

Einstieg in die Programmierung C#/VBA/Python/Java

1 Themen

1.1 Themen Lektion 1

- Hardware
- Zahlensysteme
- Von der Maschinensprache zu den Hochsprachen
- Übersicht über verschiedene Hochsprachen
- Prozedurale Sprachen: Kontrollstrukturen I

1.2 Themen Lektion 2

- Prozedurale Sprachen: Kontrollstrukturen II
- Objektorientierte Programmierung
- Hybride Sprachen
- Programmierung im Web
- Programmierung mobiler Geräte
- Werkzeuge der SW Entwicklung
- Tutorial

Inhaltsverzeichnis

1 Themen	1
1.1 Themen Lektion 1	1
1.2 Themen Lektion 2	1
1.3 Aufbau von Dezimalzahlen	3
1.4 Umwandlung von Dezimalzahlen in Dualzahlen und umgekehrt	4
1.5 Das Hexadezimalsystem	4
1.6 Verschiedene Busse	5
1.7 Assembler	7
1.8 Assembler - Hochsprache	8
1.9 Künstliche Intelligenz	9
1.10 Scratch	10
1.10.1 Pythagoras	10
1.11 Java Script	10
1.11.1 Hello World	10
1.11.2 Lampe	10
1.11.3 jsfiddle	10
1.11.4 dotnetfiddle	10
1.12 Regular Expressions	11
1.13 Regular Expressions in Visual Studio	11
1.14 Bug	12
1.15 PDP-8	13
1.16 Die Geburt des Mikroprozessors	14
1.17 Die ersten PCs	14
1.18 Geschichte (Bug)	14
1.19 Internas	15

1.3 Aufbau von Dezimalzahlen

Zahl	Bedeutung
5763	
	$3 * 1$
	$6 * 10$
	$7 * 100$
	$5 * 1000$
Summe	5763

Man könnte aber auch schreiben:

Zahl	Bedeutung
5763	
	$3 * 10^0$
	$6 * 10^1$
	$7 * 10^2$
	$5 * 10^3$
Summe	5763

Im Dezimalsystem ist 10 die Basis. Mit jeder neuen Stelle erhöht sich der Exponent um 1.

1.4 Umwandlung von Dezimalzahlen in Dualzahlen und umgekehrt

Umwandlung der Dualzahl 11111100000_2 in die entsprechende Dezimalzahl

Faktor	Wert in Dezimal	Wert in Dezimal	Faktor * Wert
0	2^0	1	0
0	2^1	2	0
0	2^2	4	0
0	2^3	8	0
0	2^4	16	0
1	2^5	32	32
1	2^6	64	64
1	2^7	128	128
1	2^8	256	256
1	2^9	512	512
1	2^{10}	1024	1024
			2016

Umwandlung der Dezimalzahl 2016_{10} in die entsprechende Dualzahl

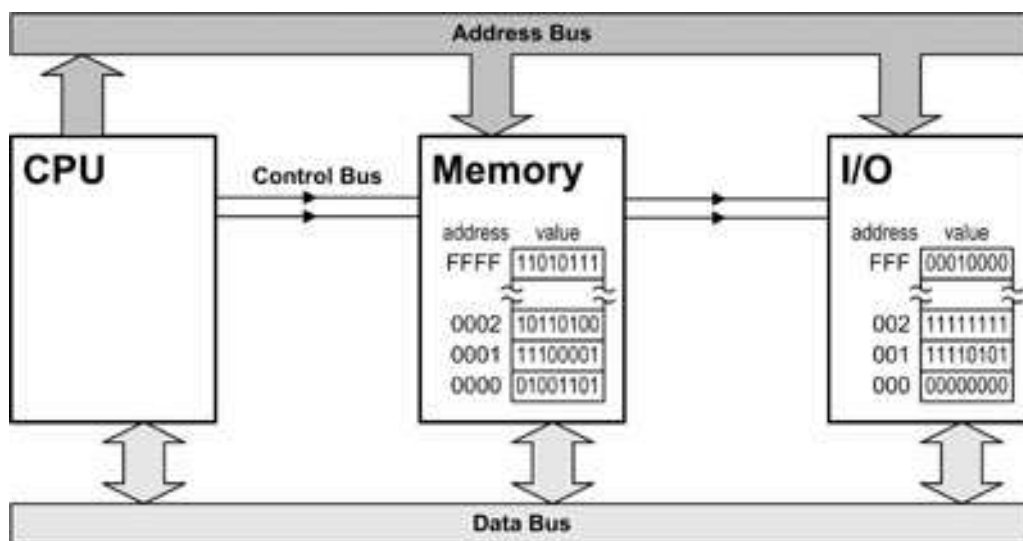
Wert in Dezimal	Operation	Resultat	Rest
2016	$\div 2$	1008	0
1008	$\div 2$	504	0
504	$\div 2$	252	0
252	$\div 2$	126	0
126	$\div 2$	63	0
63	$\div 2$	31	1
31	$\div 2$	15	1
15	$\div 2$	7	1
7	$\div 2$	3	1
3	$\div 2$	1	1
1	$\div 2$	0	1

