INFORMATIK

Klasse: 1.Klasse
Erstellt von: GRUBER ELIAS
Erstellt am: 7. November 2016

KURZFASSUNG

Einleitung

Hier werden die wesentlichen Punkte des Informatik-Unterrichts der 1.Klasse erwähnt. Das Skript sollte als Lernunterlage für die Lehrabschlussprüfung dienen und das erlernen der einzelnen Themen erleichtern. Weiteres sind alle Angaben ohne Gewähr und die einzelnen Themen sind auf das wesentliche herunter gebrochen. Die Weitergabe an Personen, die nicht von GRUBER ELIAS genehmigt wurde, ist nicht gestattet.

INHALTSVERZEICHNIS

1. Begriffe und Teilgebiete der Informatik

Die INFORMATIK ist die Wissenschaft von der systematischen und automatisierten Verarbeitung von Informationen.

Teilgebiete der Informatik

Theoretische Informatik: Befasst sich mit der Algorithmentheorie

Technische Informatik: Befasst sich mit der Schaltpreisentwicklung und den Rechnernetzen

Praktische Informatik: Befasst sich mit dem Compilerbau, Betriebssystemen und Datenbanken

Angewandte Informatik: Befasst sich mit den Anwendungsprogrammen.

Wesentlich ist, das die ersten 3 Teilgebiete den Kern bilden und die angewandte Informatik diese 3 Kernbereiche nutzt.

2. Codes - Grundgerüst

2.1 ASCII Code-Standard

Der ASCII-Code Standard hat 7-Bit (Standard) —> 128 Möglichkeiten

Der 8-Bit wird als Kontrollbit benutzt. Der 8-Bit wird auch Paritätsbit genannt.

Der ASCII-Standard ist der Unterbau von anderen Codetabellen.

Erklärung über die wichtigsten Sprungmarken im ASCII-Code-Standard		
hex	**dec**	**Erklärung**
0-20	0-32	Steuerzeichen
30-39	48-57	Zahlen 0-9
41-5A	65-90	A-Z
61-7A	97-122	a-z

2.2 ASCII Code—Erweitert

Der ASCII Code erweitert besitzt 8-Bit. Das ergibt 256 Möglichkeiten. Der wichtigste Unterschied bezüglich des ASCII Code-Standard ist:

- Umlaute z.B ö,Ä..

2.3 UNI-Code

Der UNI-Code besitzt 2 Byte. Das ergibt 16 Bit und das sind 65.535 Möglichkeiten.
Er beginnt mit dem ASCII Code Standard - 1-127
Nachfolgend geht er mit dem ISO-Latin 1 weiter - 128-255

2.4 UTF-8

Er kann zwischen 1 und 4 Byte besitzen. Der Anfang besteht aber wieder aus dem ASCII-Standard. Der UTF-8 Code ist dynamisch, weil die Byteanzahl von 1-4 Byte variieren kann. Dies ist der Standard-Code im Internet.

3. Geschichte der Datenverarbeitung

Historische Entwicklung:

1) Erste Rechenhilfen
   - Abakus ≈ 1.100 v. Chr.
   - Dezimal-System ≈ 825 n. Chr.

2) Mechanische Rechenmaschinen
   - 4 Grundrechenarten ≈ 1623

3) Programmgesteuerte Automaten
   - Webstühle ≈ 1728
   - Babbage 1840 (Speicher, Rechenwerk,Leitwerk)
   - Hollerith 1886 (USA-Volkszählung mit Lochkarten)

4) Computer
   - Zuse (1935 - Z1, 1941 - Z3 ≈ 2000 Relais)
   - Aiken (1943 Mock 1)
   - ENIAC / EDVAC ≈ 1950

Computergenerationen (1-4)

1) ≈ 1950 Elektronenröhren
   - interner Maschinencode
   - magnetischer Trommelspeicher
   - hoher Energie und Platzbedarf

2) ≈ 1955 Transistoren und Dioden
   - Magnetbänder
   - Stapelbetrieb / Assembler + höhere Programmiersprachen
   - Feritkernspeicher

3) ≈ 1965 IC - Wechselplatten
   - Direktzugriff
   - Multiprogrammierung / Timesharing
   - strukturierte Programmierung

4) ≈ MP - Mikroprozessor
   - komplexe Programmier - Umgebungen
   - weitere Datenspeicher

4. Klassifizierung von Signalen

Einteilung in Analog & Digital-Signale

Analog (Sinuskurve)

zu jedem Zeitpunkt
beliebiger Wert!

