò
    Stampa "Alla signora" e nome
finisci se
```

Qui, tuttavia, la funzione maschile deve ancora essere definita.

Un altro esempio: è necessario un modulo vuoto per avviare il programma. Genera 100 numeri casuali compresi tra 1 e 99 e controlla se sono divisibili per 2.

```
PUBLIC SUB Form_Open ()
un numero intero AS
x AS intero
Rendi casuale ()
PER x = da 1 a 100
a = Int (Rnd (1.100))
'Genera un numero casuale compreso tra 1 e 99
SE a MOD 2 = 0 ALLORA
'divisibile per 2 ??
    PRINT a & "è divisibile per 2"
ALTRO
    PRINT a & "non è divisibile per 2"
FINISCI SE
IL PROSSIMO
FINE
```

10.9.3 Indirizzo di salto GOTO

Anche il vecchio GOTO è ancora a Gambas. Tuttavia, può essere utilizzato solo in una subroutine. L'esempio seguente mostra come funziona:

Hai bisogno di un'area di testo e di un pulsante di comando per avviare il programma.

Routine di array

```
PUBLIC SUB Form_Open ()  
Larghezza ME = 400  
ME.Height = 300  
ME.Text = "Comando GOTO con etichette come indirizzo di salto"  
Button1.Text = "Disponi numero casuale"  
Button1.Width = 200  
Textarea1.Text = ""  
Textarea1.Width = 300  
Textarea1.Height = 160  
FINE  
Pulsante PUBLIC SUB1_Click ()  
z AS intero  
Rendi casuale ()  
z = Int (Rnd (1.100))  
SE z <10 POI ANDARE A dieci  
SE z <20 POI VAI A z2  
SE z <50 POI VAI A z5  
'PRINT "z> 49, poiché z =" ; z  
textarea1.Text = textarea1.Text & "z> 49, poiché z =" & z & Chr (10)  
RITORNO  
  
dieci:  
    textarea1.Text = textarea1.Text & "z <10, da z =" & z & Chr (10)  
    RITORNO  
  
z2:  
    textarea1.Text = textarea1.Text & "z <20, da z =" & z & Chr (10)  
    RITORNO  
  
z5:  
    textarea1.Text = textarea1.Text & "z <50, da z =" & z & Chr (10)  
    RITORNO  
FINE
```

10.9.4 Seleziona caso

Il comando Seleziona caso consente di scegliere tra diverse alternative.

L'esempio seguente mostra come funziona. Controlla la funzione casuale di un cubo 1000 volte e restituisce il numero di volte che un numero è stato ricevuto dal cubo nella finestra diretta. Hai solo bisogno di un modulo vuoto per avviare il programma.

```
PUBLIC SUB Form_Open ()  
x AS intero  
w AS intero  
un numero intero AS  
b AS intero  
c AS intero  
d AS intero  
e AS intero  
f AS intero  
Rendi casuale ()  
PER x = da 1 a 1000  
    w = Int (Rnd (6) + 1)  
    SELEZIONA CASE TRUE  
        CASO w = 1  
            a = a + 1
```

```

CASO w = 2
    b = b + 1
CASO w = 3
    c = c + 1
CASO w = 4
    d = d + 1
CASO w = 5
    e = e + 1
CASO w = 6
    f = f + 1
FINE SELEZIONA
IL PROSSIMO
STAMPA a, b, c, d, e, f
FINE

```

Il programma può essere leggermente accorciato:

```

PUBLIC SUB Form_Open ()
x AS intero
w AS intero
un numero intero AS
b AS intero
c AS intero
d AS intero
e AS intero
f AS intero
PER x = da 1 a 1000
    w = Int (Rnd (6) + 1)
    SELEZIONA CASO w
        CASO 1
            a = a + 1
        CASO 2
            b = b + 1
        CASO 3
            c = c + 1
        CASO 4
            d = d + 1
        CASO 5
            e = e + 1
        CASO 6
            f = f + 1
    FINE SELEZIONA
    IL PROSSIMO
    STAMPA a, b, c, d, e, f
FINE

```

Per una descrizione precisa della sintassi vedere:

- <http://www.gambasdoc.org/help/lang/select?de>

10.10 loop

I loop consentono di ripetere più volte una determinata parte del programma. Se non sai quante volte ripetere l'istruzione nel ciclo, i cicli sono una buona scelta. Se sai esattamente quanto spesso il ciclo deve essere ripetuto, allora dovresti scegliere il comando For ... Next. Il ciclo for-next utilizza un contatore, il cui valore viene incrementato o decrementato ogni volta che il ciclo viene ripetuto.

10.10.1 Per il ciclo successivo

I cicli consentono di eseguire più volte una determinata parte del programma con un numero variabile. Questo è il motivo per cui il ciclo FOR ... NEXT è noto anche come counter loop.

Il principio del ciclo è simile a questo:

```
PER i = da 1 a 9 FASE 1  
    Stampa il codice del programma  
    IL PROSSIMO
```

La variabile i ha all'inizio il valore 1. Ogni volta che il programma raggiunge la fine del ciclo con il comando NEXT, il programma torna all'inizio del ciclo con FOR. Per ogni nuovo passaggio del ciclo, i viene aumentato di 1 (FASE 1). Se i raggiunge il valore finale del ciclo, il valore to (nell'esempio 9), il ciclo si interrompe e il programma continua dopo il ciclo.

Il seguente programma mostra l'uso di un ciclo For-Next. Tutto ciò di cui hai bisogno è una forma vuota. Fare clic sulla forma in modalità Progettazione e immettere il codice del programma di seguito. Quindi avvia il programma con F5. Ogni volta che raggiungi la forma con il mouse, Hello World viene visualizzato 10 volte.

```
PUBLIC SUB Form_Enter ()  
DIM i AS intero  
PER i = da 1 a 10  
    STAMPA "Hello world"  
    IL PROSSIMO  
FINE
```

Prova una variante ancora più semplice:

```
PUBLIC SUB Form_Enter ()  
DIM i AS intero  
PER i = da 1 a 10  
    STAMPA i  
    IL PROSSIMO  
FINE
```

Prova la seguente variante:

```
PUBLIC SUB Form_Enter ()  
DIM i AS intero  
PER i = da 1 a 10  
    STAMPA i, i*io, io*io*io  
    IL PROSSIMO  
FINE
```

Puoi anche usare i 3 esempi con il comando

```
PUBLIC SUB Form_Open ()
```

si avvia automaticamente.

Per rendere i loop più leggibili, la parte di comando interna del loop è rientrata. Viene eseguito più volte. A differenza di VB, Gambas non menziona il contatore alla fine del ciclo dopo il comando Avanti.

```
Per I = da 1 a 10
    stampa I.
    ' Il prossimo lin VB possibile
    ' Eliminato come superfluo a Gambas. Il prossimo
```

Se fai a meno della grafica e usi Terminalmdous, l'ultimo programma è il seguente:

```
STATICO PUBBLICO SUB Principale ()
DIM i AS intero
PER i = da 1 a 10
    STAMPA i, i*io, io*io*io, i^4, i^5
IL PROSSIMO
FINE
```

Il risultato è simile a questo:

1	1	1	1	1
2	4 °	8 °	16	32
3	9	27	81	243
4 °	16	64	256	1024
5	25th	125	625	3125
6 °	36	216	1296	7776
7 °	49	343	2401	16807
8 °	64	512	4096	32768
9	81	729	6561	59049
10	100	1000	10.000	100.000

L'esempio seguente mostra un ciclo con il comando Step: Sorprendentemente, la virgola non funziona correttamente nell'output di stampa. Prova invece il punto e virgola.

```
PUBLIC SUB Form_Open ()
x AS intero
PER x = da 0 a 100 FASE 5
    STAMPA x,
IL PROSSIMO
FINE
```

Il risultato dovrebbe essere:

```
0 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100
```

Ma non è così! Alcuni altri strani esempi:

Routine di array

```
PUBLIC SUB Form_Open ()  
DIM i AS intero  
PER i = DA 1 A 100 PASSO i  
STAMPA i  
IL PROSSIMO  
FINE
```

```
PUBLIC SUB Form_Open ()  
DIM i AS intero  
DIM x AS intero  
PER x = da 1 a 10  
PER i = DA 1 A 100 PASSO x  
STAMPA i & "";  
IL PROSSIMO  
IL PROSSIMO  
FINE
```

10.10.2 Thu loop

Un ciclo DO LOOP è fondamentalmente un ciclo infinito e deve essere interrotto da qualche condizione nel ciclo.

Esempio 1: ciclo infinito

```
PUBLIC SUB Form_Open ()  
x AS intero  
x = 1  
FARE  
POI STAMPA "x =" & x  
x = x + 1  
CICLO CONTINUO  
FINE
```

Esempio 2: ciclo infinito, apparentemente finito.

```
PUBLIC SUB Form_Open ()  
un numero intero AS  
x AS intero  
Rendi casuale ()  
x = 1  
FARE  
a = Int (Rnd (1.100))  
IF x < a THEN PRINT "x =" & x & "a =" & a  
x = x + 1  
CICLO CONTINUO  
FINE
```

Perché tutti i numeri da 1 a 99 non vengono visualizzati nella finestra diretta? Quando e perché il ciclo finisce effettivamente?

-> Il ciclo non si ferma. Manca la condizione di risoluzione.

10.10.3 Per ogni ciclo

Utilizzato per gli array Vedere Gambas: Arrays²

10.10.4 Loop e ramo combinati

Il codice del programma ha questo aspetto. Hai solo bisogno di un pulsante di comando e inizia con F5.

```
Pulsante PUBLIC SUB1_Click ()
    DIM k AS Integer
    PER k = 1 FINO A 10000
        IF k = 5000 THEN PRINT "Il numero 5000 è stato raggiunto!"
        IL PROSSIMO
    FINE
```

10.11 Ordinamenti

Vedi Gambas: Sorting³

10.12 Generale

Le funzioni sono subroutine che restituiscono un valore (numero, testo, ecc.) Al programma principale.

Al contrario, un valore viene spesso passato dal programma principale alla funzione, dove il valore viene elaborato e viene restituito il risultato.

Le funzioni richiedono per abituarsi all'inizio. Ad un certo punto, tuttavia, sono indispensabili per l'uso efficace di un linguaggio di programmazione, poiché facilitano la creazione del programma e accorciano i programmi se una parte del programma deve essere eseguita più spesso. Quasi tutti i comandi derivati in un linguaggio di programmazione si sono evoluti dalle funzioni. Le procedure sono correlate alle funzioni, solo che non restituiscono un valore.

10.13 Un esempio molto semplice

Definisci una funzione Aggiungi che somma 2 numeri insieme.

I 2 numeri da aggiungere sono indicati semplicemente tra parentesi dopo il nome della funzione.

Per l'esempio hai bisogno di:

- Un modulo

² Capitolo 6.157 a pagina 360

³ Capitolo 10.16.7 a pagina 407

Routine di array

- Un pulsante di

comando Il codice è:

```
'Programma principale
Pulsante PUBLIC SUB1_Click ()
    Message.Info (Aggiungi (6,10))
    'Chiamata di funzione e restituzione del valore
    FINE
    'Funziona come una subroutine
    FUNZIONE PUBBLICA Aggiungi (a AS Integer, b AS Integer) AS Integer
        RETURN a + b
    FINE
```

Questa funzione dipende dai valori a e b. I valori aeb possono essere scelti arbitrariamente.

10.14 Esempi di funzioni:

- Ora (viene restituita l'ora di sistema)
- Data (viene restituita la data)
- Radice quadrata (viene inserito un numero, ritorna la radice quadrata) a * a diventa sqr (a)
- Converti il colore VGA in RGB (viene inserito uno dei 16 colori VGA, torna il valore RGB)

10.14.1 Esempio TIME: (nessun parametro è incluso nella funzione, viene comunque restituito un valore)

Un buon esempio di una semplice funzione è il comando Time. Questa funzione fornisce l'ora di sistema corrente. Puoi testare questa funzione con il seguente comando:

```
Tempo di stampa
```

Il seguente piccolo programma senza grafica mostra l'ora nella finestra diretta:

```
STATICO PUBBLICO SUB Principale ()
    Tempo di STAMPA
    FINE
```

Se vuoi avere l'ora aggiornata ancora e ancora, devi costruire un ciclo infinito.

Il programma quindi si presenta così:

```
STATICO PUBBLICO SUB Principale ()
DIM i AS intero
i = 1
FARE MENTRE i <= 10
    Tempo di STAMPA
```

```
CICLO CONTINUO
FINE
```

Un esempio simile per l'output grafico aggiorna continuamente l'output dell'ora. Per fare ciò, usi il timer, l'elemento di controllo del timer. Questo inizialmente non ha nulla a che fare con l'ora corrente, ma attiva un evento dopo un certo tempo.

Il seguente piccolo programma di esempio combina il timer e la funzione dell'ora in un'informazione sull'ora costantemente aggiornata, come puoi trovarla sul tuo PC in basso a destra.

Inizia un nuovo progetto. Chiamalo tempo trascorso. Quindi posiziona una casella di testo sulla tua forma. È inoltre necessario il controllo del timer con l'icona dell'orologio. Ottieni entrambi con F6 e un doppio clic sul simbolo dalla casella degli strumenti. Assicurati che il timer con la proprietà Enabled debba essere prima attivato. Puoi impostarlo nelle proprietà del timer in fase di progettazione o puoi farlo nel codice del programma con la riga

```
timer1.Enabled = TRUE
```

raggiungere.

```
PUBLIC SUB Timer1_Timer ()
timer1.Enabled = TRUE
textbox1.Text = Ora
FINE
```

Il programma mostra l'ora corrente nella casella di testo. Viene aggiornato ogni secondo (= 1000 millisecondi).

10.14.2 Esempio UCASE: (un parametro è incluso nella funzione e viene restituito un valore convertito)

La funzione UCASE (= capitalizzazione) è una funzione semplice che richiede solo il trasferimento di una stringa. Per trasferire qualcosa di simile (variabili, costanti o stringhe), queste devono essere racchiuse tra parentesi e apparire dopo il nome della funzione.

Crea una forma con una casella di testo e un pulsante di comando. Scrivi per quello Pulsante di comando il codice seguente:

```
Pulsante PUBLIC SUB1_Click ()
Textbox1.Text = UCASE (Textbox1.text)
FINE
```

Avvia il programma, digita un testo qualsiasi nel campo di testo e fai clic sul pulsante di comando. Di conseguenza, il testo è in lettere maiuscole nella casella di testo.

Il seguente codice sorprendentemente non funziona:

```
PUBLIC SUB TextBox1_KeyRelease ()  
IF Key.return THEN UCASE (Textbox1.text)  
FINE
```

Il testo non è in maiuscolo nella versione 0.99.

Sebbene il codice seguente funzioni per il metodo KeyRelease. Secondo me, c'è ancora un bug nel programma.

```
PUBLIC SUB TextBox1_KeyRelease ()  
IF Key.return THEN textbox1.visible = FALSE  
FINE
```

La casella di testo diventa invisibile.

Se vuoi guardare di nuovo insieme le funzioni UCASE e LCASE (= minuscolo) in un semplice programma, prova il seguente codice senza grafica:

```
STATICO PUBBLICO SUB Principale ()  
DIM una stringa AS  
a = "test"  
STAMPA a  
STAMPA LCASE (a)  
STAMPA ASASE (a)  
FINE
```

(Vedi anche nella sezione Primi passi, Terzo programma: Programmi nella finestra del terminale senza grafica)

10.14.3 Esempio 3: (diversi parametri)

Se vengono passati più valori, devono essere separati l'uno dall'altro da virgolette tra parentesi dopo il nome della funzione.

Nell'esempio 2, la funzione ha restituito un testo diverso a seconda della stringa che le è stata passata. Altre funzioni necessitano di parametri aggiuntivi per dirgli come gestire quei parametri. Ad esempio, la funzione Mid \$ (= ritaglia il testo dal centro)

- un parametro per il testo utilizzato,
- un parametro per fissare il punto di partenza (misurato in numero di caratteri),
- e poi un terzo parametro facoltativo per determinare quanti caratteri devono essere restituiti.

Nell'esempio seguente vengono riassunte tutte le possibilità di elaborazione delle stringhe = manipolazione delle stringhe che Gambas conosce. È inclusa anche la funzione Mid \$. Tutto ciò di cui hai bisogno è un modulo vuoto per avviare il programma.

