

Guida all'uso di zuclib versione 3.X-

Installazione della libreria zuclib 3.X

Java è un linguaggio ricco, con grandissime potenzialità. Tuttavia richiede un tempo di apprendimento abbastanza lungo che impedisce agli adolescenti sia di risolvere immediatamente problemi molto elementari, sia di esprimere subito la propria creatività nelle applicazioni informatiche che più li attirano: la grafica e i giochi, come dimostra il grande successo di Scratch, un linguaggio per molti aspetti opposto a Java.

Per questo motivo viene messa a disposizione una libreria Java con di rendere più facile il primo approccio con Java, con questi obiettivi:

- mascherare il problema della gestione delle eccezioni e degli eventi;
- rendere facile l'uso di ritardi, la generazione di numeri casuali, la formattazione di dati;
- rendere facile la scrittura di programmi che utilizzano la grafica, il mouse e la tastiera;
- riutilizzare alcune delle competenze sugli sprite di Scratch recuperando la nozione di Tartaruga, come esempio di oggetto.

Installazione della libreria zuclib 3.X

La libreria zuclib versione è interamente contenuta in un file compresso **zuclib3.X.jar** (circa 20 kb). Nelle informazioni che seguono X è una cifra rappresenta la sottoversione (1, 2, 3, ecc.) Come per tutte le librerie aggiunte occorre informare il compilatore (javac) e il runtime java (java) della sua esistenza

Primo metodo (farla diventare una libreria interna)

Il modo più semplice per installare la libreria è quello di copiarla nella seguente sottocartella del jdk utilizzato dal vostro ambiente di sviluppo: **jre/lib/ext**

Secondo metodo (usarla come libreria esterna)

Se questo non è possibile, perché la cartella del jdk è modificabile solo dagli amministratori potete usare questo modo:

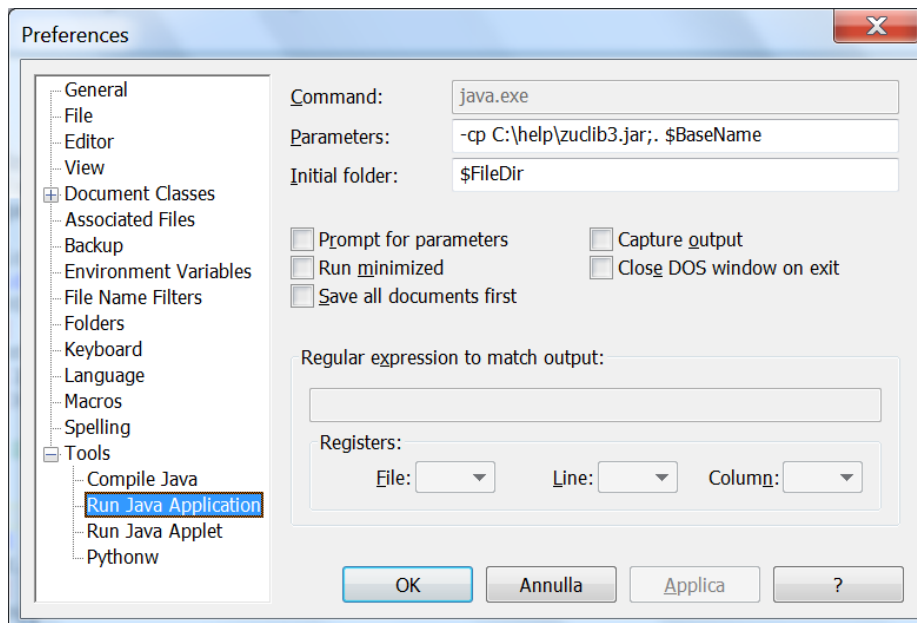
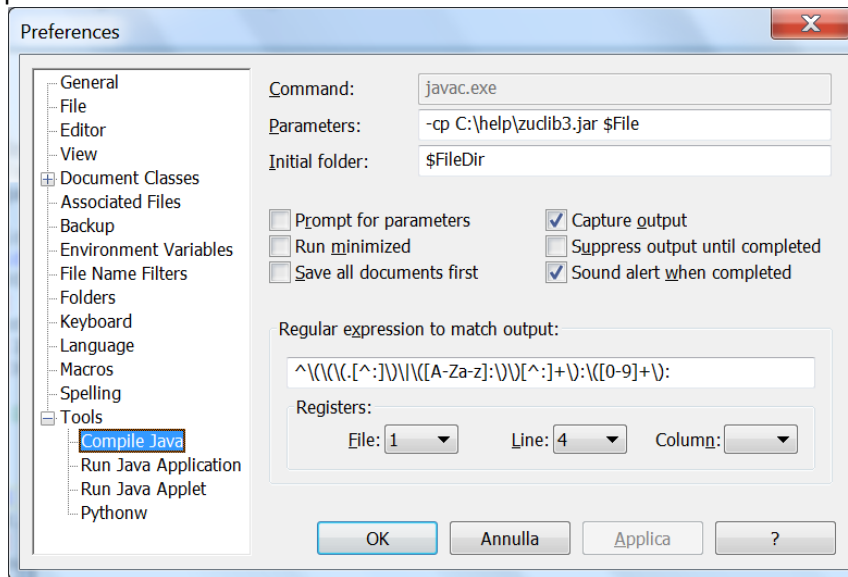
- 1) copiate il file `cp zuclib3.X.jar` dove nella cartella in cui si trova il sorgente del vostro *Programma.java*
- 2) compilate i vostri programmi con: `javac -cp zuclib3.X.jar Programma.java`
- 3) eseguite con: `java -cp zuclib3.X.jar;. Programma`