Digital (Stufenprinzip)

fixe Werte

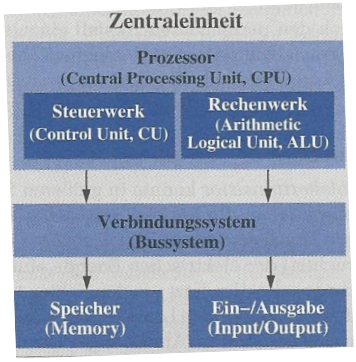
- Musik ist analog
- Für z.B. einen MP3 Player muss diese Digitalisiert werden. Dies funktioniert mit dem AD-Wandler und rückgewandelt kann es mit dem DA-Wandler werden.
- Wellen werden gespeichert als 0/1, das heißt digital
- AD-Wandler (ADC)
- DA-Wandler (DAC)
- Breite der Kurve gibt die Tonhöhe an und die Höhe gibt die Lautstärke an.

5. John von Neumann

Geboren in Budapest - 1903
Gestorben in Washington D.C. -1957

Kernaussagen John von Neumann-Prinzip

- Rechner besteht aus 5 Komponenten
  (Rechenwerk, Steuerwerk, Speicher, Ein-Ausgabe-System, Bus)
- Programm und Daten im selben Speicher
- Bus-System für Datenaustausch
- Ein-Ausgabe-Einheit für Kommunikation-EVA-Prinzip
  (Eingabe-Verarbeitung-Ausgabe)



Eingabe	***Verarbeitung***	***Ausgabe***
Tastatur	Prozessor	Monitor
Maus	RAM	Drucker

6. Zahlensystem Grundlagen

1. römisches Zahlensystem

- ❖ Zeichen sind mit Werten belegt

I. = 1, V=5, X=10, L=50, C=100, D=500, M=1000

- ❖ Einschränkungen für Häufigkeit und Anordnung

2. Dezimalsystem

- ❖ Basis ist 10
- ❖ Werte sind 0,1,2,3,4,5,6,7,8,9

3. Duales System

- ❖ Basis ist 2
- ❖ Werte sind 0 oder 1

4. Hexadeximalsystem

- ❖ Basis ist 16
- ❖ Werte 0-9, A,B,C,D,E,F

5. **Oktalsystem**

- ❖ Basis ist 8
- ❖ Werte sind von 0-7
- ❖ Übungsbeispiele Seite 52

7. Zahlendarstellung

1. BCD-Code (von 0-9 wie Dualsystem + Pseudododedraden)
2. AIKEN (entspricht BCD, jedoch Stelle 4= Wert2
3. Gray-Code (robust, vermeidet Lesefehler)

1. 2 aus 5 (2 Stellen vor 5-Bit gesetzt)
2. 1 aus 10 (1 Stellen vor 10-Bit gesetzt)

8. Diverse Codes

In folgenden Beispielen werden Ihnen verschiedene Codetabellen dargestellt

BCD-Code (grau = Pseudotetrade) ***Gray Code*** ***Aiken Code***

dezimal	hexadezimal	binär
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

4-Bit-Gray-Code:

```
0000
0001
0011
0010
0110
0111
0101
0100
1100
1101
1111
1110
1010
1011
1001
1000
```

Beispiel für Aiken-Code		
Dezimal-ziffer	**Aiken-codiert**	**BCD-codiert**
0	**0 0 0 0**	0 0 0 0
1	**0 0 0 1**	0 0 0 1
2	**0 0 1 0**	0 0 1 0
3	**0 0 1 1**	0 0 1 1
4	**0 1 0 0**	0 1 0 0
5	**1 0 1 1**	0 1 0 1
6	**1 1 0 0**	0 1 1 0
7	**1 1 0 1**	0 1 1 1
8	**1 1 1 0**	1 0 0 0
9	**1 1 1 1**	1 0 0 1

weitere Codes sind:

2 aus 5 Code

1 aus 10 Code

Hamming-Code (fehlerkorrigierend)

Detaillierte Informationen zu den einzelnen Codes sind in der Mappe der 1. Klasse Berufsschule vorhanden. In dieser werden die wichtigsten Eckpunkte zu den Codes erklärt. Diverse Codes

9. Codes im Alltag

QR Code

- Bedeutung: Quick Response
- 1994 in Japan erfunden (Automobilbereich)
- quadratische Matrix mit 3 Positionsmarken
- Binäre Codierung durch Schwarz/weiße Punkte
- auch wenn 30% zerstört, noch lesbar (fehlerkorrektur-Levels)
- speichert: 2956 Bytes
- das sind 7000 Dezimalziffern oder 4296 alphabetische Zeichen

Vorteile:

- leicht erstellbar (Online-Generator)
- einfach zu drucken
- auch farbig wenn Kontrast stimmt
- unterschiedlichste Größen
- leicht lesbar (Smartphone Scanner)

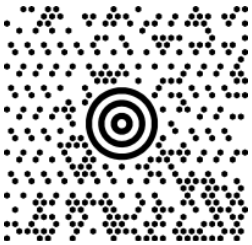
Nachteile:

- Gefahr der schädlichen Lenkangaben bzw. Programmcodes welche nach dem scannen sofort ausgeführt werden.
- primär Scanner Software, nicht QR-Code

Data-Matrix

- erkennbar mit dem unten und linken Balken
- 1556 Byte
- 2300 alphanumerische Zeichen
- 3100 Ziffern

Maxi-Code



- 138 Ziffern
- 93 ASCII
- 144 Symbolzeichen

Aztec-Code

- ❖ 1914 Bytes
- ❖ 3067 Buchstaben
- ❖ 3832 Zahlen

10. Strichcodes

EAN-Code

- Entwicklung 1949
- Einsatz 1973

a.) **13 Stellen (EAN-13)**
- 7 Bit pro Ziffer
- 3 Zeichensätze
- Randzeichen (Anfang/Ende)
- Trennzeichen (Mitte)

Bsp: 978386 894111 1

↓ ↓ ↓

6 6 1

Es sind immer 2x 6er Blöcke und die letzte Stelle ist immer die Prüfziffer. Diese wird aus den zuvor stehenden Zahlen berechnet.
Der erste Block wird immer mit einem A oder B gesetzt und der zweite 6er Block wird mit C gefüllt. Die letzte Stelle ist die Prüfziffer.

Berechnung Prüfziffer

9x1+7x13+8x1+3x13 …. es wird immer die Zahl mit 1 und danach mit 13 multipliziert.
Die Quersumme wird danach berechnet = 389 = 390 = 1

Das Ergebnis der Multiplikation wird dann auf die nächst höhere 10er Stelle aufgerundet. Die Summer zur aufgerundeten 10er Stelle ist die Prüfziffer.

b.) EAN-8
Ist das gleiche Prinzip wie der 13-stellige EAN Code, nur dieser wird auch kleineren Produkten abgebildet, weil der 13-stellige auf kleinen Produkten keinen Platz hätte.

- 8-stellig (7+1)

Struktur

- ***Basisnummern***
- ***Artikelnummer***
- ***Prüfziffer***

PLZ-Zielcode

- orange/flueszierende 5mm hohe Striche
- Feld ca. 15x150 mm
- 5 Bit / 4 Bit- Codierung und Trennstriche
- Briefzentrum - Verteilung und Weiterleitung (circa 1-20 pro Sekunde)

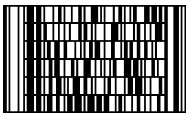
Code 39 (Pharmabereich und Automobil)

- alphanumerischer Code (kann Buchstaben und Zahlen darstellen)
- Strichcode und Zeichen jeweils gedruckt
- Nachfolger ist der Cod 128