```

PUBLIC SUB Form_Open ()
Parola DIM $ AS stringa
Word $ = "piccolo confine di stato"
STAMPA Len (parola $)
STAMPA Sinistra $ (parola $, 7)
STAMPA Destra $ (parola $, 6)
PRINT Mid $ (parola $, 3.5)
PRINT Mid $ (word $, 6)
PRINT Instr (parola $, "eng")
PRINT Instr (parola $, "pomodoro")
PRINT Instr (parola $, "eng", 12)
PRINT Instr (parola $, "e", 11)
FINE

```

Risultato

- 18 (lunghezza della parola)
- Più piccolo (le prime 7 lettere)
- limite (le ultime 6 lettere)
- una volta (5 lettere centrali che iniziano con la terza)
- confine di stato (lettere che iniziano con la sesta lettera)
- 11 ("stretto" è all'undicesima posizione nella parola)
- 0 (il pomodoro non è stato trovato)
- 0 (non trova stretto se la ricerca è da effettuarsi solo dalla 12a posizione)
- 11 ("e" si trova all'undicesima posizione se la ricerca deve essere eseguita solo dall'undicesima posizione.)

ATTENZIONE: In VB puoi usare il comando Mid \$ () per ritagliare una sequenza di lettere da un testo e inserirne un altro. Non funziona a Gambas.

Esempio:

```

rem VB:
MyString = "Il cane sta saltando"
Mid $ (MyString, 5, 4) = "Ball"

```

Result MyString = "La volpe salta".

Questo non funziona in Gambas, ma puoi fare quanto segue:

```

MyString = "Il cane sta saltando"
MyString = Left $ (MyString, 4) & "Ball" & Mid $ (MyString, 9)

```

Un programma completo per questo (senza grafica) si presenta così:

```

STATICO PUBBLICO SUB Principale ()
DIM MyString AS String
MyString = "Il cane sta saltando"
PRINT MyString
Possibile in VB: Mid $ (MyString, 5, 4) = "Ball" non in Gambas
MyString = Left $ (MyString, 4) & "Ball" & Mid $ (MyString, 9)
PRINT MyString
FINE

```

10.14.4 Comandi di manipolazione delle stringhe = manipolazione delle stringhe

&	Incollare le corde insieme
InStr ()	Riprendi la posizione di un carattere o di una sottostringa
sinistra ()	Separare le sottostringhe partendo da sinistra
Piace	Confronto di 2 stringhe o sottostringhe
LTrim ()	Tagliare gli spazi all'inizio di una stringa
metà ()	Taglia una sottostringa dal centro
destra ()	Separare le sottostringhe partendo da destra
RTrim ()	Tagliare gli spazi alla fine di una stringa
Diviso ()	Divisione di una stringa in singole parole
Taglia ()	Taglia gli spazi all'inizio e alla fine di una stringa
CStr ()	Converte un'espressione in una stringa.
Str ()	Converte un'espressione in una stringa. (Funzione inversa o opposta da val ())
val ()	Converte una stringa in un numero, un numero in virgola mobile o qualcos'altro.

10.15 Il programma funziona da soli

Comprendi correttamente le funzioni solo quando le hai programmate da solo. Quindi puoi anche vedere la sua importanza pratica. Di solito puoi trasferirli ad altri programmi completamente invariati.

10.15.1 alto 3

Di seguito va programmata una funzione che porti un numero alla potenza di 3. A tale scopo, avviare un nuovo progetto grafico Gambas: posizionare 2 caselle di testo e un pulsante di comando sul modulo. È possibile liberare le caselle di testo dal loro testo utilizzando la finestra delle proprietà F4. Immettere il codice seguente nel pulsante di comando:

```

Pulsante PUBLIC SUB1_Click ()
DIM b AS Intero
b = Val (Textbox1.text)
textbox2.Text = Cstr (Su (b))
FINE
'-La funzione inizia qui -----
FUNZIONE PUBBLICA High (a AS Integer) AS Integer
a = a^3
RITORNO a
FINE

```

Non dimenticare il comando di ritorno nella tua funzione. Altrimenti non ti verrà restituito nulla.

La programmazione delle funzioni in Gambas è leggermente diversa dalla programmazione in VB. In Gambas non è possibile chiamare il nome della funzione nella funzione, in VB è possibile.

10.15.2 Terza radice (alla potenza di 1/3)

Di seguito deve essere programmata una funzione che prende la terza radice di un numero. A tale scopo, avviare un nuovo progetto grafico Gambas: posizionare 2 caselle di testo e un pulsante di comando sul modulo. È possibile liberare le caselle di testo dal loro testo utilizzando la finestra delle proprietà F4. Immettere il codice seguente nel pulsante di comando:

```
Pulsante PUBLIC SUB1_Click ()  
DIM b AS Intero  
b = Val (Textbox1.text)  
textbox2.Text = Cstr (root (b))  
FINE  
'- Qui è dove inizia la funzione -  
FUNZIONE PUBBLICA root (a AS Float) AS Float  
a = a^ (1/3)  
RITORNO a  
FINE
```

Se ora inserisci un numero positivo nella casella di testo e fai clic sul pulsante di comando, da esso verrà calcolata la terza radice.

Perché la variabile funzione è stata definita qui come numero float. Prova la stessa cosa con una variabile intera. Non dimenticare il comando di ritorno nella tua funzione. Altrimenti non ti verrà restituito nulla.

10.15.3 Incollare testi insieme a una funzione

Il prossimo programma di esempio mostra come chiamare una funzione per incollare due parti di testo.

Hai bisogno di una casella di testo e 2 pulsanti di comando per avviare il programma.

```
' File di classe Gambas PUBLIC  
SUB Button1_Click () DIM b AS  
String  
  
    Textbox1.Text = word (b) & "Franz"  
FINE  
'-----  
PUBLIC SUB Button2_Click ()  
    Textbox1.Text = word & "Fritz"  
    'Questa chiamata di funzione restituisce un messaggio di errore perché il  
parametro previsto non è stato trasferito.  
FINE  
'-----  
FUNZIONE PUBBLICA parola (a AS String) AS String a =  
    "Hello"  
    RITORNO a  
FINE
```

Se si fa clic sul pulsante 1, la funzione viene chiamata correttamente. Se si fa clic sul pulsante 2, viene visualizzato un messaggio di errore. Capisci perché? Come potresti cambiare il codice della funzione in modo che anche il codice sotto il pulsante 2 funzioni correttamente?

Soluzione:


```
"File di classe Gambas
Pulsante PUBLIC SUB1_Click ()
    DIM b AS stringa
    Textbox1.Text = word (b) & "Franz"
FINE
PUBLIC SUB Button2_Click ()
    Textbox1.Text = words () & "Fritz"
FINE
FUNZIONE PUBBLICA parola (una stringa AS) stringa AS
    a = "Ciao"
    RITORNO a
FINE
FUNZIONE PUBBLICA parole () AS String
    una stringa AS
    a = "Ciao"
    RITORNO a
FINE
```

10.15.4 Facoltà

Nell'esempio seguente deve essere calcolato il fattoriale di un numero.

Il fattoriale di un numero (rappresentazione matematica: " $n!$ ", Si dice: "n fattoriale") è il prodotto dei numeri positivi naturali, cioè interi, da 1 a n.

Simbolo: $!$ = Punto esclamativo

Si applica quanto segue:

- $0! = 1$
- $1! = 1$
- $2! = 1 * 2 = 2$
- $3! = 1 * 2 * 3 = 6$
- $4! = 1 * 2 * 3 * 4 = 24$
- $5! = 1 * 2 * 3 * 4 * 5 = 120$
- $6! = 1 * 2 * 3 * 4 * 5 * 6 = 720$
-
- $n! = n * (n - 1)!$

Il numero più alto, il cui fattoriale può essere calcolato in Gambas Basic senza trucchi per emulare un intervallo di valori più ampio, è 170. Nel programma sottostante, 13 è il limite superiore. Sono necessari un pulsante di comando e due caselle di testo per il programma.

```
"File di classe Gambas
Pulsante PUBLIC SUB1_Click ()
    DIM un numero intero AS
    a = Val (textbox1.text)
    ' textbox2.Text = Faculty (Val (textbox1.text)) textbox2.Text =
    Faculty (a)
FINE
'--- La funzione inizia qui
-----
FUNZIONE Faculty (F AS Integer) AS Integer
DIM k AS Integer
DIM z AS Integer
```

```

SE F = 0 ALLORA F = 1
z = F - 1
PER k = 1 A z
  F = F*K
IL PROSSIMO
RITORNO F
FINE

```

10.15.5 Coefficiente binomiale

Se vuoi aumentare una somma alla potenza, puoi usare il coefficiente binomiale per questo.

Esempio:

$$\begin{aligned}
 (a + b)^2 &= 1*a^2 + 2*un*b + 1*b^2 \\
 (a + b)^3 &= 1*a^3 + 3*a^2*b + 3*un*b^2 + 1*b^3 \\
 (a + b)^4 &= 1*a^4 + 4*a^3*b + 6*a^2*b^2 + 4*un*b^3 + 1*b^4
 \end{aligned}$$

I numeri davanti ai vari termini a e b sono chiamati coefficienti e obbediscono alle regole del triangolo di Pascal. Per favore, riferisci <http://www.madeeasy.de/2/pascal.htm>

La regola del triangolo è semplice: ogni numero (tranne l'1 sul bordo) si ottiene sommando i due numeri sopra di esso (vedi esempio).

| immagine⁴

Puoi anche calcolare i numeri con la formula per il coefficiente binomiale:

| immagine⁵

Questa formula può essere ritrovata ripetutamente anche in statistica e calcolo combinatorio, motivo per cui dovrebbe essere programmata di seguito.

Riscriviamo prima la formula dalla rappresentazione grafica sopra in una rappresentazione testuale:

$$N \text{ su } k = N! / K! / (Nk)!$$

$$\text{dove } N! = 1*2*3*4 \dots * (N-1)*N = N \text{ fattoriale}$$

$N > 0$ e $k > 0$ e $N > k$ devono sempre essere applicati.

Sono necessarie 3 caselle di testo per inserire e visualizzare i valori e una per avviare il calcolo Pulsante di comando. L'intera cosa assomiglia a questo:

⁴ <http://www.madeeasy.de/2/pascal.gif>
⁵ <http://www.madeeasy.de/2/nueberk.gif>

Routine di array

Se vuoi programmare la formula, puoi farlo senza richiamare una funzione, questo è mostrato dal seguente codice di programma:

```
"File di classe Gambas
Pulsante PUBLIC SUB1_Click ()
DIM N AS Inter
DIM k AS Integer
DIM F AS Inter
DIM x AS intero
DIM nf AS Integer 'N faculty
DIM nmk AS Numero intero n meno k fattoriale
DIM kf AS Integer 'k faculty
DIM nuk AS Integer
N = Val (textbox1.text)
k = Val (textbox2.text)
F = 1
PER x = 1 A N
    F = F * X
    IL PROSSIMO
nf = F
F = 1
PER x = 1 TO k
    F = F * X
    IL PROSSIMO
kf = F
F = 1
PER x = 1 TO (n- k)
    F = F * X
    IL PROSSIMO
nmk = F
nuk = nf / nmk / kf
textbox3.Text = nuk
FINE
```

Ora prova i valori per $(a + b)^4$. Qui $N = 4$ e k aumenta da 0 a 4. Metti questi valori 4 e 0, 4 e 1, 4 e 2, 4 e 3, 4 e 4 nel tuo programma e otterrai i coefficienti dell'equazione binomiale e la linea corrispondente nel triangolo Pascal.

La programmazione funziona in modo più elegante se si passa il calcolo fattoriale a una funzione.

10.15.6 Triangolo di Pascal

Se affidi anche il coefficiente binomiale n su k a una funzione, puoi calcolare tutti i valori del triangolo di Pascal, ad esempio dalla riga 1 alla riga 20, in un doppio ciclo.

Poiché è difficile produrre in forma grafica, può essere programmato in modalità terminale:

Il programma ha questo aspetto

```
Da riga = 1 a riga = 12
Calcola N su K
dove N = riga ek cresce da 0 a N.
```

Output come stampa riga per riga separata da virgole Riga
successiva

Il programma è rifinito nel ciclo interno

```
Da riga = 1 a riga = 20
Per K = 0 To N = line
Calcola N su K
Stampa N su k;
Avanti k
Print 'inserisce un avanzamento riga
Riga successiva
```

Il programma completo si presenta così:

```
' File classe Gambas STATIC
PUBLIC SUB Main () DIM line
AS Integer DIM N AS Integer

DIM k AS Interger DIM
F AS Interger DIM x AS
Interger
DIM nf AS Integer 'N faculty
DIM nmk AS Integer 'n meno k facoltà DIM kf AS
Integer' k facoltà
DIM nuk AS Interger FOR
line = 1 TO 12 N = line

PER k = 0 FINO A
NF = 1
PER x = 1 FINO A
    NF = F * X
    SUCCE
    SSIVO
    nf = FF
    = 1
    PER x = 1 TO k F =
        F * X
        SUCCE
        SSIVO
        kf = FF
        = 1
        PER x = 1 TO (n- k) F = F
            * X
            SUCCES
            SIVO
            nmk = F
            nuk = nf / nmk / kf PRINT
            nuk & ",";
    STAM
    PA
    SUCC
    ESSIV
    A
FINE
SUCC
ESSIV
A
```

Il risultato nella finestra diretta è simile a questo.

```
1 1
1 2 1
1 3 3 1
1 4 ° 6 ° 4 ° 1
1 5 10 10 5 1
```


Routine di array

```
1 6° 15° 20th 15° 6° 1
1 7° 21 35 35 21 7° 1
8
1 8° 28 56 70 56 28 ° 1
1 9 36 84 126 126 84 36 9 1
1 10 45 120 210 252 210 120 45 10 1
1 11 55 165 330 462 462 330 165 55 11 1
12
1 ° 66 220 495 792 924 792 495 220 66 12 ° 1
```

Un triangolo Pascal storto, ma riconoscibile.

Il programma può essere notevolmente semplificato se si può calcolare n su k e il fattoriale come funzione.

10.16 Costanti predefinite

gb.zero	Valore nullo
gb.Boolean	Valore booleano
gb.Byte	Numero intero byte
gb.breve	Numero intero breve
gb.Integer	Numero intero
gb.Float	Numero in virgola mobile
gb.Date	Valore di data e ora
gb.String	Stringa di caratteri
gb.Variant	Variante
gb.Object	Riferimento oggetto

10.16.1 Tipi di file

gb.File	File regolare
gb.directory	Directory
gb.dispositivo	File speciale per un dispositivo
gb.Pipe	Pipa chiamata
gb.socket	File speciale per un socket
gb.link	Collegamento simbolico

10.16.2 Costanti stringa

gb.NewLine	Carattere di nuova riga. Equivalente a Chr \$ (10)
gb.Tab	Carattere di tabulazione. Equivalente a Chr \$ (9)

10.16.3 Ordinamento

gb.Binary	Ordinamento binario
gb.Case	Ordinamento senza distinzione tra maiuscole e minuscole

gb. lungo	Ordinamento basato sulla lingua
-----------	---------------------------------

10.16.4 Giorni della settimana

gb. lunedì	Lunedì
gb. martedì	martedì
gb. mercoledì	mercoledì
gb. giovedì	giovedì
gb. venerdì	Venerdì
gb. sabato	Sabato
gb. domenica	Domenica

10.16.5 Formati numerici preimpostati

gb.GeneralNumber	Scrivi un numero con dodici cifre decimali. Usa scientifico per- mat se il suo valore assoluto è inferiore a 10-4 o superiore a 107.
gb.Fixed	Equivalente a "0,00"
gb.Percent	Equivalente a "###%"
gb. scientifico	Scrivi un numero con il suo esponente e diciotto cifre decimali.

10.16.6 Formati di data e ora preimpostati

gb.GeneralDate	Scrivere una data solo se il valore di data e ora ha una parte relativa alla data e scrivi un'ora solo se ha una parte di data.
gb.LongDate	Formato data lungo.
gb.MediumDate	Formato medio della data.
gb.ShortDate	Formato data breve.
gb.LongTime	Formato lungo.
gb.MediumTime	Formato medio orario.
gb.ShortTime	Formato breve.

10.16.7 Diversi formati

gb. standard	Utilizza gb.GeneralNumber per la formattazione dei numeri e gb.GeneralDate per la formattazione di date e ore.
--------------	--

I metodi di ordinamento vengono utilizzati in database, tabelle, array e molto altro. Di seguito vengono descritti alcuni semplici metodi di ordinamento.

10.17 Ripplesort

Hanno una sequenza di numeri 13, 12, 15, 11, 14. Si desidera disporre questi numeri in ordine crescente in modo che il numero più piccolo sia all'inizio e il più grande alla fine.

Il seguente programma mostra come funziona. Usa un array (vedi Gambas: Arrays⁶) e diverse variabili ausiliarie. Questo array viene ordinato più volte finché l'ordine non è corretto.

È necessario un modulo e un pulsante di comando per avviare il programma.

