



University
of Basel

Programmieren I

Woche 2 – Schleifen und Methoden

Marcel Lüthi, Departement Mathematik und Informatik, Universität Basel

Heutiges Programm

Zeit	Inhalt
09:15 – 10:00	Wiederholung Grundkonzepte Gleitkommazahlen
10:00 – 10:30	Nachbesprechung Übungen
10:30 – 10:45	Kaffeepause
10:45 – 12:00	Schleifen (Theorie)
12:00 – 13:15	Mittagspause
13:15 – 14:15	Übungsblock I – Schleifen
14:15 – 15:30	Methoden (Theorie)
15:30 – 15:45	Kaffeepause
15:45 – 17:00	Übungsblock II - Methoden

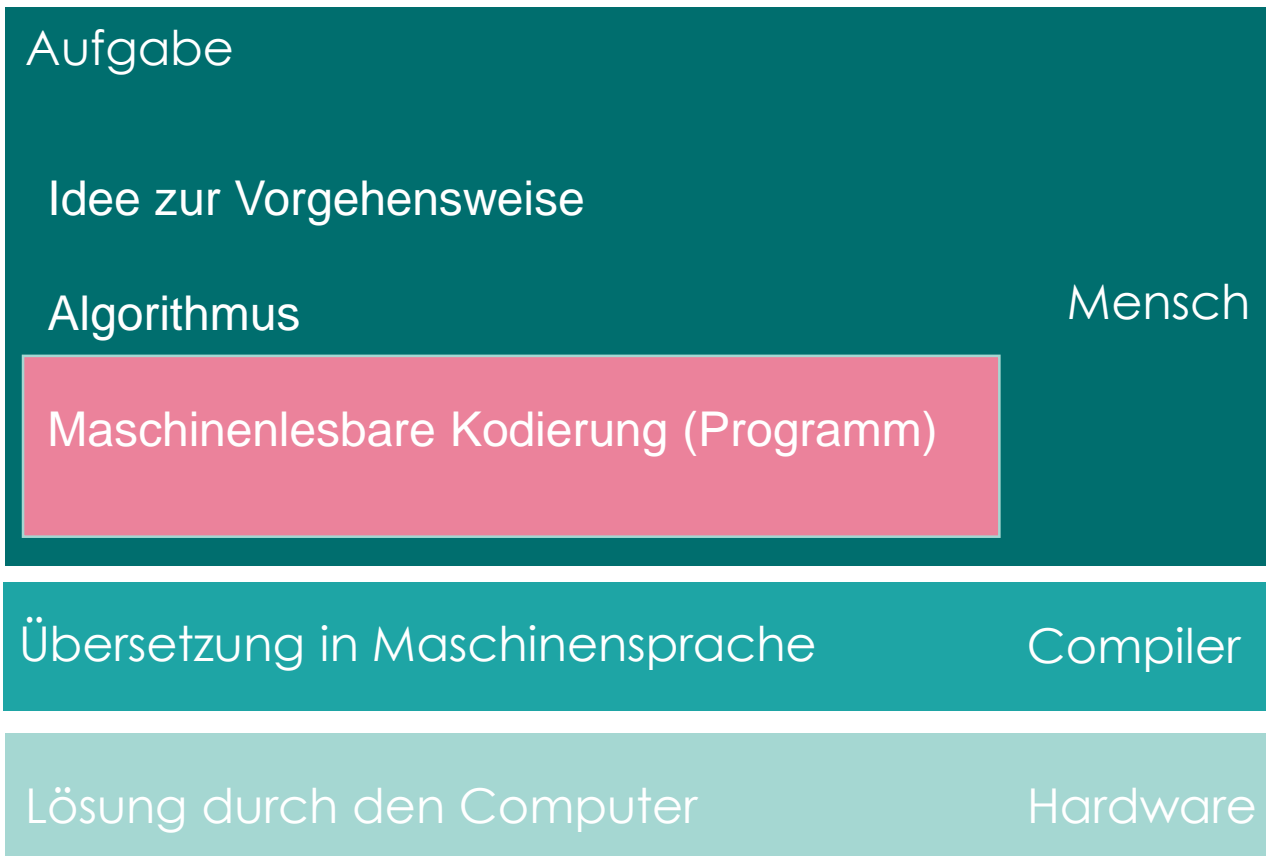


University
of Basel

Wiederholung der Grundkonzepte

Was heisst Programmieren

Exaktes Instruieren eines Computers, eine bestimmte Aufgabe zu lösen.