1.5 Das Hexadezimalsystem

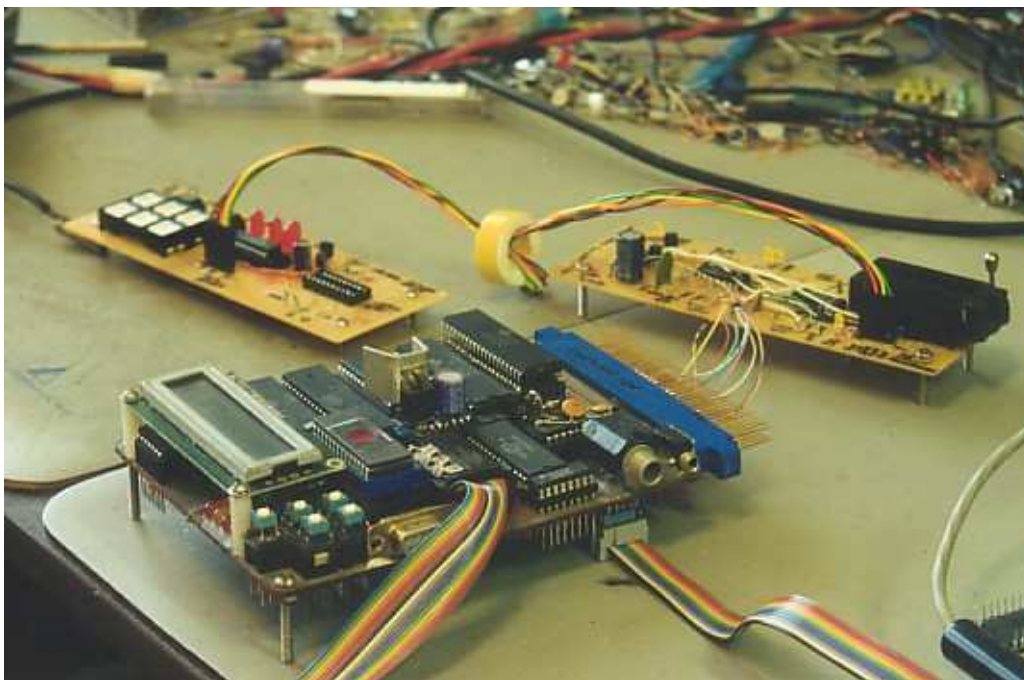


Quelle: Mars ☺

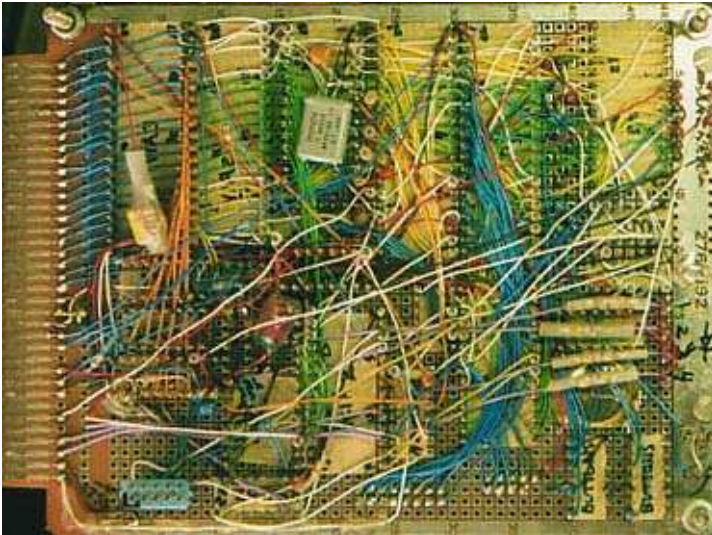
1.6 Verschiedene Busse



Quelle: <http://www.talktoanit.com/A+/aplus-website/lessons-io-principles.html>



Quelle: <http://www.6502.org/users/garth/projects.php>



Quelle: <http://www.6502.org/users/garth/projects.php>

1.7 Assembler

Ausschnitt 1:

```
NEWBASIC    STX IMAIN          ; ($0302)
             STY IMAIN+1        ; ($0303)
             JSR CHRGET          ; GET CHARACTER
             CMP #95             ; IS IT A "<-" ?
             BEQ CHK             ; YES, CONTINUE
             JMP GONE+3          ; NORMAL WORD
CHK          JSR CHRGET          ; GET NEXT CHARACTER
             CMP #$44            ; "D" Detect
             BEQ reumain
             CMP #$42            ; "B" Bank,?
             BEQ getBank
             CMP #$53            ; "S" Stash ($0800-$a000)
             BEQ stashreu
             CMP #$46            ; "F" Fetch ($0800-$a000)
             BEQ fetchreu1
             CMP #$55            ; "U" User,c64,reu,bank,bytes,command
             BEQ userreu
             JMP (IERROR)
```

Ausschnitt 2:

```
stashreu    ldx usebank
             lda banktable,x      ; check banktable
             cmp #85              ; "U" Bank in use
             bne doit             ; no, skip
             ldy #00              ;