```
' File di classe Gambas ar
AS Integer [] PUBLIC SUB
Form_Open ()
'Quando il modulo viene aperto, la matrice viene riempita con tutti zeri.

x AS intero
ar = NEW Integer []
ME.Text = "Ordina" FOR x
= 0 TO 4 ar.Add (Str (0))

FINE
SUC
CES
SIVA
PUBLIC SUB Button1_Click () c
AS Integer
j AS Interger n AS Interger y
AS Interger elemento AS
Interger y = 0

n = numero ar
[0] = 13 ar [1] =
12 ar [2] = 15 ar
[3] = 11 ar [4] =
14

RIPETI c
= 0
PER j = 0 TO n - 2
    STAMPA j, y, ar [0], ar [1], ar [2], ar [3], ar [4] SE ar
        [j]> ar [j + 1] ALLORA
            y = ar [j]
            ar [j] = ar [j + 1] ar [j +
                1] = y
            c = 1
        ENDIF
NEXT UNTIL
c = 0
FINE
```

L'output nella finestra diretta è simile a questo:

```
STAMPA j, y, ar [0], ar [1], ar [2], ar [3], ar [4]
```

0	0	1	2	3	4 °	5
---	---	---	---	---	-----	---

0	0	13 °	12 °	15 °	11	14th
1	13 °	12 °	13 °	15 °	11	14th
2	13 °	12 °	13 °	15 °	11	14th
3	15 °	12 °	13 °	11	15 °	14th
0	15 °	12 °	13 °	11	14th	15 °
1	15 °	12 °	13 °	11	14th	15 °
2	13 °	12 °	11	13 °	14th	15 °
3	13 °	12 °	11	13 °	14th	15 °
0	13 °	12 °	11	13 °	14th	15 °
1	12 °	11	12 °	13 °	14th	15 °
2	12 °	11	12 °	13 °	14th	15 °
3	12 °	11	12 °	13 °	14th	15 °
0	12 °	11	12 °	13 °	14th	15 °
1	12 °	11	12 °	13 °	14th	15 °
2	12 °	11	12 °	13 °	14th	15 °
3	12 °	11	12 °	13 °	14th	15 °

Il processo di ordinamento è nella ripetizione fino al ciclo. C'è un ciclo for next in esso e c'è un altro interscambio con if then. È un po 'difficile da capire. Con l'output dettagliato nella finestra diretta puoi seguire esattamente l'ordinamento. È possibile modificare la matrice e inserire altri numeri. Quindi puoi guardare di nuovo il processo di ordinamento. Il kernel è uno scambio di due numeri vicini se il numero a sinistra è maggiore di quello a destra. Il resto della linea rimane. Quindi viene ripetuto il ciclo For Next, in cui non accade nulla perché la posizione ar [1] dell'array è ora 13 e 13 <15. (l'array inizia con il numero 0 e termina con n-1 se ha n elementi. La prossima volta che viene eseguito il ciclo For Next, la posizione ar [2] = 15 viene confrontata con ar [3] = 11. Dal momento che 15> 11 viene scambiato. Quindi il ciclo for next termina per la prima volta, poiché j = 3 = n-2 e il ciclo di ripetizione viene eseguito una seconda volta, perché il parametro ausiliario c indica che è avvenuto uno scambio. E così va avanti all'infinito fino a c = 0, cioè non è stato necessario alcun scambio.

10.17.1 Ripplesort con numeri casuali

Il seguente programma è derivato da quanto sopra. I primi 5 numeri compresi tra 1 e 5 vengono generati in modo casuale. Questi vengono quindi ordinati. Hai bisogno di un'area di testo e 2 pulsanti di comando per avviare il programma.

```
' File di classe Gambas ar
AS Integer [] PUBLIC SUB
Form_Open ()
'Quando il modulo viene aperto, la matrice viene riempita con tutti zeri.

x AS intero
ar = NUOVO Intero []
ME.Text = "Ordina" PER x =
da 0 a 4
    ar.Add (Str (0))
FINE
SUC
CES
SIVA
PUBLIC SUB Button1_Click () x
AS Integer
r AS Integer element
AS Integer txt AS String

z AS Integer txt
= ""
```

Routine di array

```
'l'array è completamente impostato a zero
PER x = da 0 a 4
    ar [x] = 0
IL PROSSIMO
Rendi casuale ()
PER x = da 0 a 4
    r = Int (Rnd (1,6))
    ar [x] = r
IL PROSSIMO
x = 0
PER OGNI elemento IN ar
PRINT x, element
txt = txt & x & "" & element & Chr (10)
    x = x + 1
IL PROSSIMO
textarea1.Text = txt
FINE
PUBLIC SUB Button2_Click ()
c AS intero
j AS intero
n AS intero
y AS intero
elemento AS Integer
txt come stringa
y = 0
n = numero ar
RIPETERE
c = 0
PER j = 0 TO n - 2
    STAMPA j, y, ar [0], ar [1], ar [2], ar [3], ar [4]
    SE ar [j]> ar [j + 1] ALLORA
        y = ar [j]
        ar [j] = ar [j + 1]
        ar [j + 1] = y
        c = 1
    FINISCI SE
IL PROSSIMO
FINO A c = 0
PER OGNI elemento IN ar
txt = txt & elemento & Chr (10)
IL PROSSIMO
textarea1.Text = textarea1.Text & txt
FINE
```

10.17.2 Ordina in un elenco di numeri di qualsiasi lunghezza

Nell'esempio seguente l'algoritmo ripplesort viene utilizzato per ordinare un elenco di valori di qualsiasi lunghezza. Sono necessarie 2 aree di testo e 2 pulsanti di comando per avviare il programma.

Codice:

```
"File di classe Gambas
Pulsante PUBLIC SUB1_Click ()
c AS intero
j AS intero
n AS intero
y AS variante
Elenco DIM come stringa []
elemento AS String
txt come stringa
```

```

testo come stringa
text = Textarea1.Text
list = Split (text, Chr (10))
y = 0
n = liste.length
RIPETERE
c = 0
PER j = 0 TO n - 2
  'PRINT j, y, list [0], ar [1], ar [2], ar [3], ar [4]
  IF Val (list [j])> Val (list [j + 1]) THEN
    y = Val (elenco [j])
    list [j] = list [j + 1]
    elenco [j + 1] = Str (y)
    c = 1
  FINISCI SE
  IL PROSSIMO
FINO A c = 0
PER OGNI elemento IN lista
txt = txt & Str (elemento) & Chr (10)
IL PROSSIMO
'PRINT txt
textarea2.Text = ""
textarea2.Text = txt
FINE
PUBLIC SUB Button2_Click ()
'Elenco di esempio
textarea1.Text = "114,3"
textarea1.Text = textarea1.Text & Chr (10) & "135,7"
textarea1.Text = textarea1.Text & Chr (10) & "104,8"
textarea1.Text = textarea1.Text & Chr (10) & "118,5"
textarea1.Text = textarea1.Text & Chr (10) & "125,7"
textarea1.Text = textarea1.Text & Chr (10) & "121,4"
textarea1.Text = textarea1.Text & Chr (10) & "122,4"
textarea1.Text = textarea1.Text & Chr (10) & "96,8"
textarea1.Text = textarea1.Text & Chr (10) & "118,9"
textarea1.Text = textarea1.Text & Chr (10) & "120"
textarea1.Text = textarea1.Text & Chr (10) & "112,2"
textarea1.Text = textarea1.Text & Chr (10) & "127,9"
textarea1.Text = textarea1.Text & Chr (10) & "122,8"
textarea1.Text = textarea1.Text & Chr (10) & "128,9"
textarea1.Text = textarea1.Text & Chr (10) & "120,3"
FINE

```


11 Eventi degli elementi di controllo

Ogni controllo ha diversi eventi associati ad esso. L'evento più comune e di solito l'impostazione predefinita è l'evento clic. Viene utilizzato quando si desidera che accada qualcosa quando si fa clic una volta sul controllo. Anche Invio o l'evento di pressione del tasto viene utilizzato relativamente frequentemente. Molti altri eventi, invece, verranno usati raramente. La differenza esatta tra gli eventi e i metodi di un elemento di controllo probabilmente non è definita molto chiaramente.

Esempio tipico con un pulsante di comando nel modulo:

```
Pulsante PUBLIC SUB1_Click ()  
    Message.Info ("Hello World")  
FINE
```

L'evento è il semplice clic. La conseguenza innescata è l'output di un messaggio. (Un "clic" con il mouse consiste nel premere E rilasciare il pulsante del mouse, l'evento è completo solo quando lo rilasci, se vuoi una reazione quando lo premi, seleziona l'evento "MouseDown". Un "clic" con la tastiera attivata quando il pulsante è attivo e la barra spaziatrice viene premuta e rilasciata.)

Un altro evento "Click" con la tastiera è l'accesso ai pulsanti di comando con una combinazione di tasti ("Alt" e un carattere dell'identificatore del pulsante. Per fare ciò, nella finestra delle proprietà davanti alla lettera, che con il "Alt" Il tasto per attivare l'evento "Click" deve essere utilizzato, inserire una "&". Questa lettera viene quindi sottolineata mentre il programma è in esecuzione. Il pulsante corrispondente non deve avere il focus per attivare l'evento.

L'ulteriore operazione del pulsante di comando con la tastiera:

```
PUBLIC SUB Button1_KeyPress ()  
    IF key.Code = key.Space OR key.Code = key.Return OR key.code =  
key.Enter THEN  
        Message.Info ("Hello World")  
    FINISCI SE  
FINE
```

Il messaggio viene emesso quando il pulsante di comando ha il focus e uno dei tasti "Spazio", "Invio" o "Invio" è stato premuto ("Invio" è il tasto sul blocco alfanumerico, "Invio" è sul blocco numerico della tastiera).

Se l'evento 'Click' viene interrogato contemporaneamente al pulsante, il frammento di codice 'key.Code = key.Space OR' deve essere omesso, altrimenti verranno chiamate 2 subroutine.

Eventi di controllo

È possibile scoprire tutti i possibili eventi che possono essere attivati con un elemento di controllo facendo clic con il pulsante destro del mouse sull'elemento di controllo e selezionando la riga Evento.

L'elemento di controllo del pulsante (pulsante di comando) ha, ad esempio, i seguenti programmabili Eventi:

```
ClickDblClickDragDragMoveDropEnterGotFocusKeyPress  
KeyRelease  
LeaveLostFocusMenuMouseDownMouseMoveMouseUpMouseWheel
```

Prova il seguente evento:

```
PUBLIC SUB Button1_Enter ()  
    Message.Info ("Funziona quando il mouse è sul pulsante  
    In arrivo ")  
    FINE
```

O la seguente variante:

```
PUBLIC SUB Button1_Leave ()  
    Message.Info ("Funziona solo quando il mouse lascia il pulsante")  
    FINE
```

Quando si combinano gli eventi, gli eventi a volte si ostacolano a vicenda:

Prova il seguente programma, ad esempio:

```
PUBLIC SUB Button1_Leave ()  
    ME.Text = "Funziona solo quando il mouse lascia il pulsante."  
    FINE  
Pulsante PUBLIC SUB1_Click ()  
    Message.Info ("Funziona quando fai clic su")  
    FINE
```

Se fai clic sul pulsante di comando qui e non lo lasci, il titolo del modulo verrà comunque modificato.

12 Programma eventi con EVENT, STOP EVENT, RAISE

Programmare eventi da soli raramente ha senso. Ad esempio, è possibile controllare l'immissione della chiave di una casella di testo se si desidera impedire l'immissione di qualcosa di diverso dai numeri. Questa è una sorta di gestione degli errori.

Esempio: hai bisogno di una casella di testo nel modulo.

```
"La mia casella di testo elabora solo cifre.  
PUBLIC SUB TextBox1_KeyPress ()  
    IF Instr ("0123456789", Key.Text) = 0 THEN  
        STOP EVENTO  
    FINISCI SE  
END SUB
```

Se hai inserito qualcosa di diverso dalle cifre, la casella di testo rifiuta di accettarlo. I numeri, invece, vengono accettati e visualizzati nella casella di testo.

Vedi anche Gambas: Modifica del testo # Inserimento del testo¹

Nell'esempio seguente, un evento viene attivato dopo il 4 ° clic del pulsante.

Hai bisogno di una forma con un pulsante di comando su di essa. Hai anche bisogno di una nuova classe, Class1. Puoi ottenerlo facendo clic con il pulsante destro del mouse sulla riga Classe nel menu del progetto e selezionando Nuova classe. Per semplicità, dovresti mantenere il nome suggerito Class1:

Il codice seguente è stato utilizzato per il modulo:

```
PUBLIC SUB Form_Open ()  
    MyClass = NEW Class1 AS "Osserva"  
FINE  
PRIVATO MyClass AS Class1  
Pulsante PUBLIC SUB1_Click ()  
    MyClass.testen ()  
FINE  
PUBLIC SUB watch_event (Data AS String)  
    Message.Info (Dati)  
FINE
```

Il codice seguente è stato utilizzato per la classe:

¹ Capitolo 6.30 a pagina 237

```
Contatore PRIVATO AS intero
Evento EVENT (Data AS String)
Prova PUBLIC SUB ()
    contatore = contatore + 1
    SE contatore> 3 ALLORA
        RAISE evento ("pulsante premuto> 3 volte")
        contatore = 0
    FINISCI SE
FINE
```

12.1 Cattura gli errori con Catch

Il comando CATCH contrassegna l'inizio di una routine di errore all'interno di una funzione o procedura.

La parte catch viene sempre eseguita se si verifica un errore durante l'esecuzione della funzione.

L'errore può verificarsi all'interno della procedura stessa o in un'altra funzione chiamata dall'interno della procedura. Tuttavia, ciò si applica solo a condizione che la funzione chiamata stessa non contenga un comando CATCH.

Più profonda è la parte di cattura, maggiore è la priorità che ha.

Se si verifica un errore nella parte di cattura stessa, non verrà corretto. La parte di cattura non si protegge.

Se una parte Finalmente appare nella funzione, deve precedere la parte Cattura.

Vedi FINALMENTE.

12.1.1 Esempio 1

Il programma salva il contenuto dell'area di testo come file. La routine di errore intercetta il comando di annullamento nella finestra di dialogo di selezione dei file.

Hai bisogno di un'area di testo e di un pulsante di comando sul modulo per avviare il programma.

12.1.2 Il codice:

```
PUBLIC SUB Form_Open ()
    Textarea1.Text = "Questo è un test"
FINE
Pulsante PUBLIC SUB1_Click ()
    Dialog.SaveFile ()
    File.Save (Dialog.Path, TextArea1.Text)
CATTURARE
    SE Errore ALLORA RITORNA
FINE
```

Vedi anche Gambas: _Dateien # File_save²

12.1.3 Esempio 2

Hai scritto un programma che dovrebbe calcolare l'area circolare dal raggio. Si desidera impedire che nella casella di testo venga immesso qualcosa di diverso da un numero con l'immissione del raggio. Per fare ciò, aggiungi una routine di errore con CATCH al tuo codice.

```
CATTURARE
message.Info ("Immettere un numero")
FINE
```

Il programma completo si presenta così:

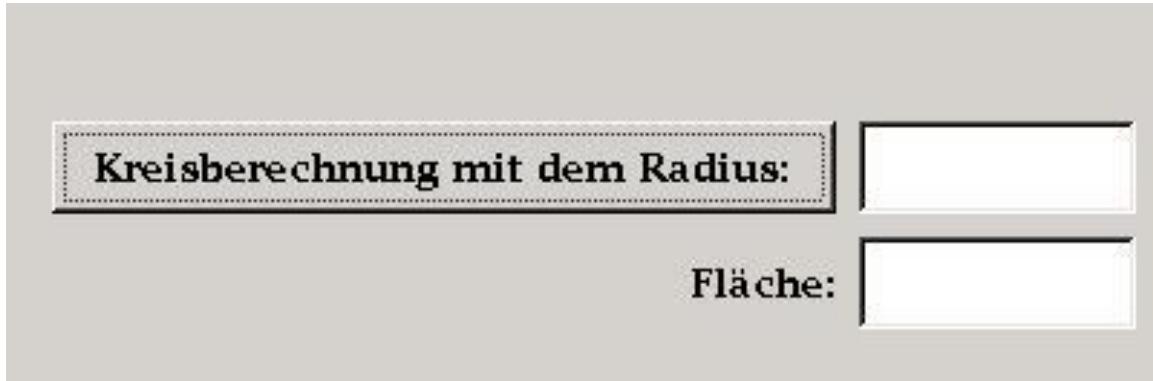


Fig.94

Hai bisogno:

- 2 caselle di testo
- 1 pulsante di comando
- 1 campo etichetta

per avviare il programma.

Il codice

```
PUBLIC SUB Form_Open ()
    Textbox1.Text = ""
    Textbox2.Text = ""
    Button1.Text = "Calcolo del cerchio con il raggio:"
    Label1.Text = "Area:"
    Label1.Alignment = 66
FINE
Pulsante PUBLIC SUB1_Click ()
r AS float
F AS galleggiante
```

² Capitolo 6.56 a pagina 265

```
IF Textbox1.Text = "" THEN
    r = 0
    Textbox1.Text = Str (0)
    ALTRO
        r = Val (textbox1.text)
    FINISCI SE
    F = Pi*r^2
    Textbox2.Text = Str (Round (F, -2))
    CATTURARE
        message.Info ("Immettere un numero")
    FINE
```

Fare uno sforzo consapevole per inserire una voce errata. Immettere una lettera nel campo Raggio. Cosa succede allora? Ora prova di nuovo la stessa cosa e disattiva la routine di errore CATCH dopo averla contrassegnata con il tasto destro del mouse e commentato il blocco.

12.1.4 Esempio 3

L'esempio seguente mostra un codice un po 'complicato dalla guida di gambas.

Esempio:

```
'Stampa un file sullo schermo
SUB PrintFile (FileName AS STRING)
    File DIM hFile AS
    DIM sLig AS STRING
        APRIRE FileName PER LEGGERE COME #hFile
        WHILE NOT EOF (hFile)
            INGRESSO LINEA #hFile, sLig
            PRINT sLig
        WEND
        FINALMENTE 'Sempre eseguito, anche se viene sollevato un
        errore CLOSE #hFile

    CATCH 'Eseguito solo in caso di errore
        PRINT "Impossibile stampare il file"; Nome del file
    FINE
```

12.2 Rilevare gli errori con TRY

Try è fondamentalmente lo stesso di Catch, solo che non l'intera funzione viene controllata per gli errori, ma solo un'istruzione speciale.

12.2.1 Esempio

In questo esempio viene creato un file nella directory home dell'utente corrente. Tuttavia, se l'accesso alla posizione di destinazione è bloccato (ad es. Se il file o la directory è di sola lettura), l'istruzione try genera un errore che può essere verificato utilizzando la parola chiave ERROR.