Programm (GGT.java)

```
class GGT{  
    public static void main (String[] arg) {  
        // ... Berechnung vom GGT  
        System.out.println("GGT ist : " + ggt);  
    }  
}
```

Übersetzen (erzeugt GGT.class)

```
> javac GGT.java
```

Ausführen:

```
> java GGT
```

Typisches Java-Programm

```
/*  
 * Berechnet den BMI  
 */  
public class BMI {  
    public static void main(String[] args) {  
        double heightInM = Integer.parseInt(args[0]) / 100.0;  
        double weightInKg = Integer.parseInt(args[1]); // Gewicht muss ganzzahlig sein  
        double bmi = weightInKg / (heightInM * heightInM);  
  
        System.out.println("Ihr BMI beträgt: " + bmi);  
        if (bmi < 20) {  
            System.out.println("Sie haben einen BMI unter 20");  
        } else if (bmi > 25) {  
            System.out.println("Sie haben einen BMI über 25");  
        } else {  
            System.out.println("Sie haben einen BMI zwischen 20 und 25.");  
        }  
    }  
}
```

Kommentar

Name

Zahl

Schlüsselwort

Kommentar

Zeichenkette

Typisches Java-Programm

```
/*  
 * Berechnet den BMI  
 */  
public class BMI {  
  
    public static void main(String[] args) {  
        double heightInM = Integer.parseInt(args[0]) / 100.0;  
        double weightInKg = Integer.parseInt(args[1]); // Gewicht muss ganzzahlig sein  
  
        double bmi = weightInKg / (heightInM * heightInM);  
  
        System.out.println("Ihr BMI beträgt: " + bmi);  
        if (bmi < 20) {  
            System.out.println("Sie haben einen BMI unter 20");  
        } else if (bmi > 25) {  
            System.out.println("Sie haben einen BMI über 25");  
        } else {  
            System.out.println("Sie haben einen BMI zwischen 20 und 25.");  
        }  
    }  
}
```