t1           lda bankuse,y
             JSR CHROUT
             iny
             cpy #20
```

Quelle: <http://www.cbmhardware.de/show.php?r=1&id=7>

1.8 Assembler - Hochsprache

X-86 Assembler:

Label	Mnemonic	Operand	Comment

	.data		
exCode	DB	0	;A byte variable
myWord	DW	?	;Uninitialized word var.
	.code		
MAIN	PROC		
	mov	ax,@data	;Initialize DS to address
	mov	ds,ax	; of data segment
	jmp	Exit	;Jump to Exit label
	mov	cx,10	;This line skipped!
Exit:	mov	ah,04Ch	;DOS function: Exit prog
	mov	al, exCode	;Return exit code value
	int	21h	;Call DOS. Terminate prog
MAIN	ENDP		;End Program
	END	MAIN	; and specify entry point

Quelle: http://www.shsu.edu/csc_tjm/fall2004/cs272/intro_to_asm.html

C-Sharp:

```

1  using System.Reflection; // reflection namespace
2
3  // get all public static properties of MyClass type
4  PropertyInfo[] propertyInfos;
5  propertyInfos = typeof(MyClass).GetProperties(BindingFlags.Public |
6                                              BindingFlags.Static);
7  // sort properties by name
8  Array.Sort(propertyInfos,
9              delegate(PropertyInfo propertyInfo1, PropertyInfo propertyInfo2)
10                 { return propertyInfo1.Name.CompareTo(propertyInfo2.Name); });
11
12 // write property names
13 foreach (PropertyInfo propertyInfo in propertyInfos)
14 {
15     Console.WriteLine(propertyInfo.Name);
16 }