2-D Strichcodes

Formen

gestapelt



- Zeilenform

- Code 49
- PDF 417
- Codablock

Matrix

- Matrix / mit oder ohne Positionsmarken

- QR-Code
- Data Matrix
- Maxi-Code
- Aztec-Code

11. Audio/Videoformate

Audioformate

Komprimiert —> verlustfrei: **mp4, wma**
Komprimiert —> verlustbehaftet: **mp3, wma**
Unkomprimiert —> **aiff,pcm,wav**

Videoformate

Beispiel Vollbildberechnung:
1024 x 1024 Pixel/Bild
24 Bit/Pixel
30 Bilder/s
= 1024x1024x24x30 = 755 mBits/s

Grafikformate

Pixelgrafik: Pixel sind in einem Raster hinterlegt. Verwendung bei Fotos. Dateitypen: bmp,gif,jpg,png
Vektografik: Bildbeschreibung. Ohne Qualitätsverlust. Dateitypen: ai,dxf,svg,cdr

Farbräume

RGB: Rot,Grün,Blau - Verwendung: Monitor,Projektoren,Displays
CMYK: Cyan,Magenta,Yellow,K(black) - Verwendung: Drucktechnik 4-Farbendruck

12. Algorithmus-Definition

Definition:

Vorschrift zur schrittweisen Lösung eines Problems bzw. Vorganges.

Eigenschaften:

- deterministisch
  = Eindeutigkeit
- endlich
  = Lösung nach bestimmter Anzahl von Schritten
- effizient
  = geringer Aufwand
- effektiv
  = beste Lösung

Algorithmus-Darstellung (Notation)

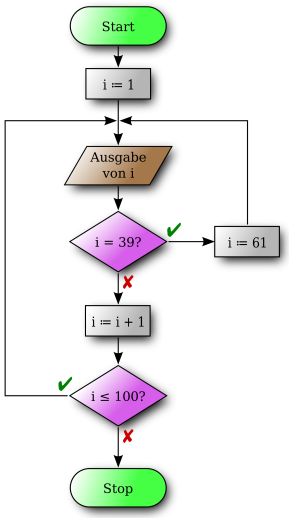
text

- Pseudocode
- Programmiersprache

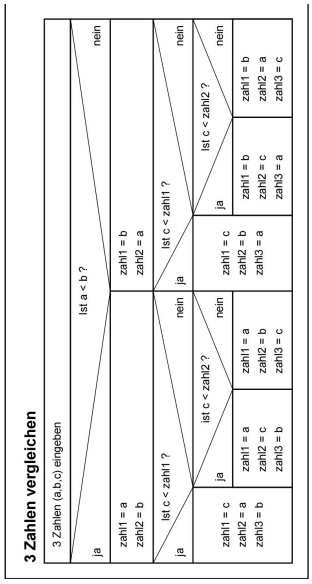
grafisch

- Flußdiagramm
- Struktogramm
- Syntaxdiagramm

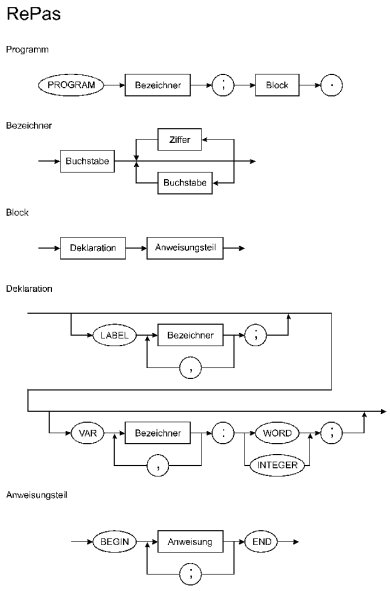
Flußdiagram



Struktogramm



Syntaxdiagramm



13. Grundstruktur Algorithmus

- **Anweisung**	=	Lineare Folge von Anweisungen Einzelschritte, Zuweisungen
- **Auswahl**	=	Selektion / Verzweigung ein-,zweiseitig, mehrfach if, else
- **Wiederholung**	=	Iteration / Schleife Kopf-,fuss-, Zahlenschleife
- **Unterprogramm**	=	Modul - Funktion (mit Return) - Prozedur (ohne Return)

14. Programm

➡ Satz von Befehlen zur Ausführung einer bestimmten Aufgabe.