Se il programma crea una finestra grafica, per terminare il programma è necessario chiudere la finestra.

Sulla base di questi comandi è anche possibile configurare correttamente gli ambienti IDE per lo sviluppo di programmi java in modo che sia facile compilare ed eseguire programmi java che facciano uso della zuclib.

Secondo metodo applicato a TextPad

In particolare, l'ambiente TextPad consente lo sviluppo molto semplice di programmi java con zuclib utilizzando impostazioni come le seguenti, basate sull'idea di collocare la libreria, in unica copia, in una particolare cartella:



Naturalmente l'aiuto migliore per il programmatore si ha con l'uso di IDE moderni come Eclipse e Netbeans.

Struttura di un programma che usa zuclib

```
import static zuclib.Varie.*; //per leggere/scrivere, generare interi casuali, ritardi, op. su stringhe,
import static zuclib.GraficaSemplice.*; // se si usa la grafica
import zuclib.Tartaruga.*; //se si usano più tartarughe
```

```
public class NomeProgramma {
    public static void main(String[] args) {
        istruzione;
        istruzione;
        ...;
        istruzione;
    }
}
```

Operazioni della classe Tartaruga

Costruttori

[Tartaruga\(\)](#)

Crea una Tartaruga al centro (0.5,0.5), orientata verso l'alto (90 gradi), pennello abbassato di colore NERO, grossezza di default, ritardo 0.

[Tartaruga\(double x0, double y0, double alfa, Color c\)](#)

Crea una Tartaruga su (x0,y0), orientata di alfa gradi (in senso antiorario, rispetto all'asse X), colore c, pennello abbassato, grossezza di default, ritardo 0

Uso dei costruttori nei programmi

```
Tartaruga nome = new Tartaruga( ... )
```

Comandi di movimento

[avanti\(double passo\)](#)

Spostati in avanti nella direzione corrente.

[indietro\(double passo\)](#)

Spostati indietro nella direzione corrente.

[gotoXY\(double x, double y\)](#)

Porta la tartaruga nel punto (x,y).

[pennaGiu\(\)](#)

Abbassa il pennello.

[pennaSu\(\)](#)

Alza il pennello.

[setAngolo\(double angolo\)](#)

Imposta l'angolo verso cui sarà orientata la tartaruga.

[setColore\(Color c\)](#)

Imposta un dato colore al pennello della tartaruga.

[setGrossezza\(double gross\)](#)

Imposta una data grossezza al pennello della tartaruga.

[setRitardoMillisec\(int n\)](#)

Ritarda l'esecuzione di ogni comando avanti, indietro, destra, sinistra, gotoXY di n millisecondi.

Osservatori

double [getAngolo\(\)](#)
Angolo verso cui è orientata la tartaruga.

Color [getColore\(\)](#)
Colore corrente del pennello.

double [getGrossezza\(\)](#)
Grossezza corrente del pennello.

int [getRitardo\(\)](#)
Ritardo corrente per i comandi del pennello.

double [getX\(\)](#)
Coordinata X del pennello.

double [getY\(\)](#)
Coordinata Y del pennello.

boolean [isPennaSu\(\)](#)
Il pennello è alzato ?

Operazioni della classe **GraficaSemplice**

Fornisce costanti e operazioni per la creazione relativamente facile di semplici applicazioni grafiche anche con uso di mouse e tastiera.