```
PUBLIC SUB Main ()
    PROVA File.Save ("/ home /" & User.Name & "/file.txt", "Hello" &
    Nome utente)
```

```
SE ERRORE ALLORA
PRINT "Si è verificato il seguente errore:" & Error.Text
PRINT "Proprio qui:" & Error.Where
ALTRO
PRINT "Ho un file con successo nella tua home directory
creato!"
FINISCI SE
FINE
```

12.3 F11

La finestra immediata viene chiamata con F11.

12.4 Inserimento diretto nella finestra diretta senza un programma

Benvenuto nella console Gambas. Per valutare un'espressione, digita "?" seguito dall'espressione e premere Invio.

Esempio:

```
? 5 + 3
```

Edizione: 8

12.5 Output nella finestra diretta con il comando PRINT

A differenza di Visual Basic, il comando di stampa viene emesso nella finestra diretta.

Esempio. Tutto ciò di cui hai bisogno è un modulo vuoto per avviare il programma. Inizia con F5.

```
PUBLIC SUB Form_Open ()
    STAMPA "Hello world!"
FINE
```



Fig.95

12.6 Programmazione senza grafica solo nella finestra diretta

È inoltre possibile utilizzare Gambas per programmi senza interfaccia utente grafica. Questi programmi vengono eseguiti nella finestra diretta (= finestra del terminale) sulla console. Con Visual Basic non c'è più questa possibilità.

Avvia Gambas come al solito.

Scegli Nuovo progetto.

Segui la procedura guidata e premi Avanti.

In Seleziona tipo di progetto, seleziona Crea progetto terminale.

Segui la procedura guidata e premi Avanti.

Dai un nome al tuo progetto, ad es. 1mal1.

Segui la procedura guidata e premi Avanti.

Seleziona la directory in cui salvare il tuo progetto come sottodirectory. (Di solito lasci il valore predefinito /gambas-0.99)

Segui la procedura guidata e premi Avanti.

I dati del tuo progetto vengono riepilogati.

Premere OK

Viene ora visualizzata la finestra del progetto. Fare clic con il pulsante destro del mouse sulla directory Classes.

Quindi premere il pulsante sinistro del mouse su Nuovo e Classe.

Lascia il nome a Class1 come classe iniziale e premi OK.

Immettere il seguente programma come esempio:

```
STATICO PUBBLICO SUB Principale ()  
DIM x AS intero  
DIM y AS intero
```

```
PER x = da 1 a 10  
    PER y = da 1 a 10  
        PRINT x; "volte"; y; "="; X*y  
    IL PROSSIMO  
IL PROSSIMO
```

```
FINE
```

Premi F5 per avviarlo.

Dai un'occhiata al risultato nella finestra diretta. Possibilmente passare avanti e indietro tra le finestre con Alt + Tab.

Esci dal programma cliccando sul quadrato rosso nella finestra del progetto. O con ALT + F4.

Esci da Gambas con File Esci o con Ctrl + Q o continua a programmare.

12.7 Domanda:

Come puoi rendere un programma non grafico accessibile come file eseguibile sul desktop?

Perché gli array non funzionano in modalità terminale?

Come puoi ridurre la finestra di output diretto a livello di programmazione?

12.8 Tipi di dati

gb.zero	Valore zero
gb.Boolean	Valore di verità
gb.Byte	Byte intero
gb.breve	Intero breve
gb.Integer	Numeri interi
gb.Float	Numeri con virgola mobile
gb.Date	Data e ora
gb.String	String (testo)
gb.Varient	Variante
gb.Object	Riferimento oggetto

12.9 tipi di file

gb.File	File regolare
gb.directory	directory
gb.dispositivo	File speciale per un dispositivo
gb.Pipe	Pipa chiamata
gb.socket	File speciale per un socket
gb.link	Collegamento simbolico

12.10 Costanti stringa (testo)

gb.NewLine	Nuova linea. Corrisponde a Chr \$ (10)
gb.Tab	Tab. Corrisponde a Chr \$ (9)

12.11 Tipi di ordinamento

gb.Binary	Ordinamento binario
gb.Case	Non è case sensitive
gb.lungo	Ordinamento basato sulla lingua

12.12 giorni della settimana

gb.lunedì	Lunedì
gb.martedì	martedì
gb.mercoledì	mercoledì
gb.giovedì	giovedì
gb.venerdì	Venerdì
gb.sabato	Sabato
gb.domenica	Domenica

12.13 Formati numerici predefiniti

gb.GeneralNumber	Scrive un numero con 12 cifre decimali. Usa la conoscenza notazione sociale se l'importo assoluto è inferiore a 1 / 10.000 o maggiore di 10.000.000.
gb.Fixed	corrisponde a "0,00"
gb.Percent	corrisponde a "##%"
gb.scientifico	Scrive un numero con il suo esponente e 18 cifre decimali.

12.14 Formati di data e ora predefiniti

gb.GeneralDate	Scrivi una data solo se il valore di data e ora ha una parte data e scrivi a tempo solo se ha una parte di data.
gb.LongDate	Formato data lungo.
gb.MediumDate	Formato medio della data.
gb.ShortDate	Formato data breve.
gb.LongTime	Formato lungo.
gb.MediumTime	Formato medio orario.
gb.ShortTime	Formato breve.

12.15 Diversi formati

gb. standard	Utilizza gb.GeneralNumber per la formattazione dei numeri e gb.GeneralDate per formattazione di date e ore.
--------------	---

12.16 Ulteriori costanti

12.16.1 colori

Nero, blu, ciano, blu scuro, ciano scuro, grigio scuro, verde scuro, magenta scuro, giallo scuro rosso scuro, predefinito, grigio, verde, grigio chiaro, magenta, arancione, rosa, rosso, trasparente, viola, bianco, giallo

12.16.2 Chiavi

BackSpace, BackTab, CapsLock, Elimina, Giù, Fine, Invio, Esc, Esc, F1, F10, F11, F12, F13, F14, F15, F16, F17, F18, F19, F2, F20, F21, F22, F23, F24, F3, F4, F5, F6, F7, F8, F9, Guida, Home, Inserisci, Sinistra, Menu, Bloc Num, Pagina giù, Pagina su, Pausa, Stampa, A capo, Destra, Blocco scorrimento, Spazio, SysReq, Tab, Su

12.16.3 cornice

Inciso, Nessuno, Semplice, Sollevato, Affondato

12.17 costanti

Costanti	Esempi
Il vero valore.	VERO
Il falso valore.	FALSO
Numeri interi.	0, 123, -32769

Costanti	Esempi
Numeri esadecimali brevi.	& H1F5, & HFFFF e FFFF
Numeri contrassegnati in esadecimali.	& H10BF332E e 10BF332E
Numeri esadecimali senza segno.	& H8000 &, & HFFFF &
Numeri binari.	& X1010010101,% 101001011
Numeri in virgola mobile.	1.0, -5.345219E + 45
Costanti di testo.	"Ciao mondo!"
Costanti stringa da tradurre.	("Questo software è fantastico")
Zero costante / testo vuoto.	ZERO

12.18 Sequenze di escape

Le costanti di testo possono contenere le seguenti sequenze di escape:

Sequenza di fuga	Equivalenti ASCII	Corrispondenza
\n	CHR \$ (13)	nuova linea
\r	CHR \$ (10)	Ritorno in carrozza
\t	CHR \$ (9)	tab
\ "	CHR \$ (34)	Virgolette
\ \	CHR \$ (92)	Barra rovesciata
\ xx	CHR \$ (& Hxx)	Codice esadecimale

È possibile scrivere costanti di testo in parti o insieme. Ad esempio, "Gambas" "è" "ottimo" risultati allo stesso modo di "Gambas è ottimo"

Componenti da 12,19 GB

I componenti sono plug-in Gambas che aggiungono nuove classi all'interprete. Ogni progetto ha un elenco dei componenti che devono essere caricati all'avvio. Puoi trovare questo elenco sotto la voce di menu Proprietà del progetto.

Un progetto Gambas senza alcun componente è una semplice applicazione di testo. Se deve essere una vera applicazione grafica, deve essere caricato almeno il componente gb.qt. Per utilizzare un database è necessario il componente gb.db ecc.

gb	Classe base interna di Gamba
gb.compress	Libreria di compressione
gb.corba	Cliente Corba
gb.crypt	Componente di crittografia
gb.db	Accesso al database
gb.db.form	Rappresentazione di dati e database con DataControl e DataSource
gb.debug	Guida di Gambas Debugger
gb.eval	La valutazione di Gamba delle espressioni matematiche
gb.form	ulteriori controlli per applicazioni grafiche
gb.gtk	Componente grafico GTK + toolkit

gb.image	Elaborazione delle immagini
gb.info	Informazioni sui componenti installati
gb.ldap	Libreria di codice OpenLdap
gb.net	Componente di rete
gb.net.curl	Gestione dei protocolli di rete di alto livello
gb.opengl	Libreria OpenGL
gb.pcre	Corrispondenza di espressioni regolari compatibile con Perl
gb.pdf	Componente PDF basato sulla libreria Poppler
gb.qt	Toolkit grafico QT
gb.qt.editor	Editor di Gambas con evidenziazione della sintassi
gb.qt.ext	Estensione del toolkit QT grafico
gb.qt.kde	Integrazione e scripting di KDE
gb.qt.kde.html	Browser web KDE
gb.qt.opengl	Componente grafico QT toolkit opengl
gb.sdl	Libreria basata su SDL
gb.sdl.image	Libreria 2D basata su SDL
gb.sdl.opengl	Libreria OpenGL basata su SDL
gb.sdl.sound	Libreria di suoni basata su SDL
gb.impostazioni	Gestione delle preferenze e dei file di configurazione
gb.v4l	Componente di acquisizione video
gb.vb	Compatibilità con Visual Basic
gb.xml.libxml	Strumenti XML basati su libxml
gb.xml.libxml.rpc	Client XML-RPC basato su libxml e libcurl
gb.xml.libxml.xslt	Strumenti XSLT basati su libxslt

12.19.1 Caricare componenti aggiuntivi

È possibile accedere a componenti aggiuntivi di Gambas tramite la voce di menu Progetto - Proprietà della schermata principale di Gambas.



Fig.96

Selezionare i componenti desiderati nella visualizzazione dei componenti.

Ad esempio, nella cassetta degli attrezzi viene visualizzato un carattere con la designazione QT. Fare clic su questo e quindi avere una serie di altri controlli disponibili.

12.19.2 Elenco dei componenti attuali del gas

componente	descrizione	Controlli
gb	Classi native interne di Gambas	
gb.compress	Libreria di compressione	
gb.db	Componente di accesso al database	
gb.debug	Debugger dell'applicazione Gambas aiutante	
gb.eval	Valutatore di espressione di Gambas	
gb.net	Componente di rete	DnsClient, ServerSocket, Serial-Porta, Socket, UdpSocket
gb.qt	Componente grafico QT toolkit	Etichetta, TextLabel, PictureBox, ProgressBar, pulsante, casella di controllo, RadioButton, ToggleButton, Tool-Pulsante, casella di testo, casella combinata, TextArea, ListBox, ListView, Tree-Visualizza, IconView, GridView, ColumnView, HBox, VBox, HPanel, VPanel, Frame, Panel, TabStrip, ScrollView, DrawingArea, Timer
gb.qt.editor	Editor di Gambas con sintassi alta	GambasEditor

componente	descrizione	Controlli
gb.qt.ext	Estensione grafica del toolkit QT componente	LCDNumber, quadrante, casella di selezione, ScrollBar, Slider, MovieBox, TableView, HSplit, VSplit, Workspace, TextView
gb.qt.kde	Integrazione e scripting di KDE	URLLabel, ColorBox, DatePicker

12.19.3 Componenti sperimentali

AVVERTIMENTO! I seguenti componenti sono ancora in fase BETA e possono quindi cambiare in qualsiasi momento.

gb.qt.kde.html	Browser web KDE
gb.vb	Compatibilità con Visual Basic
gb.xml.libxml	Strumenti XML basati su libxml
gb.xml.libxml.rpc	Client XML-RPC basato su libxml e libcurl
gb.xml.libxml.xslt	Strumenti XSLT, basati su libxslt

12.19.4 File di programma

Il programma Gambas si trova in directory diverse a seconda della distribuzione o del sistema operativo utilizzato. Linux dovrebbe avere un file

individuare i gambas

Fornire informazioni su dove si trovano gli stessi Gamba, il programma di esempio, ecc.

12.19.5 Progetti propri

I tuoi progetti sono solitamente in / home / USERS / gambasxx /

12.19.6 File di configurazione

La directory nascosta /home/BENUTZER/.gambas contiene i file di configurazione

- file: /home/USER/.gambas/gambas.conf
- file: /home/USER/.gambas/gambas-database-manager.conf

12.19.7 File di progetto

Ogni progetto ha la propria sottodirectory. Per esempio:

/home/rho/gambas-1.0/addy	Directory del progetto
---------------------------	------------------------

/home/rho/gambas-1.0/addy/addy	File eseguibile
/home/rho/gambas-1.0/addy/addy.png	Un'icona per il programma
/home/rho/gambas-1.0/addy/Form1.class	GambasCode in testo normale
/home/rho/gambas-1.0/addy/Form1.class~	File di backup
/home/rho/gambas-1.0/addy/Form1.form	File di forma in testo normale
/home/rho/gambas-1.0/addy/.project	Dati di progetto in testo normale
/home/rho/gambas-1.0/addy/.gambas	Un'altra sottodirectory nascosta. Cosa serve?

Esempio di un file di progetto

```
# Gambas Project File 1.0
Project = addy Startup = Form1
TabSize = 2 Version = 0.0.2
Library = gb.qt SnapToGrid = 1
ShowGrid = 1

Snap = 8
Localizza = 0
KeepDebugInfo = 0
ControlPublic = 0 ExecPath = / home / name /
gambas-1.0 / addy / addy Prefix = 0
```

Esempio di un semplice file di forma

```
# Gambas Form File 1.0
{Form1 Form
    Sposta (227.172.216.400)
    Testo = ("")
    Border = Window.Fixed
    {TextArea1 TextArea Move
        (16,8,184,280)
        Testo = ("")
    }
    {Button1 Button Move
        (16,296,184,32)
        Carattere = Carattere ["14"]
        Testo = ("somma")
    }
    {TextBox1 TextBox Move
        (16,336,184,32) Text =
        ("")
    }
}
```

12.20 Come funziona Gambas?

Quando scrivi un programma in Gambas, crei

- uno o più file di modulo
 - Il file del modulo contiene una descrizione dell'interfaccia del programma desiderata
- uno o più file di classe
 - Il file di classe contiene il codice del programma che hai scritto tu stesso.
- un file .project
 - Il file di progetto contiene informazioni sul progetto come il nome e il numero di versione.

Questi file vengono tradotti in pcode binario dal compilatore gbc. Il pcode viene salvato nella directory .gambas.

Quindi l'interprete gbx valuta il pcode e recupera i componenti Qt richiesti. I componenti Qt fanno parte dei gambas. Vedere Gambas: componenti³

Un componente QT è un programma C ++ / qt che riceve la descrizione della GUI dal pCode e che genera Qt Gui.

gba è un archiviatore. Prende le informazioni .project e le incolla insieme al codice p per formare un file eseguibile del programma eseguibile. Per eseguire il programma è comunque necessario gbx.

- Progetto - "compile (F7)" avvia gbc.
- Progetto - "run (F5)" avvia gbc e gbx.
- Progetto - "Rendi eseguibile (Ctrl-Alt-M)" avvia gbc e gba

Chi ha sviluppato i gamberi?

Guarda anche: <http://gambas.sourceforge.net/fosdem2005.jpg> Benoit l'uomo con la maglietta rossa

- Benoît Minisini
 - Francia
 - Il capo di tutto
- Fabien Bodard
 - Francia
 - Selettore di file / progetto / immagine IDE e riprogettazione della mascotte.
- Nigel Gerrard
 - Regno Unito
 - Il driver del database MySQL, il componente esteso QT, il driver del database SQLite.
- Paul Gardner-Stephen
 - Australia
 - Porting di Gambas su Solaris
- Daniel Campos
 - Spagna
 - Il componente di rete, il componente di compressione. Funziona anche sul componente GTK +!
- Carlos FA Paniago
 - Brasile

³ Capitolo 12.18 a pagina 424

- Portare Gambas su FreeBSD.
- Rob Kudla
 - stati Uniti
 - Webmaster GambasWiki e manutentore dei pacchetti Mandrake RPM. Il componente Perl Compatible Regular Expression.
- Ronald Onstenk
 - Olanda
 - L'elenco di ricerca nell'ambiente di sviluppo.
- Brandon Bergren
 - Porting di Gambas su CygWin.
- Laurent Carlier
 - Francia
 - Il componente SDL.
- José L. Redrejo Rodríguez
 - Spagna
 - Lascia che Gambas compili su tutte le architetture Debian.

Gambas è più strettamente correlato a VB2005 (Vb.net) che a VB3,4,5,6 perché è realmente orientato agli oggetti. Le varianti VB 3-6 non sono più supportate da Microsoft e scadranno prima o poi. Con Gambas esiste un vero ambiente di programmazione di base open source orientato agli oggetti sotto Linux, come l'ambiente di programmazione orientato agli oggetti VB.Net sotto Windows.

12.21 Differenze tra Visual Basic e Gambas

Vedere la pagina seguente: <http://www.gambasdoc.org/help/doc/diffvb?de>

La programmazione delle funzioni in Gambas è leggermente diversa dalla programmazione in VB. In Gambas non è possibile chiamare il nome della funzione nella funzione, in VB è possibile.

Il comando Seleziona caso di comando in Gambas differisce leggermente dal suo utilizzo in VB.

La funzione timer emette il tempo in secondi trascorsi dall'inizio del programma. Questo è diverso in Visual Basic. Il tempo trascorso dalla mezzanotte viene emesso con lo stesso programma.

In VB, il codice del programma dell'oggetto forma e la definizione della forma sono memorizzati in uno stesso file. Gambas separa i due: c'è un file .form e un file .class.

12.22 Panoramica delle estensioni di file a confronto:

Tipo di file:	VB	Gamberi
Definizione del progetto	.vbp	.progetto

Moduli	.bas	.modulo
Classe di estensione del file	.cls	.classe
File di definizione del modulo	.frm	.forma
File binari	.frx	qualunque

12.23 Importazione modulo VB

C'è un ImportVBProject di Rob Kudla, che converte i moduli VB in classi Gambas.
La conversione automatica del codice VB in codice Gambas è più difficile. C'è qualcosa su questo Necessario lavoro manuale.

12.24 Variabili globali in Gambas

In VB era possibile, ad esempio, indirizzare una casella di testo in Form1 direttamente da Form2. Il comando per questo è: Form1! Textbox1.Text = "zb". Questo non funziona a Gambas. Non ci sono variabili globali cross-project in Gambas. Si può andare in giro creando una classe chiamata Global. Le variabili globali sono dichiarate come variabili pubbliche statiche in questa classe. Quindi puoi accedervi con Global.variablename nell'intero progetto.

Minsini: È ancora una cattiva pratica di programmazione, ma almeno verranno identificate come variabili globali ogni volta che le usi;)

Vedi Gambas: _Form # Globale_Variablen_in_Gambas⁴

12.25 IsNumber

Domanda su isnumber