```

Quelle: <http://www.csharp-examples.net/reflection-property-names/>

1.9 Künstliche Intelligenz

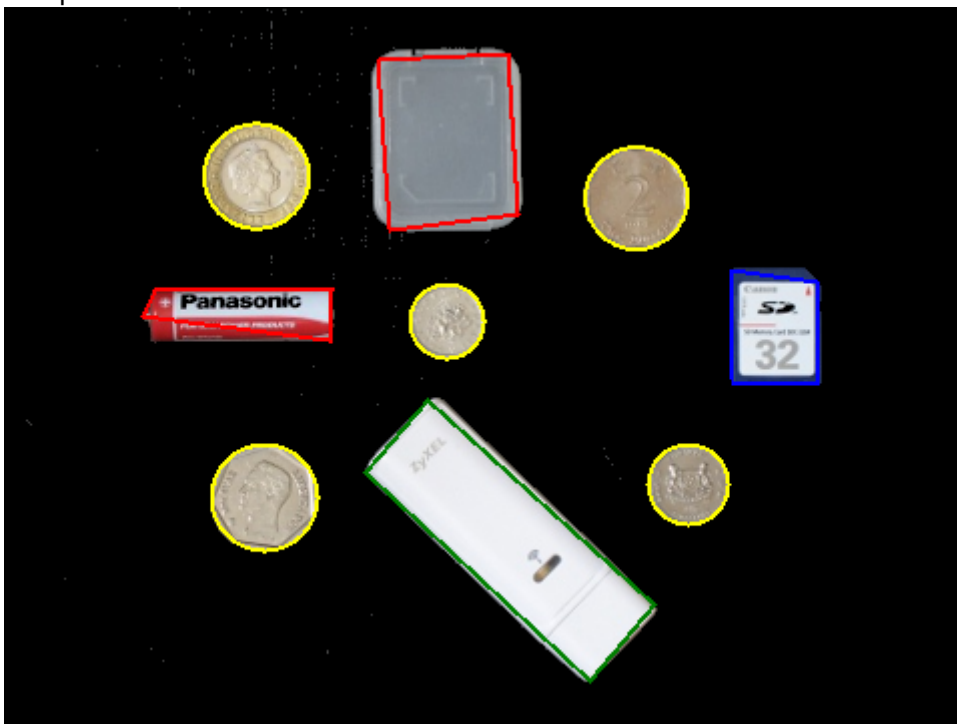
Das AForge.NET Framework

Link: <http://www.aforgenet.com/framework/projects.html>

Es beinhaltet folgende Teilbereiche:

- Image processing
- Computer vision
- Accessing/playing video
- Math
- Robotics
- Artificial Neural Networks
- Genetic Algorithms
- Fuzzy Systems.

Beispiel:

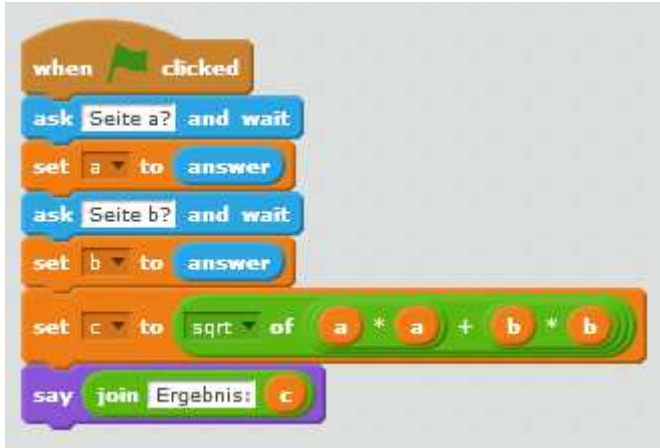


Quelle: http://www.aforgenet.com/articles/shape_checker/coins3.png

1.10 Scratch

Scratch IDE: <https://scratch.mit.edu/>

1.10.1 Pythagoras



1.11 Java Script

1.11.1 Hello World

http://www.w3schools.com/html/tryit.asp?filename=tryhtml_scripts_intro

1.11.2 Lampe

http://www.w3schools.com/js/tryit.asp?filename=tryjs_intro_lightbulb

1.11.3 jsfiddle

<https://jsfiddle.net/>

1.11.4 dotnetfiddle

<https://dotnetfiddle.net/>

1.12 Regular Expressions

<http://regexr.com/>

<https://regexper.com/>

Regular Expression	Findet	Kommentar
/over/g	over	
/./g	alle Zeichen	Siehe Cheatsheet
/[a-z]/g	Kleinbuchstaben	Charakterklasse
/[c-s]/g	Kleinbuchstaben c bis s	Charakterklasse
/[c-s][c-s]/g	c bis s hintereinander	Charakterklasse
/[A-Z]/g	Grossbuchstaben	Charakterklasse
/\./g	Punkt	Escaped Dot
/expressions?/g	0 bis n Wiederholungen	Quantifier
/[a-zA-Z0-9]+/g	1 bis n Wiederholungen	Quantifier
/[a-z]{9,10}/g	9 bis 10 Wiederholungen	Quantifier
/(.*)((\.net \.com))/g	Domains / URLs	Groups
/[a-zA-Z0-9\.]++@[a-zA-Z0-9]+\.[a-zA-Z0-9\.]++/g	Email Adressen	
/[A-Z]/g	ersetzt	Q
/((https?:)(.))/g	ersetzt	\$2
/([A-Z])([a-z])/g	ersetzt	\$1[\$2]

1.13 Regular Expressions in Visual Studio

Die folgenden Beispiele beziehen sich auf das File: NorthwindDataSet.Designer.cs

Beispiel:

attribute2.*demog -> L2817

IP Adresse:

\d+\.\d+\.\d+\.\d+

1.14 Bug

Die Verknüpfung des Begriffs mit Computern geht möglicherweise auf die Computerpionierin Grace Hopper zurück. Sie verbreitete die Geschichte, dass am 9. September 1945 eine Motte in einem Relais des Computers Mark II Aiken Relay Calculator zu einer Fehlfunktion führte.

Quelle: <https://de.wikipedia.org/wiki/Programmfehler>

1.15 PDP-8

Der PDP-8 war ein früher Minirechner aus der Reihe Programmed Data Processor von Digital Equipment Corporation. Er wurde unter Leitung von Edson de Castro entwickelt und 1965 auf den Markt gebracht. Der Preis von 18.000 \$ ermöglichte auch kleineren Unternehmen den Einsatz eines Rechners. Es wurden 50.000 Einheiten produziert, womit die PDP-8 kommerziell sehr erfolgreich war.

Quelle: <https://de.wikipedia.org/wiki/PDP-8>



Quelle: <http://www.blinkenlights.com/pc.shtml>

1.16 Die Geburt des Mikroprozessors

<http://www.klammeraffe.org/~fritsch/uni-sb/fsinfo/Papers/PC/node9.html>

1.17 Die ersten PCs

<http://www.blinkenlights.com/pc.shtml>

1.18 Geschichte (Bug)

<http://www.adp-gmbh.ch/personal/histoire/histoire.html>

1.19 Internas

Öffentliche Site <https://sites.google.com/view/klubschule>

Addendum Herdt

- Grundlagen zu Programmen 8
App (engl. für Application)
Programm: Folge von Befehlen
- Zahlensysteme und Zeichencodes 48 Römisches vs. arabisches Zahlensystem
Nennwerte
Massvorsätze
ASCII beginnt mit 1: 65, 97
- Grundlagen der Softwareentwicklung 156 Unit tests
Moq Framework
- Klassifizierung von Programmiersprachen 9 Fortran: Komplexe Zahlen, auch als OO
Brian Wilson Kernighan und Dennis Ritchie
Wie lauten die 4 Säulen der objektorientierten Programmierung?
 - Kapselung
 - Abstraktion
 - Vererbung
 - Polymorphismus

Visual J++

.NET Core

Hybride Programmiersprachen: Python, Powershell

Funktionen sind pur, d. h. sie bekommen etwas übergeben und liefern etwas zurück, und das ist alles was sie tun. Also keine Seiteneffekte, wie zB etwas zu verändern, dass sich ausserhalb der Funktion befindet.

F#

Web

- Rias
- http html
- Web 1.0 vs. Web 2.0
- Ajax Rias
- Mobile Apps: iOS: Objective C; Android: Java; Windows Phone C#
- Cross Platform Frameworks: Titanium, Phone Gap, Corova, Xamarin

- Werkzeuge der Softwareentwicklung 30 Syntax Highlighting
Intellisense
Bereich kommentieren
Multiline Edit

Regex
- Einführung in die objektorientierte Programmierung (OOP) 112