Colori in italiano:

ARANCIONE, BIANCO, BLU, CELESTE, CIANO, GIALLO, GRIGIO, GRIGIO_CHIARO, GRIGIO_SCURO, INDACO, MAGENTA, MARRONE, MARRONE_CHIARO, MARRONE_SCURO, NERO, ROSA, ROSSO, VERDE, VERDE_CHIARO, VERDE_SCURO, VIOLA

Nome Tartaruga predefinita:

TARTA

Nomi tasti freccia:

FRECCIA_DX, FRECCIA_GIU, FRECCIA_SU, FRECCIA_SX

Comandi

[arco](#)(double x, double y, double r, double angolo1, double angolo2)
Disegna con il colore corrente un arco di circonferenza avente raggio r dall'angolo1 all'angolo2 (in gradi).

[cerchio](#)(double x, double y, double r)
Disegna un cerchio dato il centro e il raggio.

[cerchioPieno](#)(double x, double y, double r, Color c)
Disegna un cerchio pieno con un dato colore, dato il centro e il raggio.

[disegnaImmagine](#)(double x, double y, String s, double gradi)
Disegna un'immagine (.gif, .jpg o .png) centrata su (x, y), ruotata di un dato angolo

[disegnaImmagineRidimensionata](#)(double x, double y, String percorso, double gradi, double largh, double alt)
Disegna un'immagine (.gif, .jpg, o .png) centrata su (x, y), ruotata di un dato angolo e ridimensionata su un'area di date dimensioni.

[ellisse](#)(double x, double y, double semiAsseMaggiore, double semiAsseMinore)
Disegna un'ellisse con il colore corrente, dato il centro e le misure dei due semiassi.

[ellissePiena](#)(double x, double y, double semiAsseMaggiore, double semiAsseMinore, Color c)
Disegna un'ellisse riempita con il colore c, dato il centro e le misure dei due semiassi

[linea](#)(double x0, double y0, double x1, double y1)

Disegna una linea con il colore corrente da (x0,y0) a (x1,y1).

[`pulisci\(\)`](#)

Pulisce la finestra con il colore BIANCO.

[`pulisci\(Color c\)`](#)

Pulisce con il colore specificato.

[`punto\(double x, double y\)`](#)

Disegna un punto in coordinate (x, y).

[`quadrato\(double x, double y, double lato\)`](#)

Disegna un quadrato centrato su (x, y).

[`quadratoPieno\(double x, double y, double lato, Color c\)`](#)

Disegna un quadrato riempito con il colore c centrato su (x, y).

[`rettangolo\(double x, double y, double larghezza, double altezza\)`](#)

Disegna con il colore corrente un rettangolo centrato su (x, y)

[`rettangoloPieno\(double x, double y, double larghezza, double altezza, Color c\)`](#)

Riempie un'area rettangolare con il colore c, centrata su (x, y).

[`salvaDisegno\(String nomefile\)`](#)

Salva l'immagine corrente su un file con un dato nome.

[`setColore\(\)`](#)

Imposta il pennello al colore di default.

[`setColore\(Color c\)`](#)

Imposta il colore del pennello al colore specificato.

[`setFinestra\(\)`](#)

Imposta la finestra di default.

[`setFinestra\(int larghezza, int altezza, String titolo\)`](#)

Imposta la finestra alle dimensioni di larghezza x altezza pixels e con il titolo specificato.

[`setFont\(\)`](#)

Imposta il font di default (Times, 16 punti).

[`setFont\(Font f\)`](#)

Imposta un dato font per la scrittura di testi in grafica.

[`setGroschezza\(\)`](#)

imposta la grossezza del pennello al valore di default (0.002).

[`setGroschezza\(double r\)`](#)

Imposta la grossezza (raggio) della penna grafica

[`setNonXor\(\)`](#)

Imposta la modalità non XOR di scrittura grafica: ripassando 2 volte una linea viene NON VIENE ripristinato quello che c'era prima.

[`setPixel\(int x, int y\)`](#)

Accende un pixel con il colore corrente in coordinate schermo.

[`setXor\(\)`](#)

Imposta la modalità XOR di scrittura grafica: ripassando 2 volte una linea viene ripristinato quello che c'era prima.

[`testo\(double x, double y, String s\)`](#)

Scrivi la stringa S centrata su (x,y) usando il font corrente.

[`testo\(double x, double y, String s, double gradi, Color c\)`](#)

Scrivi la stringa s sulla finestra grafica centrata su (x,y) usando il font corrente, colore c e ruotata di alfa gradi.

[`testoDestra\(double x, double y, String s\)`](#)

Scrivi la stringa s sulla finestra grafica a destra di (x,y) usando il font corrente.