```
TextArea1.Text = "349"  
PRINT IsNumber (TextArea1.Text)
```

risulta in FALSE. Perché ?

IsNumber () controlla solo il tipo di un valore che inizia da booleano a float. Si comporta in modo diverso da IsNumber () in VB.

Se vuoi sapere se una stringa di testo è un numero, devi usare Val () / CInt () / CFloat () e controllare se ricevi un messaggio di errore.

4 Capitolo 0.97 a pagina 91

12.26 Comando di scaricamento non disponibile in Gambas

Il metodo Unload è disponibile in VB. Questo non è disponibile in Gambas. In alternativa, puoi utilizzare il metodo Nascondi o Elimina. C'è anche il metodo Close per un modulo.

Esempio: per questo esempio sono necessari un modulo e un pulsante di comando.

```
Pulsante PUBLIC SUB1_Click ()  
    Button1.delete  
FINE
```

O

```
Pulsante PUBLIC SUB1_Click ()  
    ME.delete  
    'Form1.close  
FINE
```

13 autori

EditsUser

3 Ammiraglio Snyder¹
114 BK²
1 bastie³
3 Berni⁴ °
1 Blak⁵
8 Carlo⁶ °
1 Chirak⁷ °
1 Daniel B.⁸ °
1 Delphimann⁹
7 Dieter¹⁰
119 Dirk Huenniger¹¹
1 FeG¹² °
1 Gronau¹³ °
49 Harald Weber^{14th}
7 hero03¹⁵ °
1 hoo man¹⁶
10IslandAX¹⁷ °
2 Joel wtal¹⁸ °
2 Jorunn¹⁹ °
2 Testo normale^{20th}
1 Klaus Eifert²¹

-
- 1 http://de.wikibooks.org/w/index.php?title=Benutzer:Admiral_Snyder
 - 2 <http://de.wikibooks.org/w/index.php?title=Benutzer:BK>
 - 3 <http://de.wikibooks.org/w/index.php?title=Benutzer:Bastie>
 - 4 <http://de.wikibooks.org/w/index.php?title=Benutzer:Berni>
 - 5 <http://de.wikibooks.org/w/index.php?title=Benutzer:Blak>
 - 6 <http://de.wikibooks.org/w/index.php?title=Benutzer:Carlo>
 - 7 <http://de.wikibooks.org/w/index.php?title=Benutzer:Chirak>
 - 8 http://de.wikibooks.org/w/index.php?title=Benutzer:Daniel_B
 - 9 <http://de.wikibooks.org/w/index.php?title=Benutzer:Delphimann>
 - 10 <http://de.wikibooks.org/w/index.php?title=Benutzer:Dieter>
 - 11 http://de.wikibooks.org/w/index.php?title=Benutzer:Dirk_Huenniger
 - 12 ° <http://de.wikibooks.org/w/index.php?title=Benutzer:FeG>
 - 13 ° <http://de.wikibooks.org/w/index.php?title=Benutzer:Gronau>
 - 14th http://de.wikibooks.org/w/index.php?title=Benutzer:Harald_Weber
 - 15 ° <http://de.wikibooks.org/w/index.php?title=Benutzer:Held03>
 - 16 http://de.wikibooks.org/w/index.php?title=Benutzer:Hoo_man
 - 17 ° <http://de.wikibooks.org/w/index.php?title=Benutzer:InselAX>
 - 18 ° http://de.wikibooks.org/w/index.php?title=Benutzer:Joel_wtal
 - 19 ° <http://de.wikibooks.org/w/index.php?title=Benutzer:Jorunn>
 - 20th <http://de.wikibooks.org/w/index.php?title=Benutzer:Klartext>
 - 21 http://de.wikibooks.org/w/index.php?title=Benutzer:Klaus_Eifert

11 Klausmach²²
1 Kristjan²³
1 Martin Fuchs²⁴
1 Contro Matthias M.^{25th}
3 Merkel^{26th}
1 Michael Frey²⁷
48 Momomu²⁸
1 Norarg²⁹
11 Patrick Hilker^{30th}
1 planktome³¹
5 Progman³²
6 Rathjen³³
154Rho³⁴
5 Robbo21³⁵
6 Rudolf73³⁶
2 Sundance Raphael³⁷
8 ThePacker³⁸
1 Timo Müller³⁹
1 tomodachi⁴⁰
1 Tschäfer⁴¹
3 Wiki marmotta⁴²
3 YellowZeta⁴³

-
- 22 <http://de.wikibooks.org/w/index.php?title=Benutzer:Klausmach>
23 <http://de.wikibooks.org/w/index.php?title=Benutzer:Kristjan>
24 http://de.wikibooks.org/w/index.php?title=Benutzer:Martin_Fuchs
25 http://de.wikibooks.org/w/index.php?title=Benutzer:Matthias_M.
26 <http://de.wikibooks.org/w/index.php?title=Benutzer:Merkel>
27 <http://de.wikibooks.org/w/index.php?title=Benutzer:MichaelFrey>
28 <http://de.wikibooks.org/w/index.php?title=Benutzer:Momomu>
29 <http://de.wikibooks.org/w/index.php?title=Benutzer:Norarg>
30 http://de.wikibooks.org/w/index.php?title=Benutzer:Patrick_Hilker
31 <http://de.wikibooks.org/w/index.php?title=Benutzer:Planktom>
32 <http://de.wikibooks.org/w/index.php?title=Benutzer:Progman>
33 <http://de.wikibooks.org/w/index.php?title=Benutzer:Rathjen>
34 <http://de.wikibooks.org/w/index.php?title=Benutzer:Rho>
35 <http://de.wikibooks.org/w/index.php?title=Benutzer:Robbo21>
36 <http://de.wikibooks.org/w/index.php?title=Benutzer:Rudolf73>
37 http://de.wikibooks.org/w/index.php?title=Benutzer:Sundance_Raphael
38 <http://de.wikibooks.org/w/index.php?title=Users:ThePacker>
39 http://de.wikibooks.org/w/index.php?title=Benutzer:Timo_M%C3%BCller
40 <http://de.wikibooks.org/w/index.php?title=Benutzer:Tomodachi>
41 <http://de.wikibooks.org/w/index.php?title=Benutzer:Tsch%C3%A4fer>
42 <http://de.wikibooks.org/w/index.php?title=Benutzer:Wikimurmeltier>
43 <http://de.wikibooks.org/w/index.php?title=Benutzer:YellowZeta>

Lista delle figure

- GFDL: Gnu Free Documentation License. <http://www.gnu.org/licenses/fdl.html>
- cc-by-sa-3.0: Licenza Creative Commons Attribution ShareAlike 3.0. <http://creativecommons.org/licenses/by-sa/3.0/>
- cc-by-sa-2.5: Licenza Creative Commons Attribution ShareAlike 2.5. <http://creativecommons.org/licenses/by-sa/2.5/>
- cc-by-sa-2.0: Licenza Creative Commons Attribution ShareAlike 2.0. <http://creativecommons.org/licenses/by-sa/2.0/>
- cc-by-sa-1.0: Licenza Creative Commons Attribution ShareAlike 1.0. <http://creativecommons.org/licenses/by-sa/1.0/>
- cc-by-2.0: Licenza Creative Commons Attribution 2.0. <http://creativecommons.org/licenze/by/2.0/>
- cc-by-2.0: Licenza Creative Commons Attribution 2.0. <http://creativecommons.org/licenze/by/2.0/deed.en>
- cc-by-2.5: Licenza Creative Commons Attribution 2.5. <http://creativecommons.org/licenze/by/2.5/deed.en>
- cc-by-3.0: Licenza Creative Commons Attribution 3.0. <http://creativecommons.org/license/by/3.0/deed.en>
- GPL: GNU General Public License. <http://www.gnu.org/licenses/gpl-2.0.TXT>
- LGPL: GNU Lesser General Public License. <http://www.gnu.org/licenses/lgpl.html>
- PD: Questa immagine è di pubblico dominio.
- ATTR: Il detentore del copyright di questo file consente a chiunque di utilizzarlo per qualsiasi scopo, a condizione che il detentore del copyright sia correttamente attribuito. La ridistribuzione, il lavoro derivato, l'uso commerciale e tutti gli altri usi sono consentiti.
- EURO: Questa è la faccia comune (rovescio) di una moneta in euro. Il diritto d'autore sul disegno della faccia comune delle monete in euro appartiene alla Commissione europea. È autorizzata la riproduzione in un formato senza rilievo (disegni, dipinti, filmati) a condizione che non rechino pregiudizio all'immagine dell'euro.
- LFK: Licenza artistica gratuita. <http://artlibre.org/licence/lal/de>
- CFR: utilizzo senza copyright.

Lista delle figure

- EPL: Eclipse Public License. <http://www.eclipse.org/org/documents/ epl-v10.php>

Le copie della GPL, della LGPL e della GFDL sono incluse nelle licenze del capitolo⁴⁴. Please note that images in the public domain do not require attribution. You may click on the image numbers in the following table to open the webpage of the images in your webbrowser.

⁴⁴Kapitel 14 auf Seite 441

1	Benoît Minisini	GPL
2	Harald Weber	GFDL
3	User:Rho ⁴⁵	GFDL
4		PD
5		PD
6		PD
7		PD
8		PD
9		PD
10	Benoît Minisini	GPL
11	Benoît Minisini	GPL
12	Benoît Minisini	GPL
13		PD
14		PD
15	Benoît Minisini	GPL
16	Benoît Minisini	GPL
17	Benoît Minisini	GPL
18	Benoît Minisini	GPL
19	Benoît Minisini	GPL
20	Benoît Minisini	GPL
21	Benoît Minisini	GPL
22	Benoît Minisini	GPL
23	Benoît Minisini	GPL
24	Benoît Minisini	GPL
25	Benoît Minisini	GPL
26	Benoît Minisini	GPL
27	Benoît Minisini	GPL
28	Benoît Minisini	GPL
29	Benoît Minisini	GPL
30		PD
31		PD
32		PD
33	Benutzer:Rho ⁴⁶	GFDL
34		PD
35		PD
36		PD
37		PD
38	Benutzer:Rho ⁴⁷	GFDL
39		PD
40		PD
41	Benutzer:Rho ⁴⁸	GFDL
42		PD
43	Benutzer:Rho ⁴⁹	GFDL
44	Benutzer:Rho ⁵⁰	GFDL

45 <http://de.wikibooks.org/wiki/User%3ARho>

46 <http://de.wikibooks.org/wiki/Benutzer%3ARho>

47 <http://de.wikibooks.org/wiki/Benutzer%3ARho>

48 <http://de.wikibooks.org/wiki/Benutzer%3ARho>

49 <http://de.wikibooks.org/wiki/Benutzer%3ARho>

50 <http://de.wikibooks.org/wiki/Benutzer%3ARho>

45		PD
46		PD
47	Benutzer:Rho ⁵¹	GFDL
48		PD
49		PD
50		PD
51	Benutzer:Rho ⁵²	GFDL
52	Benutzer:Rho ⁵³	GFDL
53	Benutzer:Rho ⁵⁴	GFDL
54	Benoît Minisini	GFDL
55	Dr.Reiner Hoffmann	GFDL
56	Dr.Reiner Hoffmann	GFDL
57	Dr.Reiner Hoffmann	GFDL
58		PD
59		PD
60		PD
61		PD
62		PD
63		PD
64	Benoît Minisini	GFDL
65	Benoît Minisini	GFDL
66	Dr.Reiner Hoffmann	GFDL
67		PD
68		PD
69	Dr.Reiner Hoffmann	GFDL
70	Benutzer:Rho ⁵⁵	GFDL
71		PD
72		PD
73		PD
74		PD
75	Harald Weber	GFDL
76	Benoît Minisini	GPL
77		PD
78		PD
79		PD
80		PD
81		PD
82		PD
83		PD
84		PD
85		PD
86		PD
87	Harald Weber	GFDL
88		PD
89		PD

51 <http://de.wikibooks.org/wiki/Benutzer%3ARho>

52 <http://de.wikibooks.org/wiki/Benutzer%3ARho>

53 <http://de.wikibooks.org/wiki/Benutzer%3ARho>

54 <http://de.wikibooks.org/wiki/Benutzer%3ARho>

55 <http://de.wikibooks.org/wiki/Benutzer%3ARho>

90		PD
91		PD
92		PD
93		PD
94		PD
95	Harald Weber	GFDL
96	Benoît Minisini	GFDL

14 licenze

14.1 LICENZA PUBBLICA GENERALE GNU

Versione 3, 29 giugno 2007

Copyright © 2007 Free Software Foundation, Inc. <<http://fsf.org/>>

A tutti è consentito copiare e distribuire copie letterali di questo documento di licenza, ma non è consentito modificarlo. Preambolo

La GNU General Public License è una licenza gratuita con copyleft per software e altri tipi di opere.

Le licenze per la maggior parte del software e altri lavori pratici sono progettate per toglierti la libertà di condividere i lavori. Al contrario, la GNU General Public License ha lo scopo di garantire la libertà di condividere e modificare tutte le versioni di un programma — per assicurarsi che rimanga software gratuito per tutti i suoi utenti. Noi, la Free Software Foundation, utilizziamo la GNU General Public License per la maggior parte del nostro software. Si applica anche a qualsiasi altro lavoro pubblicato in questo modo dai suoi autori. Puoi applicarlo anche ai tuoi programmi.

Quando parliamo di software libero, ci riferiamo alla libertà, non al prezzo. Le nostre licenze per il pubblico generico sono progettate per assicurarti che tu abbia la libertà di distribuire copie di software gratuito (e addebitarle se lo desideri), che tu riceva il codice sorgente o che puoi ottenerlo se lo desideri, che puoi cambiare il software o usarne parti in nuovi programmi gratuiti e che sai di poter fare queste cose.

Per proteggere i tuoi diritti, dobbiamo impedire ad altri di negarti questi diritti o chiederti di rinunciare ai diritti. Pertanto, hai determinate responsabilità se distribuisci copie del software o se lo modifichi: responsabilità nel rispetto della libertà degli altri.

Ad esempio, se distribuisci copie di tale programma, sia gratuitamente che a pagamento, devi trasferire ai destinatari le stesse libertà che hai ricevuto. Devi assicurarti che anche loro ricevano o possano ottenere il codice sorgente. E devi mostrare loro questi termini in modo che conoscano i loro diritti.

Gli sviluppatori che utilizzano la GNU GPL proteggono i tuoi diritti con due passaggi: (1) rivendicano il copyright sul software e (2) ti offrono questa licenza dandoti il permesso legale di copiare, distribuire e / o modificarlo.

Per la protezione degli sviluppatori e degli autori, la GPL spiega chiaramente che non esiste alcuna garanzia per questo software gratuito. Per il bene degli utenti e degli autori, la GPL richiede che le versioni modificate siano contrassegnate come cambiate, in modo che i loro problemi non vengano attribuiti erroneamente agli autori di versioni precedenti.

Alcuni dispositivi sono progettati per negare agli utenti l'accesso per installare o eseguire versioni modificate del software al loro interno, sebbene il produttore possa farlo. Ciò è fondamentalmente incompatibile con l'obiettivo di proteggere la libertà degli utenti di modificare il software. Il modello sistematico di tale abuso si verifica nell'area dei prodotti che le persone possono utilizzare, che è precisamente dove è più inaccettabile. Pertanto, abbiamo progettato questa versione della GPL per vietare la pratica per quei prodotti. Se tali problemi sorgono sostanzialmente in altri domini, siamo pronti ad estendere questa disposizione a quei domini nelle future versioni della GPL, come necessario per proteggere la libertà degli utenti.

Infine, ogni programma è costantemente minacciato dai brevetti software. Gi Stati non dovrebbero consentire che i brevetti limitino lo sviluppo e l'uso del software su computer di uso generale, ma in quelli che lo fanno, desideriamo evitare il pericolo speciale che i brevetti applicati a un programma libero possano renderlo effettivamente proprietario. Per evitare ciò, la GPL assicura che i brevetti non possono essere utilizzati per rendere il programma non libero.

Seguono i termini e le condizioni precisi per la copia, la distribuzione e la modifica. TERMINI E CONDIZIONI 0. Definizioni.

Questa Licenza si riferisce alla versione 3 della GNU General Public License.

"Copyright" indica anche leggi simili al copyright che si applicano ad altri tipi di opere, come le maschere di semiconduttori.

"Il Programma" si riferisce a qualsiasi opera coperta da copyright concessa in licenza in base a questa Licenza. Ogni licenza è indicata come "tu", "Licenziatari" e "destinatari" possono essere individui o organizzazioni.

Per "modificare" un'opera significa copiare o adattare tutta o parte dell'opera in un modo che richiede il permesso di copyright, diverso dalla realizzazione di una copia esatta. Il lavoro risultante è chiamato una "versione modificata" del lavoro precedente o un lavoro "basato sul" il lavoro precedente.

Un "lavoro coperto" indica il programma non modificato o un lavoro basato sul programma.

"Propagare" un'opera significa fare qualsiasi cosa con essa che, senza permesso, ti renderebbe direttamente o in secondo luogo responsabile di violazione ai sensi della legge sul copyright applicabile, tranne eseguirla su un computer o modificare una copia privata. La propagazione include la copia, la distribuzione (con o senza modifica), la messa a disposizione del pubblico e, in alcuni paesi, anche altre attività.

Per "trasmettere" un'opera si intende qualsiasi tipo di propagazione che consenta ad altre parti di fare o ricevere copie. La semplice interazione con un utente attraverso una rete di computer, senza trasferimento di una copia, non è trasmissione.

Un'interfaccia utente interattiva visualizza "Avvisi legali appropriati" nella misura in cui include una funzione comoda e ben visibile che (1) visualizza un avviso di copyright appropriato e (2) comunica altuterni che non esiste alcuna garanzia per il lavoro (ad eccezione di la misura in cui vengono fornite le garanzie), che le licenze possono trasmettere il lavoro sotto questa licenza e come visualizzare una copia di questa licenza. Se l'interfaccia presenta un elenco di comandi o opzioni utente, come un menu, un elemento in evidenza nel menu, anche altre attività.

Il "codice sorgente" di un'opera indica la forma preferita dell'opera per apportare modifiche ad essa. "Codice oggetto" indica qualsiasi forma non originale di un'opera.

Una "interfaccia standard" indica un'interfaccia che è uno standard ufficiale definito da un ente di normalizzazione riconosciuto o, nel caso di interfacce specificate per un particolare linguaggio di programmazione, ampiamente utilizzato tra gli sviluppatori che lavorano in quel linguaggio.

Le "Librerie di sistema" di un'opera eseguibile includono qualsiasi cosa, oltre all'opera nel suo insieme, che (a) è inclusa nella normale forma di

confezionamento di un Componente Principale, ma che non fa parte di quella Componente Principale, e

(b) serve solo per consentire l'uso del lavoro con quel componente principale, o per implementare un'interfaccia standard per la quale un'implementazione è disponibile al pubblico sotto forma di codice sorgente. Un "Componente Principale", in questo contesto, significa un componente essenziale importante (kernel, sistema a finestre e così via) del sistema operativo specifico (se presente) su cui viene eseguito il lavoro eseguibile, o un compilatore utilizzato per produrre il lavoro o un interprete del codice oggetto utilizzato per eseguirlo.

La "Sorgente corrispondente" per un'opera in forma di codice oggetto indica tutto il codice sorgente necessario per generare, installare e (per un lavoro eseguibile) eseguire