Klassenkontext

main-Methode

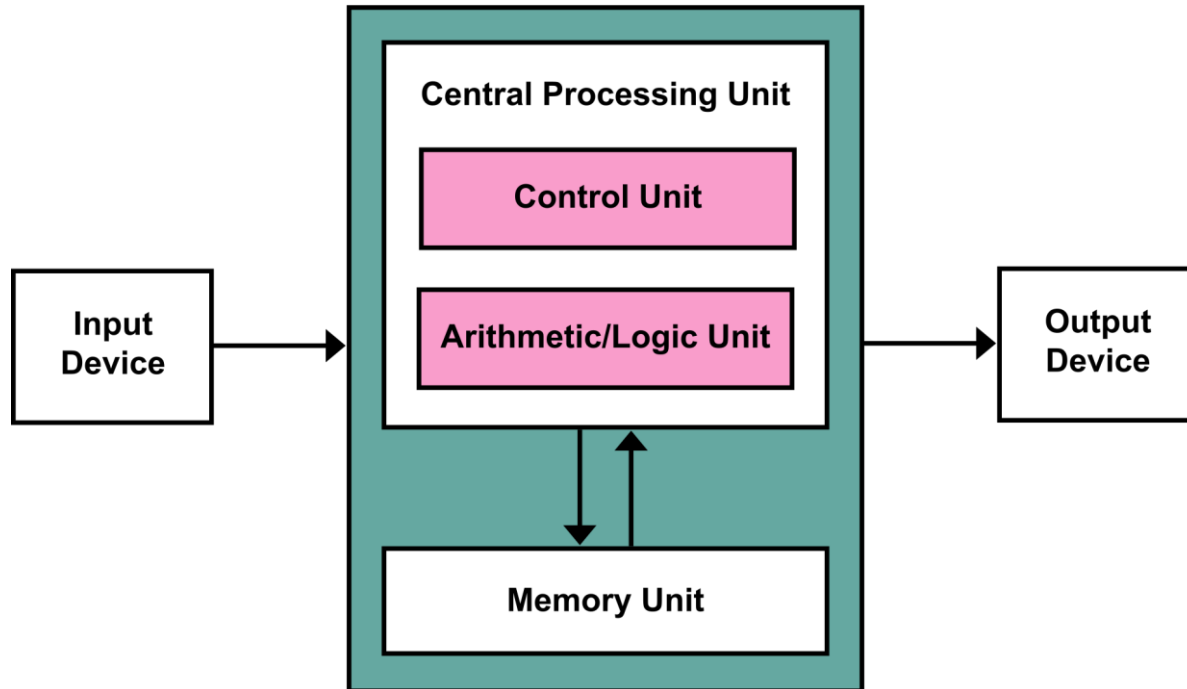
Variablendeklaration

Zuweisung

Arithematische Ausdrücke

Verzweigung

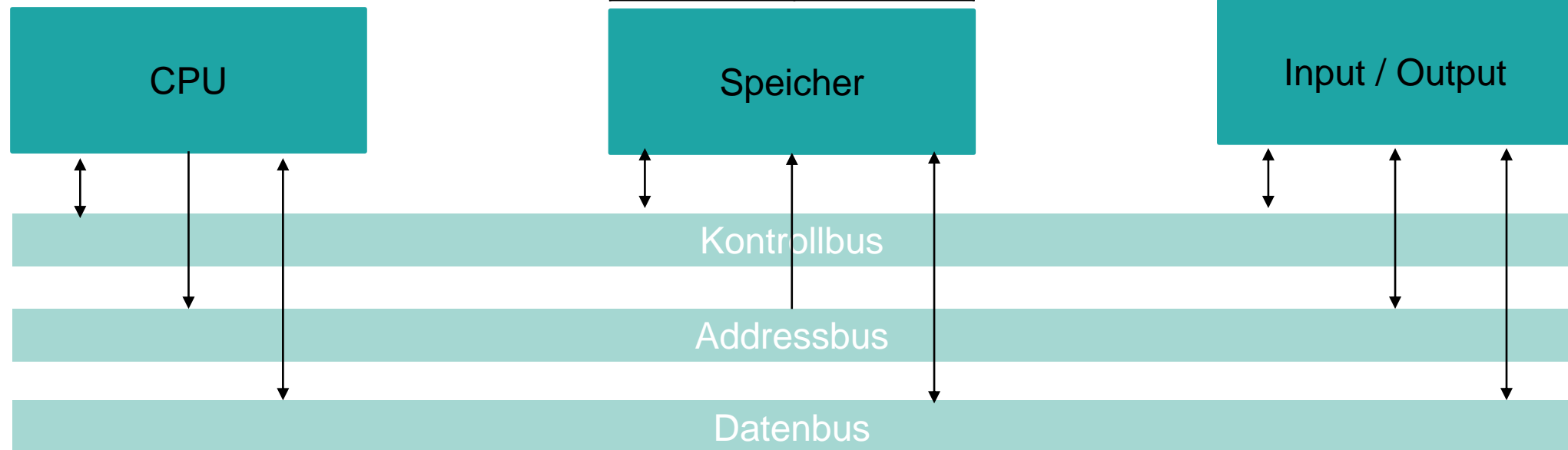
Von-Neumann Architektur



By Kapoht - Own work, CC BY-SA 3.0,
<https://commons.wikimedia.org/w/index.php?curid=25789639a>

Speicher & Adressierung

Adresse	Inhalt
00000001	01011011
00000010	11100001
00000011	10010111
00000100	00011011
00000101	11000111



Variablen

Variablen sind Namen für Speicheradressen

- Einfacher zu merken als Zahl

Variable hat einen Wert

- Inhalt des Speichers an dieser Adresse

Variable hat einen Typ

- Wie gross ist Speicherstelle?
- Wie muss Bitmuster interpretiert werden?