[`testoSinistra\(double x, double y, String s\)`](#)

Scrivi la stringa s sulla finestra grafica a sinistra di (x,y) usando il font corrente.

Operazioni della classe Varie

Osservatori grafica, tastiera e mouse

int	<u>altezza()</u> Altezza in pixel della finestra.
Color	<u>coloreACaso()</u> Restituisce un colore a caso tra un insieme fisso di colori classici.
Color	<u>coloreRGB(int r, int g, int b)</u> Restituisce un colore date le sue componenti r,g,b .
Color	<u>getColore()</u> Colore corrente del pennello.
Font	<u>getFont()</u> Restituisce il font attualmente utilizzato per scrivere testi.
double	<u>getGroschezza()</u> Restituisce la groschezza attuale del pennello.
int	<u>getMouseRGBcode()</u> Restituisce il codice intero del colore del punto sotto il mouse in coordinate schermo
int	<u>getPixelRGBcode(int x, int y)</u> Restituisce il codice intero del colore di un dato pixel (x,y) in coordinate schermo.
int	<u>getRGBcode(Color c)</u> Restituisce il codice intero di un dato colore basandosi sulle componenti r,g,b.
boolean	<u>isXorMode()</u> Restituisce la modalità di disegno.
int	<u>larghezza()</u> Larghezza in pixel della finestra
boolean	<u>mousePremuto()</u> Il mouse risulta premuto ?
double	<u>mouseX()</u> Coordinata X del mouse
double	<u>mouseY()</u> Coordinata Y del mouse
boolean	<u>premutoTasto(int keycode)</u> Restituisce true se il tasto specificato con codice keycode risulta premuto e false altrimenti.

Letture e scrittura di dati

double	<u>askDouble</u> (String invito) restituisce un double letto da tastiera dopo aver scritto un messaggio di invito sulla tastiera.
int	<u>askInt</u> (String invito) restituisce un intero letto da tastiera dopo aver scritto un messaggio di invito sulla tastiera.
String	<u>askString</u> (String invito) Restituisce una stringa letta da tastiera dopo aver scritto a video un messaggio di invito.
	<u>write</u> (Object... args) Scrivere un elenco di valori di espressioni senza andare a capo alla fine.
	<u>writeln</u> (Object... args) Scrivere un elenco di valori di espressioni andando a capo alla fine.

Operazioni su stringhe e formattazione

boolean	<u>convOK()</u> L'ultima chiamata a <code>intConv</code> o <code>doubleConv</code> aveva un dato valido ?
double	<u>doubleConv</u> (String s) Effettua la conversione da stringa a double.

<code>String</code>	<code>doubleFormat</code> (double x, int larghezza, int precisione) Converte in stringa un numero double con un dato numero di cifre decimali nella parte frazionaria
<code>int</code>	<code>intConv</code> (String s) Effettua la conversione da stringa ad intero.
<code>String</code>	<code>intFormat</code> (int x, int larghezza) Converte in stringa un numero intero
<code>boolean</code>	<code>isVuota</code> (String s) la stringa è vuota ?
<code>String</code>	<code>leftFormat</code> (String s, int larghezza) Formatta a sinistra un stringa introducendo spazi a destra se necessario
<code>int</code>	<code>lunghezza</code> (String s) lunghezza di una stringa
<code>char</code>	<code>primoCar</code> (String s) Primo carattere di una stringa.
<code>String</code>	<code>rightFormat</code> (String s, int larghezza) Formatta a destra un stringa introducendo spazi a sinistra se necessario
<code>boolean</code>	<code>rispettaRegex</code> (String s, String regex) verifica se una stringa rispetta la grammatica descritta da una data espressione regolare
<code>char</code>	<code>selCar</code> (String s, int i) seleziona un carattere di una stringa specificando la sua posizione
<code>String</code>	<code>selString</code> (String s, int i, int n) Seleziona una sottostringa di una stringa specificando la posizione del primo carattere e lunghezza (massima) della sottostringa.
<code>String</code>	<code>senzaPrimo</code> (String s) Stringa senza il primo carattere
<code>String</code>	<code>senzaUltimo</code> (String s) Stringa senza il primo carattere.
<code>char</code>	<code>ultimoCar</code> (String s) Ultimo carattere di una stringa.

Numeri casuali e ritardi

<code>int</code>	<code>random</code> (int a, int b) Genera un numero casuale intero compreso in un dato intervallo estremi inclusi.
	<code>ritardo</code> (int millis) Crea un ritardo di una data durata.