oggetto è un server di rete, la fonte corrispondente potrebbe trovarsi su un server diverso (gestito da te o da una terza parte) che supporta funzioni di copia equivalenti, a condizione di mantenere chiare indicazioni accanto al codice dell'oggetto che dicono dove trovare la sorgente corrispondente. Indipendentemente da quale server ospita la sorgente corrispondente, l'utente resta obbligato a garantire che sia disponibile per tutto il tempo necessario a soddisfare questi requisiti. * e) Trasmettere il codice oggetto utilizzando la trasmissione peer-to-peer, a condizione di informare altri peere dove il codice oggetto e la fonte corrispondente del lavoro vengono forniti al pubblico in generale senza alcun costo ai sensi della sottosezione 6d.

La sorgente corrispondente non deve includere nulla che gli utenti possano rigenerare automaticamente da altre parti della sorgente corrispondente.

La fonte corrispondente per un'opera sotto forma di codice sorgente è la stessa opera.
2. Autorizzazioni di base.

Tutti i diritti concessi in base a questa Licenza sono garantiti per la durata del copyright sul programma e sono irrevocabili a condizione che siano soddisfatte le condizioni indicate. Questa Licenza afferma esplicitamente il tuo permesso illimitato per eseguire il programma non modificato. L'output derivante dall'esecuzione di un'opera coperta è coperto da questa licenza solo se l'output, dato il suo contenuto, costituisce un'opera coperta. Questa licenza riconosce i tuoi diritti di fair use o altro equivalente, come previsto dalla legge sul copyright.

Puoi creare, eseguire e propagare opere coperte che non trasmetti, senza condizioni purché la tua licenza rimanga in vigore. Puoi trasmettere le opere coperte ad altri al solo scopo di fargli apportare modifiche esclusivamente per te, o fornirti le strutture per eseguire quelle opere, a condizione che tu rispetti i termini di questa licenza nel trasmettere tutto il materiale per il quale non controlli diritto d'autore. Coloro che quindi realizzano o eseguono le opere coperte per te devono farlo esclusivamente per tuo conto, sotto la tua direzione e controllo, a condizioni che vietano loro di fare copie del tuo materiale protetto da copyright al di fuori del loro rapporto con te.

Il trasporto in qualsiasi altra circostanza è consentito esclusivamente alle condizioni indicate di seguito. La sublicenza non è consentita: la sezione 10 lo rende inutile. 3. Protezione dei diritti legali degli utenti dalla legge antitelusione.

Nessun lavoro coperto sarà considerato parte di una misura tecnologica efficace ai sensi di qualsiasi legge applicabile che adempia agli obblighi di cui all'articolo 11 del trattato sul diritto d'autore dell'OMPI adottato il 20 dicembre 1996, o leggi simili che vietano o limitano l'esigenza di tali misure.

Quando si trasmette un'opera coperta, si rinuncia a qualsiasi potere legale per vietare l'elusione di misure tecnologiche nella misura in cui tale elusione viene effettuata esercitando i diritti previsti dalla presente Licenza in relazione all'opera coperta e si declina qualsiasi intenzione di limitare il funzionamento o la modifica dell'opera come mezzo per far valere, nei confronti degli utenti dell'opera, i diritti legali vostri o di terzi di vietare l'elusione delle misure tecnologiche. 4. Trasmissione di copie letterali.

È possibile trasmettere copie letterali del codice sorgente del Programma così come lo si riceve, con qualsiasi mezzo, a condizione di pubblicarle in modo evidente e appropriato su ciascuna copia un avviso di copyright appropriato; mantenere intatti tutti gli avvisi che affermano che questa Licenza è qualsiasi termine non permesso aggiunto in accordo con la sezione 7 si applica al codice; mantenere intatti tutti gli avvisi di assenza di qualsiasi garanzia; e dare a tutti i destinatari una copia di questa licenza insieme al programma.

Puoi addebitare qualsiasi prezzo o nessun prezzo per ogni copia che trasmetti e puoi offrire supporto o protezione in garanzia a pagamento. 5. Trasmissione di versioni sorgente modificate.

È possibile trasmettere un lavoro basato sul Programma, o le modifiche per prodotto dal Programma, sotto forma di codice sorgente secondo i termini della sezione 4, a condizione che soddisfi anche tutte queste condizioni:

a) L'opera deve riportare avvisi ben visibili che affermano che è stata modificata e che indica una data pertinente. * b) Il lavoro deve riportare avvisi in evidenza che affermano che è rilasciato sotto questa Licenza e qualsiasi condizione aggiunta nella sezione 7.

Questo requisito modifica il requisito nella sezione 4 di "mantenere intatti tutti gli avvisi". * c) È necessario concedere in licenza l'intera opera, nel suo insieme, con questa licenza a chiunque ne venga in possesso di una copia. La presente licenza si applicherà quindi, insieme a tutti i termini aggiuntivi applicabili della sezione 7, all'intera opera ed alle sue parti, indipendentemente da come sono imballate. Questa licenza non consente di concedere in licenza l'opera in altro modo, ma non invalida tale autorizzazione se è stata ricevuta separatamente. *

d) Se l'opera dispone di interfacce utente interattive, ciascuna deve visualizzare le opportune note legali. Tuttavia, se il programma dispone di interfacce interattive che non visualizzano Avvisi legali appropriati, non è necessario che il tuo lavoro lo faccia.

Una compilazione di un'opera coperta con altre opere separate e indipendenti, che per loro natura non sono estensioni dell'opera coperta e che non sono combinate con essa in modo da formare un programma più ampio, in o su un volume di un archivio o di una distribuzione mezzo, è chiamato "aggregato" se la compilazione e il diritto d'autore che ne deriva non vengono utilizzati per limitare l'accesso e i diritti legali degli utenti della compilazione oltre a quanto consentito dalle singole opere. L'inclusione di un'opera coperta in un aggregato non fa sì che questa licenza si applichi alle altre parti dell'aggregato. 6. Trasmissione di moduli non di origine.

È possibile trasmettere un'opera coperta sotto forma di codice oggetto secondo i termini delle sezioni 4 e 5, a condizione che trasmetta anche la Sorgente di correzione legibile dalla macchina secondo i termini di questa Licenza, in uno dei seguenti modi:

* a) Trasmettere il codice oggetto in, o incorporato in, un prodotto fisico (incluso un supporto di distribuzione fisico), accompagnato dalla sorgente corrispondente fissata su un supporto fisico durevole usato abitualmente per l'interconnessione software. * b) Trasmetti il codice oggetto in, o incartato in, un prodotto fisico (incluso un mezzo di distribuzione fisico), accompagnato da un'offerta scritta, valida per almeno tre anni e valida per il tempo in cui ci offri pezzi di ricambio o assistenza clienti per quel modello di prodotto, per fornire a chiunque possieda il codice oggetto o (1) una copia della sorgente corrispondente per tutto il software nel prodotto coperto da questa licenza, su un supporto fisico durevole usato abitualmente per l'interscambio di software, per un prezzo non più del costo ragionevole per eseguire fisicamente questo trasferimento della fonte.

* c) Trasmettere copie individuali del codice oggetto con una copia dell'offerta scritta per fornire la fonte corrispondente. Questa alternativa è consentita solo occasionalmente e non a scopo commerciale e solo se hai ricevuto il codice oggetto con tale offerta, in conformità con la sottosezione 6b. * d) Trasmettere il codice oggetto offrendo l'accesso da un luogo designato (gratuitamente o a pagamento) e offrire un accesso equivalente alla fonte corrispondente nello stesso modo attraverso lo stesso luogo senza ulteriori costi. Non è necessario richiedere ai destinatari di copiare l'origine corrispondente insieme al codice oggetto. Se il luogo in cui copiare il codice

Una parte separabile del codice oggetto, il cui codice sorgente è escluso dalla sorgente corrispondente come libreria di sistema, non deve essere inclusa nella trasmissione del lavoro del codice oggetto.

Un "Prodotto utente" è (1) un "prodotto di consumo", il che significa qualsiasi proprietà personale tangibile normalmente utilizzata per scopi personali, familiari o domestici, o (2) qualsiasi cosa progettata o venduta per essere incorporata in un'abitazione. Nel determinare se un prodotto è un prodotto di consumo, i casi dubbi devono essere risolti a favore della copertura. Per un particolare prodotto ricevuto da un particolare utente, "normalmente utilizzato" si riferisce a un uso tipico o comune di quella classe di prodotto, indipendentemente dallo stato del particolare utente o dal modo in cui il particolare utente utilizza effettivamente, e si aspetta o è previsto che utilizzi il prodotto. Un prodotto è un prodotto di consumo indipendentemente dal fatto che il prodotto abbia usi commerciali, industriali o non di consumo sostanziali, a meno che tali usi non rappresentino l'unica modalità d'uso significativa del prodotto.

"Informazioni di installazione" per un prodotto utente indica qualsiasi metodo, procedura, chiavi di autorizzazione o altre informazioni necessarie per installare ed eseguire versioni modificate di un'opera coperta in quel prodotto utente da una versione modificata della sua fonte corrispondente. Le informazioni devono essere sufficienti per garantire che il funzionamento continuo del codice oggetto modificato non sia in nessun caso impedito o interferito solo perché è stata apportata una modifica.

Se si trasmette un lavoro in codice oggetto in questa sezione in, o con, o specificamente per l'uso in un prodotto dell'utente, e il trasporto avviene come parte di una transazione in cui viene trasferito il diritto di possesso e utilizzo del prodotto dell'utente ai destinatari in perpetuo o in tempo determinato (indipendentemente da come si caratterizza la transazione), la corrispondente fonte comunicata in questa sezione deve essere accompagnata dalle informazioni di installazione. Ma questo requisito non si applica se né tu né terze parti conservate la possibilità di installare codice oggetto modificato sul Prodotto utente (ad esempio, il lavoro è stato installato nella ROM).

Il requisito di fornire le informazioni di installazione non include un requisito per continuare a fornire servizio di supporto, garanzia o aggiornamenti per un lavoro che è stato modificato o installato dal destinatario o per il prodotto difluente in cui è stato modificato o installato. L'accesso a una rete può essere negato quando la modifica stessa influisce materialmente e negativamente sul funzionamento della rete o viola le regole e i protocolli per la comunicazione attraverso la rete.

La sorgente corrispondente trasmessa e le informazioni di installazione fornite, in accordo con questa sezione, devono essere in un formato documentato pubblicamente (e con un'implementazione disponibile al pubblico sotto forma di codice sorgente) e non devono richiedere password o chiavi speciali per decodificare, leggere o copy-ing. 7. Termini aggiuntivi.

"Autorizzazioni aggiuntive" sono termini che integrano i termini di questa Licenza facendo eccezioni da una o più delle sue condizioni. Le autorizzazioni aggiuntive applicabili all'intero programma devono essere tratte come se fossero incluse in questa licenza, nella misura in cui sono valide ai sensi delle leggi applicative. Se autorizzazioni aggiuntive si applicano solo a una parte del programma, quella parte può essere utilizzata separatamente in base a tali autorizzazioni, ma l'intero programma rimane regolato da questa licenza indipendentemente dalle autorizzazioni aggiuntive.

Quando si trasmette una copia di un'opera coperta, è possibile, a propria discrezione, rimuovere eventuali autorizzazioni aggiuntive da quella copia o da qualsiasi parte di essa. (È possibile scrivere autorizzazioni aggiuntive per richiedere la loro rimozione in alcuni casi quando si modifica il lavoro.) È possibile assegnare autorizzazioni aggiuntive sul materiale, aggiunto da te a un'opera coperta, per il quale si dispone o si può concedere l'autorizzazione di copyright appropriata.

Nonostante qualsiasi altra disposizione di questa Licenza, per il materiale che aggiungi a un'opera coperta, puoi (se autorizzato dai detentori del copyright di quel materiale) integrare i termini di questa Licenza con termini:

- a) Esoneri di garanzia o limitazioni di responsabilità diversamente dai termini delle sezioni 15 e 16 di questa Licenza; o
- b) Richiedere la conservazione di note legali ragionevoli specificate o attribuzioni dell'autore in quel materiale o negli Avvisi legali appropriati visualizzati dalle opere che lo contengono; o
- c) Proibire la falsa dichiarazione dell'origine di quel materiale, o richiedere che le versioni modificate di tale materiale siano contrassegnate in modi ragionevoli come differenti dalla versione originale; o
- d) Limitare l'uso a fini pubblicitari di nomi di finanziari e autori del materiale; o
- e) Rifiutare di concedere diritti ai sensi della legge sui marchi per l'uso di alcuni nomi commerciali, marchi di fabbrica o marchi di servizio;

Tutti gli altri termini aggiuntivi non permisivi sono considerati "ulteriori restrizioni" ai sensi della sezione 10. Se il programma come lo avete ricevuto, o parte di esso, contiene un avviso in cui si afferma che è regolato da questa licenza insieme a un termine che è un'ulteriore restrizione, puoi rimuovere quel termine. Se un documento di licenza contiene un'ulteriore restrizione che consente la nuova licenza o il trasferimento ai sensi della presente licenza, è possibile aggiungere a un materiale di lavoro coperto regolato dai termini di tale documento di licenza, a condizione che l'ulteriore restrizione non sopravviva a tale rilascio o trasferimento di licenza.

Se si aggiungono termini a un'opera coperta in accordo con questa sezione, è necessario inserirli nei file di origine pertinenti, una dichiarazione dei termini aggiuntivi che si applicano a tali file o un avviso che indichi dove trovare i termini applicabili.

Termini aggiuntivi, permissivi o non permissivi, possono essere dichiarati sotto forma di una licenza scritta separatamente o indicati come eccezioni; i requisiti di cui sopra si applicano in entrambi i casi. 8. Risoluzione.

Non è possibile propagare o modificare un'opera coperta salvo quanto esplicitamente previsto dalla presente licenza. Qualsiasi tentativo altrimenti di propagare o modificarlo è nullo e terminerà automaticamente i tuoi diritti ai sensi della presente licenza (compresa eventuali licenze di brevetto concesse ai sensi del terzo paragrafo della sezione 11).

Tuttavia, se cessi ogni violazione di questa licenza, la tua licenza da un particolare titolare del copyright viene ripristinata (a) provvisoriamente, a meno che e fino a quando il titolare del copyright non risolve esplicitamente e definitivamente la tua licenza, e (b) in modo permanente, se il titolare del copyright fallisce per informarti della violazione con mezzi ragionevoli prima di 60 giorni dopo la cessazione.

Inoltre, la tua licenza da un particolare detentore del copyright viene ripristinata in modo permanente se il detentore del copyright ti notifica la violazione con qualche mezzo ragionevole, questa è la prima volta che hai ricevuto avviso di violazione di questa licenza (per qualsiasi opera) da il titolare del copyright e tu rimedi alla violazione entro 30 giorni dalla ricezione della notifica.

La cessazione dei tuoi diritti ai sensi di questa sezione non comporta la revoca delle licenze delle parti che hanno ricevuto copie o diritti da te in base a questa licenza. Se i tuoi diritti sono stati revocati e non sono stati ripristinati in modo permanente, non sei idoneo a ricevere nuove licenze per lo stesso materiale ai sensi della sezione 10. 9. Accettazione non richiesta per avere copie.

Non è necessario accettare questa licenza per ricevere o eseguire una copia del programma. Analogamente, la propagazione accessoria di un'opera coperta che si verifica esclusivamente come conseguenza dell'utilizzo della trasmissione peer-to-peer per ricevere una copia non richiede l'accettazione. Tuttavia, nient'altro che questa licenza concede

il tuo permesso di propagare o modificare qualsiasi lavoro coperto. Queste azioni sono in violazione del copyright se non si accetta questa licenza. Pertanto, modificando o propagando un'opera coperta, indichi la tua accettazione di questa licenza per farlo. 10. Licenza automatica dei destinatari a valle.