Kleine Zahl $x = 11$

Grosse Zahl $y = 257$

Zeichen $z = 'a'$

Adresse	Inhalt
00000000	00001011
00000001	00000001
00000010	00000001
00000011	10010001
00000100	11000111

Datentypen in Java

byte	8 Bit Zahl	$-2^7 \dots 2^7 - 1$	(-128, , 127)
short	16 Bit-Zahl	$-2^{15} \dots 2^{15} - 1$	(-32768, , 32767)
int	32 Bit-Zahl	$-2^{31} \dots 2^{31} - 1$	(-2 147 483 648, , 2 147 483 647)
long	64 Bit-Zahl	$-2^{63} \dots 2^{63} - 1$	
float	32 Bit IEEE-754-1985 Gleitkommazahl		
double	64 Bit IEEE-754-1985 Gleitkommazahl		
char	16 Bit Unicode		
boolean	Wahrheitswert, <i>false</i> oder <i>true</i>		

Variablen in Java

Deklaration

- macht den Namen und den Typ der Variablen bekannt
- Compiler reserviert Speicherplatz für die Variable

<code>int x;</code>	deklariert eine Variable x vom Typ int (integer)
<code>short a;</code>	deklariert Variable a vom Typ short (short integer)

Initialisierungen

<code>int x = 100;</code>	deklariert int- Variable x; weist ihr den Anfangswert 100 zu
<code>short a = 0;</code>	deklariert short-Variablen mit Anfangswert 0

Gleitkommazahlen

Jupyter
nbviewer

JUPYTER FAQ </> [Menu] [Refresh] [Share] [Download]

gyminf-programmieren / notebooks

Einführung in Jupyter notebooks

In diesem Notebook geben wir eine kleine Einführung in Jupyter Notebooks und zeigen wie man einfache Java Programme schreibt. Ausserdem führen wir Turtlegrafik ein, die wir während dem Kurs immer wieder verwenden werden um Programmierkonzepte zu illustrieren.

Einfache Java Programme in Jupyter Notebooks

Jupyter Notebooks sind im Grunde Dokumente, die neben Text auch Programmcode enthalten. aus einer Sequenz von einzelnen Zeilen bestehen. Die aktuelle Zeile erkennt man jeweils daran, dass diese umrahmt ist. Jede Zeile ist entweder eine Textzeile oder eine Code Zeile. Codezeilen erkennt man dadurch, dass diese mit In [...] gekennzeichnet sind. Codezeilen enthalten ausführbaren Java Code und können durch drücken von **Shift+Enter** ausgeführt werden. Die folgenden zwei Zeilen sind Codezeilen und können ausgeführt werden.

```
In [ ]: 5 + 3;
```

```
In [ ]: System.out.println("Programmieren macht Spass");
```

Mini-Übung:

- Führen Sie die obigen 2 Codezeilen aus.

Wir hier sehen können wir hier im Gegensatz zu richtigen Java Programmen, direkt einzelne Java Ausdrücke eingeben, die dann direkt ausgeführt werden. Wir brauchen kein Klassengertist und müssen nicht erst den Compiler aufrufen. Jeder Ausdruck wird direkt von Java (genauer dem Programm JShell) interpretiert.

Turtle grafik

Wir werden in diesem Kurs häufig Turtle Grafik einsetzen um Konzepte zu illustrieren. Man kann sich vorstellen, dass eine Stiftragende Schildkröte sich auf einer Zeichnungsebene bewegt, und dabei eine Zeichnung erstellt. Die Bewegungen der Schildkröte folgen genau unseren Befehlen.

Damit wir Turtle Grafik nutzen können, müssen wir zuerst eine Programmbibliothek in Jupyter laden:

```
In [ ]: %mavenRepo bintray https://dl.bintray.com/egg/maven
%maven ch.unibas.informatik:jturtle:0.5
import ch.unibas.informatik.jturtle.Turtle;
```

Danach können wir die Schildkröte mit Java-Befehlen steuern:

```
In [ ]: Turtle turtle = new Turtle();
turtle.penDown();
turtle.forward(50);
turtle.turnRight(120);
turtle.forward(50);
turtle.turnRight(120);
turtle.forward(50);
turtle.toImage();
```

Das System ist sehr einfach zu verstehen. Turtle wurde ursprünglich auch dazu entwickelt, Kindern das Programmieren beizubringen. Mithilfe von Turtle Grafik können jedoch auch sehr komplexe und abstrakte Konzepte graphisch illustriert werden. Das folgende Programm ist ein Beispiel für ein komplexes Programm, welches Sie am Ende dieser Vorlesung ohne Probleme verstehen werden.

Notebook: bit.ly/gyminf-programmieren -> Alle Notebooks im Überblick -> Gleitkommazahlen.ipynb