Programm —> Maschinensprache

a.) Compiler

gesamtes Programm auf einmal übersetzt im
1. Analyse: (Zwischendarstellung)
2. Synthese: Linker für Module + Library

b.) Interpreter

Befehl unmittelbar vor Ausführung geprüft + übersetzt

15. Software-Lebenszyklus

1.) Problemanalyse
- Pflichtenheft (ist ein muss)
- Kunde macht Abnahme ganz zum Schluss mit dem Pflichtenheft
- Schriftlicher Punkteplan muss erstellt werden
- GO oder NOTGO für das Projekt

2.) Systementwurf
- Systemspezifikation

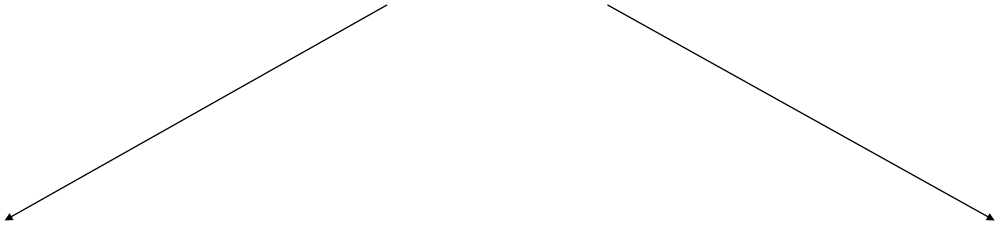
3.) Programmentwurf
- Programmspezifikation

4.) Implementierung + Test
- Programm

5.) Betrieb + Wartung
- Service Level Agreement (SLA) / Wartungsvertrag

16. Programmiersprachen

Programmiersprachen



Generationen

1. Maschinensprache
- Anweisungen = Daten
- binär —> 0/1
- Hardware abhängig
- schwer lesbar

2. Assemblersprache
- Anweisungen = Wortkürzel
- ADD 3,4 = Addiere den Inhalt von Register 3 und 4
- MOV ax 09h
- zeitkritische Anwendungen

3. Problemorientierte Sprachen
- unabhängig von PC-System (Compiler)
- höhere Programmiersprachen
- Summe = 3+4

4. Deklarative Sprache
- Funktion wird beschrieben (einzelne Anweisung Vielzahl von internen Schritten)
- Select * form t_Kunden;

5. Künstliche Intelligenz
- Wiesenverarbeitung / Spracherkennung

Sprachtypen

1. prozendurale Programmier-Sprachen
- Programmabläufe werden beschrieben
- z.b. als Folge von Einzelschritten
- 1-3 Generation
- UP-Technik = mehrmalige Verwendung
- Strukturierte Programmierung
- Daten und Funktionen sind getrennt

>>> C, BASIC <<<

2. objektorientierte Programmier-Sprachen
- Zusammenfügen von Daten & Funktionen
- in Klassen mit Attribute + Methoden = Objekte gebildet
- Prinzip:
- Kapselung
- Vererbung
- Polymorphie

>>> C++,Java, C# <<<

3. logische Programmier-Sprachen
- Bedingung für konkrete Lösung

>>> SQL <<<

4. funktionelle Programmier-Sprachen
- Einsatz von Funktionen

>>>> F# <<<

Verschiedene Programmiersprachen:
Genauere Details zu den Programmiersprachen finden Sie in der Mappe der 1.Klasse!

- C
- C#
- C++
- Java
- JavaScript
- PHP
- Phyton

17. Software Entwicklungsmodelle

Definition:

Software-Entwicklungsmodelle dienen zur strukturierten Vorgangsweise. Dabei wird das Risiko zu Fehlern minimiert und der Projektstand/Status kann klar definiert werden.
Weiteres dienen Sie auch noch zur Steuerung einer Software-Entwicklung

Da es eine Vielzahl von Software-Entwicklungsmodellen gibt und dies den Rahmen dieser Kurzfassung sprengen wurde, werden hier nur die Namen der Software-Entwicklungsmodelle angeführt. Genauere Details zu den einzelnen Entwicklungsmodellen entnehmen Sie bitte aus der Mappe der 1.Klasse Berufsschule.