Ogni volta che si trasmette un'opera coperta, il destinatario riceve automaticamente una licenza dai licenziati originali, per eseguire, modificare e propagare quell'opera, soggetta a questa licenza. Non sei responsabile del rispetto di questa licenza da parte di terzi.

Una "transazione di entità" è una transazione che trasferisce il controllo di un'organizzazione o sostanzialmente tutte le risorse di una, o che suddivide un'organizzazione o che unisce organizzazioni. Se la propagazione di un'opera coperta risulta da una transazione dell'entità, ciascuna parte di quella transazione che riceve una copia dell'opera riceve anche qualsiasi licenza per l'opera che il predecessore della parte interessata aveva o poteva dare ai sensi del paragrafo precedente, più il diritto di possesso della fonte corrispondente del lavoro dal predecessore interessato, se il predecessore ce l'ha o può ottenerlo con sforzi ragionevoli.

Non è possibile imporre ulteriori restrizioni all'esercizio dei diritti concessi o affermati in base a questa licenza. Ad esempio, non è possibile imporre un canone di licenza, royalty o altri addebiti per l'esercizio dei diritti concessi ai sensi della presente Licenza, e non è possibile avviare contenzioso (inclusa una rivendicazione incrociata o una controprestazione in una causa legale) sostenendo che qualsiasi rivendicazione di brevetto viene violato facendo, utilizzando, vendendo, offrendo in vendita o importando il programma o parte di esso. 11. Brevetti.

Un "collaboratore" è un detentore del copyright che autorizza l'uso in base a questa Licenza del Programma o di un'opera su cui si basa il Programma. Il lavoro così concesso in licenza è chiamato "versione collaboratore" del contributore.

Le "rivendicazioni di brevetto essenziali" di un collaboratore sono tutte le rivendicazioni di brevetto possedute o controllate dal collaboratore, già acquisite o acquisite in seguito, che sarebbero violate in qualche modo, consentito dalla presente Licenza, di ma-re, utilizzando o vendendo la sua versione di contributore , ma non includono affermazioni che sarebbero state violate solo come conseguenza di ulteriori modifiche della versione contraria. Ai fini di questa definizione, "controllo" include il diritto di concedere licenze secondarie di brevetto in modo coerente con i requisiti di questa licenza.

Ogni collaboratore ti concede una licenza di brevetto non esclusiva, mondiale, esente da royalty in base alle rivendicazioni di brevetto essenziali del contributore, per fare, utilizzare, vendere, offrire in vendita, importare e in altro modo eseguire, modificare e propagare i contenuti della sua versione per contributore .

Nei tre paragrafi seguenti, una "licenza di brevetto" è qualsiasi accordo o impegno espresso, comunque denominato, a non far valere un brevetto (come un permesso esplicito a praticare un brevetto o una convenzione di non citare in giudizio per violazione di brevetto). Per "concedere" una tale licenza di brevetto a una parte significa fare un tal accordo o impegno a non far valere un brevetto contro la parte.

Se trasmetti un'opera coperta, affidandoti consapevolmente a una licenza di brevetto, e la fonte corrispondente dell'opera non è disponibile per la copia gratuita e secondo i termini di questa licenza, attraverso un server di rete pubblicamente disponibile o altro facilmente accessibile significa, quindi è necessario (1) rendere disponibile la fonte corrispondente, o (2) fare in modo di privarsi del beneficio della licenza di brevetto per questo particolare lavoro, o (3) ar-range, in modo coerente con i requisiti di questa licenza, per estendere la licenza di brevetto al destinatario a valle. "Affidarsi consapevolmente" significa che hai la conoscenza effettiva che, tranne per la licenza di brevetto, il trasferimento del lavoro coperto in un paese o fuso del lavoro coperto da parte del destinatario in un paese.

Se, ai sensi o in relazione a una singola transazione o accordo, trasmetti o propghi procurando il trasferimento di un'opera coperta e concedi una licenza di brevetto ad alcune delle parti che ricevono l'opera coperta autorizzandole a utilizzare, propagare, modificare o trasmettere una copia specifica dell'opera coperta, quindi la licenza di brevetto concessa viene automaticamente estesa a tutti i destinatari dell'opera coperta e delle opere basate su di essa.

Una licenza di brevetto è "discriminatoria" se non include nell'ambito della sua copertura, vieta l'esercizio o è condizionata al non esercizio di uno o più dei diritti specificatamente concessi in base a questa licenza. Non puoi trasmettere un'opera coperta se sei una parte di un accordo con una terza parte che si occupa di distribuzione di software, in base alla quale effetti il pagamento alla terza parte in base alla portata della tua attività di trasmissione del lavoro, e in base alla quale la terza parte concede, a una qualsiasi delle parti che riceverebbero il lavoro coperto da te, una licenza di brevetto discriminatorio (a) in relazione alle copie dell'opera coperta da te trasmesse (o copie fatte da tali copie), o (b) principalmente per e in connessione con prodotti o raccolte specifici che contengono l'opera coperta.

Niente nella presente Licenza deve essere interpretato come escludente o limitante qualsiasi licenza implicita o altre difese contro la violazione che potrebbero essere altrimenti disponibili ai sensi della legge sui brevetti applicabile. 12. Nessuna resa della libertà altri.

Se ti vengono imposte condizioni (sia per ordine del tribunale, accordo o altro) che contraddicono le condizioni di questa licenza, non ti esonerano dalle condizioni di questa licenza. Se non puoi trasmettere un'opera coperta in modo da soddisfare contemporaneamente i tuoi obblighi ai sensi della presente Licenza e qualsiasi altro obbligo pertinente, di conseguenza potresti non trasmettere una royalty per ulteriore trasmissione da parte di coloro a cui si trasmette il programma, l'unico modo per soddisfare entrambi i termini e questa licenza sarebbe astenersi completamente dal trasmettere il programma. 13. Utilizzare con la GNU Affero General Public License.

Nonostante qualsiasi altra disposizione di questa licenza, hai il permesso di collegare o combinare qualsiasi lavoro coperto con un lavoro concesso in licenza in base alla versione 3 della GNU Affero General Public License in un'opera combinata e di trasmettere il lavoro risultante. I termini di questa Licenza continueranno ad applicarsi alla parte che è l'opera coperta, ma i requisiti speciali della GNU Affero General Public License, sezione 13, riguardanti l'interazione attraverso una rete si applicheranno alla combinazione in quanto tale. 14. Versioni inuite di questa licenza.

La Free Software Foundation può pubblicare periodicamente versioni riviste e / o nuove della GNU General Public License. Tali nuove versioni saranno simili nello spirito alla versione attuale, ma potrebbero differire nei dettagli per affrontare nuovi problemi o preoccupazioni.

A ciascuna versione viene assegnato un numero di versione distintivo. Se il Programma specifica che ad esso si applica una certa versione numerata della GNU General Public License "qualsiasi versione successiva", ha la possibilità di seguire i termini e le condizioni di quella versione numerata o di qualsiasi versione successiva pubblicata dalla Free Software Foundation. Se il Programma non specifica un numero di versione della GNU General Public License, puoi scegliere qualsiasi versione pubblicata dalla Free Software Foundation.

Se il programma specifica che un proxy può decidere quali versioni future della GNU General Public License possono essere utilizzate, la dichiarazione pubblica di quel proxy di

L'accettazione di una versione ti autorizza in modo permanente a scegliere quella versione per il programma.

Le versioni successive della licenza potrebbero fornire autorizzazioni aggiuntive o diverse. Tuttavia, non sono imposti obblighi aggiuntivi a nessun autore o detentore del copyright a seguito della tua scelta di seguire una versione successiva. 15. Esclusione di garanzia.

NESSUNA GARANZIA PER IL PROGRAMMA, NELLA MISURA CONSENTITA DALLA LEGGE VIGENTE, SALVO QUANDO ALTRIMENTI DICHIARATO PER SCRITTO I TITOLARI DEL COPYRIGHT E / O ALTRE PARTI FORNISCONO IL PROGRAMMA COSÌ COM'È SENZA GARANZIA DI ALCUN TIPO, ESPlicita o IMPLICITA, INCLUSE, MA NON SOLO, LE GARANZIE IMPLICITE DI PARTICOLARE COMMERCIALITÀ E IDONEITÀ . L'INTERO RISCHIO RELATIVO ALLA QUALITÀ E ALLE PRESTAZIONI DEL PRO-GRAMM È A CARICO DELL'UTENTE. SE IL PROGRAMMA RISULTA DEFETTUOSO, VI ASSUMETTE IL COSTO DI TUTTI GLI INTERVENTI DI MANUTENZIONE, RIPARAZIONE O CORREZIONE NECESSARI. 16. Limitazione di responsabilità.

IN NESSUN CASO SE NON PREVISTO DALLA LEGGE VIGENTE O ACCETTATO PER SCRITTO, QUALSIASI TITOLARE DEL COPYRIGHT, O QUALSIASI ALTRA PARTE CHE MODIFICA E / O TRASMETTE IL PRO-

GRAM COME SOPRA CONSENTITO, ESSERE RESPONSABILE NEI CONFRONTI DI DA-MAGES, INCLUSI QUALSIASI DANNO GENERALE, SPECIALE, INCIDENTALE O CONSEQUENZIALE DERIVANTE DALL'UTILIZZO O DALL'IMPATIBILITÀ DI UTILIZZARE IL PROGRAMMA (COMPRESA MA NON LIMITATA ALLA PERDITA DI DATI O DATI DA RENDERE INACCURATE O PERDITE SOSTENUTE DA VOI O DA TERZI O UN MANCATO FUNZIONAMENTO DEL PROGRAMMA CON QUALSIASI ALTRI PROGRAMMI), ANCHE SE TALE TITOLARE O ALTRA PARTE È STATA AVVISATA DELLA POSSIBILITÀ DI TALI DANNI. 17. Interpretazione delle sezioni 15 e 16.

Se il disclaimer di garanzia e limitazione di responsabilità di cui sopra non può avere effetto legale locale secondo i loro termini, i tribunali di revisione applicheranno la legge locale che si avvicina più da vicino a una rinuncia assoluta a tutte le responsabilità civili in relazione al programma, a meno che una garanzia o l'assunzione di responsabilità accompagni una copia del programma a pagamento.

FINE DEI TERMINI E DELLE CONDIZIONI. Come applicare i presenti Termini ai nuovi programmi

Se sviluppi un nuovo programma e vuoi che sia di grande utilità per il pubblico, il modo migliore per ottenere questo risultato è renderlo un software libero che tutti possono ridistribuire e modificare in questi termini.

A tal fine, allegare al programma le seguenti comunicazioni. È sicuro allegarli all'inizio di ogni file sorgente per dichiarare in modo più efficace l'esclusione della garanzia; e ogni file dovrebbe avere almeno la riga "copyright" e un puntatore al punto in cui si trova l'avviso completo.

<una riga per dare il nome del programma e una breve idea di cosa fa> Co-copyright (C) <year> <nome dell'autore>

Questo programma è un software libero: puoi ridistribuirlo e / o modificarlo secondo i termini della GNU General Public License come pubblicata dalla Free Software Foundation, versione 3 della licenza o (a tua scelta) qualsiasi versione successiva.

Questo programma è distribuito nella speranza che possa essere utile, ma SENZA ALCUNA GARANZIA; senza nemmeno la garanzia implicita di COMMERCIALITÀ o IDONEITÀ PER UN PARTICOLARE SCOPO. Vedere la GNU General Public License per maggiori dettagli.

Dovresti aver ricevuto una copia della GNU General Public License insieme a questo programma. In caso contrario, vedere <http://www.gnu.org/licenses/>.

Aggiungi anche informazioni su come contattarti tramite posta elettronica e cartacea.

Se il programma interagisce con il terminale, fai in modo che emetta un breve avviso come questo quando viene avviato in modalità interattiva:

<programma> Copyright (C) <anno> <nome dell'autore> Questo programma viene ASSOLUTAMENTE NESSUNA GARANZIA; per i dettagli digitare "show w". Questo è un software gratuito e sei libero di ridistribuirlo a determinate condizioni; digitare "mostra c" per i dettagli.

Gli ipotetici comandi "show w" e "show c" dovrebbero mostrare le parti appropriate della General Public License. Naturalmente, i comandi del tuo programma potrebbero essere diversi; per un'interfaccia GUI, dovresti usare un "riquadro informazioni".

Se necessario, dovresti anche chiedere al tuo datore di lavoro (se lavori come programmatore) o alla scuola, se esiste, di firmare una "dichiarazione di non responsabilità sul copyright" per il programma. Per ulteriori informazioni su questo e su come applicare e seguire la GNU GPL, vedere <http://www.gnu.org/licenses/>.

La GNU General Public License non consente di incorporare il proprio programma in programmi proprietari. Se il tuo programma è una libreria di subroutine, potresti considerare più utile consentire il collegamento di applicazioni proprietarie con la libreria. Se questo è ciò che vuoi fare, sarà la GNU Les-Ser General Public License invece di questa Licenza. Ma prima, leggi <http://www.gnu.org/philosophy/why-not-gpl.html>.

14.2 Licenza GNU Free Documentation

Versone 1.3, 3 novembre 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc. <http://fsf.org/>

A tutti è consentito copiare e distribuire copie letterali di questo documento di licenza, ma non è consentito modificarlo. 0. PREAMBOLO

Lo scopo di questa licenza è quello di rendere un manuale, un libro di testo o un altro documento funzionale e utile "liberando il senso di libertà: assicurare a tutti l'effettiva libertà di copiarlo e ridistribuirlo, con o senza modifica, commercialmente o non Commercialmente. Secondariamente, questa Licenza offre all'autore e all'autore un modo per ottenerne credito per il proprio lavoro, pur non essendo ritenuto responsabile per modifiche apportate da altri.

Questa licenza è una sorta di "copyleft", il che significa che le opere derivate del documento devono esse stesse essere libere nello stesso senso. Completa la GNU General Public License, che è una licenza copyleft progettata per il software libero.

Abbiamo progettato questa Licenza in modo da poterla utilizzare per manuali per software libero, perché il software libero necessita di documentazione gratuita: un programma gratuito dovrebbe avere manuali che forniscono le stesse libertà del software. Ma questa licenza non è limitata ai manuali del software; può essere utilizzato per qualsiasi lavoro testuale, indipendentemente dall'argomento o dal fatto che sia pubblicato come libro stampato. Raccomandiamo questa licenza principalmente per operi il cui scopo è l'istruzione o il riferimento. 1. APPLICABILITÀ E DEFINIZIONI

Questa Licenza si applica a qualsiasi manuale o altro lavoro, con qualsiasi mezzo, che contiene un avviso posto dal detentore del copyright che afferma che può essere distribuito secondo i termini di questa Licenza. Tale avviso garantisce una licenza mondiale, esente da royalty, di durata illimitata, per utilizzare quell'opera alle condizioni indicate in, "Il Documento", di seguito, si riferisce a qualsiasi manuale o opera di questo tipo. Qualsiasi membro del pubblico è un licenziario e viene chiamato "tu". Accetti la licenza se copi, modifichi o distribuischi il lavoro in un modo che richiede l'autorizzazione ai sensi della legge sul copyright.

Una "Versione modificata" del Documento indica qualsiasi opera contenente il Documento o una parte di esso, copiata alla lettera o con modifiche e / o tradotta in un'altra lingua.

Una Sezione Secondaria è un'appendice denominata o una sezione di frontespizio del Documento che si occupa esclusivamente del rapporto degli editori o autori del Documento con l'argomento complessivo del Documento (o con argomenti correlati) e non contiene nulla che possa rientrare direttamente nell'argomento generale. (Pertanto, se il Documento è in parte un libro di testo di matematica, una Sezione Secondaria potrebbe non spiegare alcuna matematica.) La relazione potrebbe essere una questione di connessione storica con l'argomento o con questioni correlate, o di natura giuridica, commerciale, filosofica, etica, o posizione politica riguardo a loro.

Le sezioni "varianti" sono alcune sezioni secondarie i cui titoli sono designati, come quelli delle sezioni non modificabili, nell'avviso che dice che il documento è rilasciato sotto questa licenza. Se una sezione non si adatta alla definizione di Secondario di cui sopra, non è consentita la designazione come Invariante. Il documento può contenere zero sezioni invariante. Se il documento non identifica sezioni non modificabili, non ce ne sono.

I "testi di copertina" sono alcuni brevi passaggi di testo che sono elencati, come testi di copertina anteriore o testi di copertina posteriore, nell'avviso che dice che il documento è rilasciato sotto questa licenza. Un testo di copertina anteriore può contenere al massimo 5 parole, e un testo di retrocopertina può contenere al massimo 25 parole.