Diverse Entwicklungsmodelle:

Evolutionäres-Modell
Wasserfall-Modell
XP-Modell
V-Modell
Spiral-Modell

18. SCRUM

Scrum-Rollen

1.) Product-Owner = eine Person

- ➡ verantwortlich für Eigenschaften + wirtschaftlicher Erfolg
- ➡ vermittelt und entscheidet was wann umgesetzt wird

2.) Entwicklerteam = 3-9 Personen

- ➡ liefert Produktfunktionalität
- ➡ besteht aus unterschiedlichen Spezialisten die teilweise alles können
- ➡ Wandelt Sprinteinträge in (24h)-Tasks um

3.) Scrum-Master = Prozess-Coach

- ➡ verantwortlich dass Scrum gelingt
- ➡ führt Scrum ein und vermittelt zwischen Product-Owner(1) und Entwicklungsteam(2)

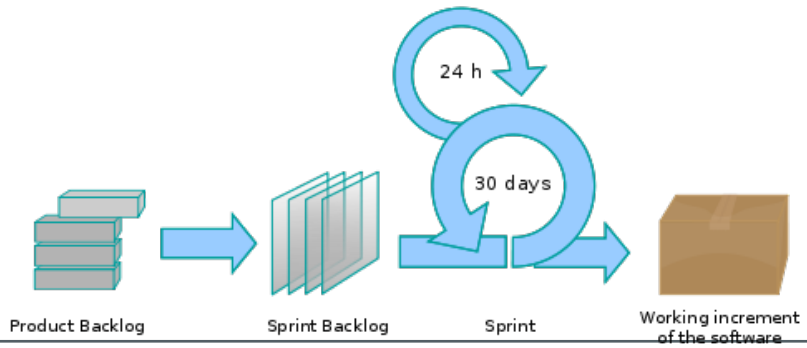
weitere Rollen = Stakeholder

- ➡ Kunden
- ➡ Anwender
- ➡ Management

Aktivitäten:

- ➡ fixe Zeitboxen!
- ➡ Sprint-Planung max 2 Stunden für 1 Woche (Was+Wie)
- ➡ Daily Scrum max 15 Minuten (Überblick + Informationsaustausch) für Lösung eigenes Meeting!
- ➡ Sprint Review max 1 Stunde pro Sprintwoche (am Sprint-Ende, Meeting Ergebnisse präsentiert und ob Ziel erreicht wurde)
- ➡ Feedback Stakeholder
- ➡ Sprint Retrospektive (maximal 45 Minuten pro Woche) bessere Zusammenarbeit im Scrum-Team
- ➡ Product Backlog Refinement (Product-Owner und Entwicklungsteam) organisieren und bereinigen - Product-backlog

Scrum-Skizze



19. Qualitätsmerkmale von Software

Robust: Wir sprechen von einer robusten Software, wen trotz unvorhergesehenen Umständen das System vernünftig reagiert und funktioniert.
Beispiel: Bei der Datumseingabe den Namen eingeben darf nicht funktionierten

Effizienz: Eine Software verhält sich effizient, wenn sie die zur Verfügung stehenden Hardware-Ressourcen ökonomisch nutzt.
Beispiel: Quard Core nutzen.

Benutzerfreundlichkeit: Leicht zu erlernen und zu benutzen. Intuitive Bedienung.

Wartbarkeit: Dokumentation im Quelltext.

Portierbarkeit: Auf anderen Plattformen laufend.

Kompatibilität: Schnittstellen sind auch für andere Systeme funktionsfähig

Korrektheit: Exakte Erfüllung der Anforderungen.

Zuverlässigkeit: lange Laufzeit

20. Software-Dokumentation

SYSTEM-Dokumentation

- Quellcode
- Flussdiagramme
- Datenstrukturen
- Funktionsbeschreibung
- Lasten/Pflichtenheft

    - Basis für Wartung
    - sehr detailliert
    - teilweise vertraulich

BENUTZER-Dokumentation

- Handbuch / F1-Hilfe
- Installationsanleitung
- Systemvoraussetzungen
- logischer Aufbau
- FAQ

    - klar verständlich
    - aktuelle Version/Stand
    - Minimalversion notwendig