Una copia "trasparente" del documento indica una copia leggibile da una macchina, riproposta in un formato la cui specifica è disponibile al pubblico in generale, che è adatto per la revisione del documento direttamente con editor di testo generici o (per i disegni, alcuni editori di disegni ampiamente disponibili, e che sono adatti per input ai formattatori di testo o per la traduzione automatica in una varietà di formati adatti per input ai formattatori di testo. Una copia realizzata in un formato di file altrimenti trasparente il cui markup, o assenza di markup, è stato predisposto per ostacolare o scoraggiare le successive modifiche da parte dei lettori non è trasparente. Un formato immagine non è trasparente se utilizzato per una quantità sostanziale di testo. Una copia che non sia "La trasparenza si chiama Opaque".

Esempi di formati adatti per copie trasparenti includono ASCII semplice senza markup, formato di input TexInfo, formato di input LaTeX, SGML o XML utilizzando un DTD disponibile pubblicamente e HTML semplice, PostScript o PDF conforme agli standard progettati per la modifica umana. Esempi di formati di immagine trasparenti includono PNG, XCF e JPG. I formati opachi includono formati proprietari che possono essere letti e modificati solo da elaboratori di testi proprietari, SGML o XML per i quali il DTD e / o gli strumenti di elaborazione non sono generalmente disponibili e l'HTML, il PostScript o il PDF generati dalla macchina prodotti da qualche parossioso solo per scopi di output.

Il "Frontespizio" significa, per un libro stampato, il frontespizio stesso, più le pagine seguenti che sono necessarie per contenere, in modo leggibile, il materiale che questa licenza richiede per apparire nel frontespizio. Per le opere in formati che non hanno un frontespizio in quanto tale, "Frontespizio" indica il testo vicino all'aspetto più evidente del titolo dell'opera, che precede l'inizio del corpo del testo.

Per "editore" si intende qualsiasi persona o entità che distribuisce copie del documento al pubblico.

Una sezione "intitolata XYZ" indica una subunità denominata del documento il cui titolo è precisamente XYZ o contiene XYZ tra parentesi dopo il testo che

traduce XYZ in un'altra lingua. (Qui XYZ sta per il nome di una sezione specifica menzionata di seguito, come "Ringraziamenti", "Dediche", "Enders-memo" o "Chronologia".) Per "Preservare il titolo di una sezione del genere quando si modifica il documento" significa che rimane un sezione intitolata XYZaccord-ding a questa definizione.

Il documento può includere dichiarazioni di non responsabilità sulla garanzia, accanto all'avviso in cui si afferma che questa licenza si applica al documento. Queste esclusioni di garanzia sono considerate incluse per riferimento in questa licenza, ma solo per quanto riguarda le esclusioni di garanzie; qualsiasi altra implicazione che queste esclusioni di garanzia possono avere è nulla e non ha alcun effetto sul significato di questa licenza. 2. COPIA VERBATIM

È possibile copiare e distribuire il Documento con qualsiasi mezzo, commercialmente o non commercialmente, a condizione che questa Licenza si applichi sul copiato. Il documento, può, a tua discrezione, designare alcune o tutte queste sezioni come invarianti. A tale scopo aggiungere i loro titoli all'interno delle sezioni non modificabili nell'avviso di licenza della versione modificata. Questi titoli devono essere distinti da qualsiasi altro titolo di sezione.

Puoi anche prestare copie, alle stesse condizioni sopra indicate, e puoi esporle pubblicamente. 3. COPIA IN QUANTITÀ

Se pubblicherai copie stampate (o copie su supporti che normalmente hanno coperte stampate) del Documento, numero più di 100, e l'avviso di licenza del Documento richiederà Testi di copertina, devi allegare le copie in copertina che riportino, in modo chiaro e leggibile tutti questi testi di copertina: testi di copertina anteriore sulla copertina anteriore e testi di copertina posteriore sulla copertina posteriore. Entrambe le copertine devono inoltre identificare il titolo in modo chiaro e leggibile come editore di questi copie. La copertina deve presentare il titolo completo con tutte le parole del titolo ugualmente prominenti e visibili. È possibile aggiungere altre materiali sulle copertine in aggiunta. Le copie con modificate alle copertine, purché preservino il titolo del Documento e soddisfino queste condizioni, può essere trattata come copia letterale per altri aspetti.

Se i testi richiesti per una delle copertine sono troppo voluminosi per addattarsi in modo leggibile, dovresti mettere i primi elencati (quanti più ragionevolmente adatti) sulla copertina effettiva e continuare il resto sulle pagine adiacenti.

Se pubblicherai copie opache del documento che contano più di 100, devi includere una copia trasparente leggibile dalla macchina insieme a ciascuna copia opaca, oppure indicare in o con ciascuna copia opaca un percorso di rete di computer da cui la rete generale - utilizzando public ha accesso per scaricare protocolli di rete a standard pubblico una copia trasparente completa del documento, priva di materiale aggiuntivo. Se utilizzi quest'ultima opzione, devi adottare misure ragionevolmente prudenti, quindi innanzi la distribuzione di copie opache in quantità, per assicurarti che questa copia trasparente rimanga così accessibile nel luogo indicato fino ad almeno un anno dopo l'ultima volta che distribuischi un Copia opaca (direttamente o tramite i tuoi agenti o rivenditori) di quell'edizione al pubblico.

È richiesto, ma non obbligatorio, contattare gli autori del documento prima di ridistribuire un numero elevato di copie, per dare loro la possibilità di fornire una versione aggiornata del documento. 4. MODIFICHE

È possibile copiare e distribuire una versione modificata del documento alle condizioni delle sezioni 2 e 3 di cui sopra, a condizione di rilasciare la versione modificata esattamente sotto questa licenza, con la versione modificata che ricopre il ruolo del documento, autorizzando a chi la distribuisce e Modifica della versione modificata a chi ne possiede una copia.

È possibile estirare dalla pagina del titolo (e sulle copertine, se presenti) un titolo diverso da quello del documento e da quelli delle versioni precedenti (che dovrebbero, se ne fossero, essere elencati nella sezione Storia del documento). È possibile utilizzare lo stesso titolo di una versione precedente se l'autore originale di quella versione ti permette. "B." Elencare nella pagina del titolo, in qualità di autori, ora o più persone o entità responsabili delle modifiche nella Versione Modificata, insieme ad almeno cinque dei principali autori del Documento (tutti i suoi principali autori, se ne ha meno di cinque), a meno che non ti liberino da questa richiesta. "C." Indicare nella pagina del titolo il nome dell'autore della versione modificata, in qualità di editore. "D." Conserva tutte le note sul copyright del documento. "E." Aggiungi un avviso di copyright appropriato per le modifiche adattate agli altri avvisi di copyright.

È possibile utilizzare le copertine anteriusi, come indicato nel testo di copertina anteriore o testi di copertina posteriore, se non è presente una sezione intitolata "Storia nel documento", creare una indicante il titolo, l'anno, i nuovi autori e l'autore del documento come indicato nella sua pagina del titolo, quindi aggiungere ad essa un elemento che descrive la versione modificata come indicato nella frase precedente. "F." Includere l'eventuale percorso di rete fornito nel Documento per l'accesso pubblico a una copia trasparente del Documento, e allo stesso modo i percorsi di rete indicati nel Documento per le versioni precedenti su cui era basato. Questi possono essere inseriti nella sezione "Chronologia". Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su cui era basato. Puoi omettere un percorso di rete per un'opera che è stata pubblicata almeno quattro anni prima del documento stesso, o se l'autore originale della versione è allo stesso modo le posizioni di rete fornite nel documento per le versioni precedenti su

14.3 GNU Lesser General Public License

LICENZA PUBBLICA GENERALE GNU LESSER

Questa versione della GNU Lesser General Public License incorpora il
termini e condizioni della versione 3 della GNU General Public License,
sup-
completato dalle autorizzazioni aggiuntive elencate di seguito. 0. Definizioni
aggiuntive
ons.

Versione 3, 29 giugno 2007

Copyright © 2007 Free Software Foundation, Inc. <<http://fsf.org/>>

A tutti è consentito copiare e distribuire copie letterali di questa licenza
documento, ma non è consentito modificarlo.

Come usato in, "questa Licenza" si riferisce alla versione 3 di GNU Lesser
General
Public License, e la "GNU GPL" si riferisce alla versione 3 di GNU General
Licenza pubblica.

"La Libreria" si riferisce a un'opera coperta disciplinata dalla
presente Licenza, diversa da un'Applicazione o un'Opera combinata
come definito di seguito.

Un "lavoro combinato" è un lavoro prodotto combinando o collegando
un'applicazione con la libreria. La particolare versione della libreria con cui è
stata realizzata l'Opera Combinata è anche chiamata "Versione Collegata".

Una "Applicazione" è qualsiasi lavoro che utilizza un'interfaccia
fornita dalla libreria, ma che non è altrimenti basata sulla libreria. La
definizione di una sottoclassificazione di una classe definita dalla libreria è
considerata una modalità di utilizzo di un'interfaccia fornita dalla
libreria.

La "Sorgente Corrispondente Minima" per un'Opera Combinata indica la
Sorgente Corrispondente per l'Opera Combinata, escluso qualsiasi codice
sorgente per porzioni dell'Opera Combinata che, considerate isolatamente, si
basano sull'Applicazione e non sulla Versione Collegata.

Il "Codice dell'applicazione corrispondente" per un'opera combinata indica il codice oggetto e / o il codice sorgente dell'applicazione, inclusi tutti i dati e programmi di utilità necessari per riprodurre il lavoro combinato dall'applicazione, ma escluse le librerie di sistema dell'opera combinata. 1. Eccezione alla Sezione 3 della GNU GPL.

Puoi trasmettere un lavoro coperto dalle sezioni 3 e 4 di questa Licenza senza essere vincolato dalla sezione 3 della GNU GPL. 2. Trasporto di versioni modificate.

Se modifichi una copia della libreria e, nelle tue modifiche, una struttura fa riferimento a una funzione o ai dati che devono essere forniti da un'applicazione che utilizza la struttura (diverso da un argomento passato quando la struttura viene invocata), allora puoi trasmettere una copia della versione modificata:

* a) ai sensi della presente Licenza, a condizione che si faccia uno sforzo in buona fede per garantire che, nel caso in cui un'applicazione non fornisca la funzione o i dati, la struttura continui a funzionare e ad eseguire qualsiasi parte del suo scopo rimanga significativa, o * b) sotto la GNU GPL, senza che nessuno dei permessi aggiuntivi di questa licenza sia applicabile a quella copia.

3. Codice oggetto che incorpora materiale dai file di intestazione della libreria.

La forma del codice oggetto di un'applicazione può incorporare materiale da un file header che fa parte della libreria. Puoi trasmettere tale codice oggetto sotto

termini di tua scelta, a condizione che, se il materiale incorporato non è limitato a parametri numerici, layout di strutture dati e funzioni di accesso, o piccole macro, funzioni inline e modelli (dieci o meno righe di lunghezza), esegui entrambe le seguenti operazioni:

* a) Dare un avviso ben visibile con ogni copia del codice oggetto che la libreria è utilizzata in essa e che la libreria e il suo utilizzo sono coperti da questa licenza. * b) Accompagnare il codice oggetto con una copia della GNU GPL e questo documento di licenza.

4. Opere combinate.

Puoi trasmettere un'Opera Combinata in base a termini di tua scelta che, presi insieme, di fatto non limitino la modifica delle parti del Libra-ry contenute nell'Opera Combinata e il reverse engineering per il debug di tali modifiche, se esegui anche ciascuna delle seguenti operazioni. :

* a) Con ogni copia dell'Opera combinata, comunica chiaramente che la Biblioteca è utilizzata in essa e che la Biblioteca e il suo utilizzo sono coperti da questa licenza. * b) Accompagnare il lavoro combinato con una copia della GNU GPL e questo documento di licenza. * c) Per un'opera combinata che mostra avvisi di copyright durante l'esecuzione, includere l'avviso di copyright per la Libreria tra questi avvisi, così come un riferimento che indirizza l'utente alle copie della GNU GPL e di questo documento di licenza. * d) Effettuare una delle seguenti operazioni: o () Trasmettere la Sorgente Minima Corrispondente secondo i termini di questa Licenza e il Codice dell'Applicazione Corrispondente in una forma adatta e

in termini che consentono, futute di ricombinare o ricollegare l'applicazione con una versione modificata della versione collegata per produrre un'opera combinata modificata, nel modo specificato dalla sezione 6 della GNU GPL per trasmettere la fonte corrispondente. o 1) Utilizzare un meccanismo di libreria condivisa adatto per eseguire il collegamento con la libreria. Un meccanismo adatto è quello che (a) utilizza in fase di esecuzione una copia della libreria già presente sul sistema informatico dell'utente e (b) funzionerà correttamente con una versione modificata della libreria che sia compatibile con l'interfaccia con la versione collegata. * e) Fornisci informazioni sull'installazione, ma solo se ti verrà richiesto di fornire tali informazioni secondo la sezione 6 della GNU GPL, e solo nella misura in cui tali informazioni sono necessarie per installare ed eseguire una versione modificata dell'Opera combinata prodotta ricombinando o ricollegando l'applicazione con una versione modificata della Versione collegata. (Se si utilizza l'opzione 4d), le informazioni sull'installazione devono accompagnare la Sorgente corrispondente minima e il Codice dell'applicazione corrispondente. Se si utilizza l'opzione 4d, è necessario fornire le informazioni sull'installazione nel modo specificato dalla sezione 6 della GNU GPL per trasmettere la Sorgente corrispondente.)

5. Biblioteche combinate.

È possibile collocare le strutture della biblioteca che sono un'opera basata sulla biblioteca fianco a fianco in una singola libreria insieme ad altre strutture della biblioteca che non sono applicazioni e non sono coperte da questa licenza, e trasmettere tale libreria combinata secondo i termini di tua scelta, se esegui entrambe le seguenti operazioni:

* a) Accompagnare la libreria combinata con una copia della stessa opera basata sulla libreria, non combinata con altre strutture della biblioteca, trasmessa secondo i termini di questa licenza. * b) Dare un avviso evidente con la libreria combinata che parte di essa è un'opera basata sulla libreria e spiegare dove trovare la forma non combinata di accompagnamento della stessa opera.

6. Versioni riviste della GNU Lesser General Public License.

La Free Software Foundation può pubblicare periodicamente versioni riviste e / o nuove della GNU Lesser General Public License. Tali nuove versioni saranno simili nello spirito alla versione attuale, ma potrebbero differire nei dettagli per affrontare nuovi problemi o preoccupazioni.

A ciascuna versione viene assegnato un numero di versione distintivo. Se la Libreria così come l'hai ricevuta specifica che ad essa si applica una certa versione numerata della GNU Lesser General Public License "o qualsiasi versione successiva", hai la possibilità di seguire i termini e le condizioni di quella versione pubblicata o di qualsiasi versione successiva pubblicata dalla Free Software Foundation. Se la Libreria come l'hai ricevuta non specifica un numero di versione della GNU Lesser General Public License, puoi scegliere qualsiasi versione della GNU Lesser General Public License mai pubblicata dalla Free Software Foundation.

Se la Libreria così come è stata ricevuta specifica che un proxy può decidere se applicare le versioni future della GNU Lesser General Public License, la dichiarazione pubblica di accettazione di qualsiasi versione di tale proxy è un'autorizzazione permanente per te a scegliere quella versione per la Libreria.