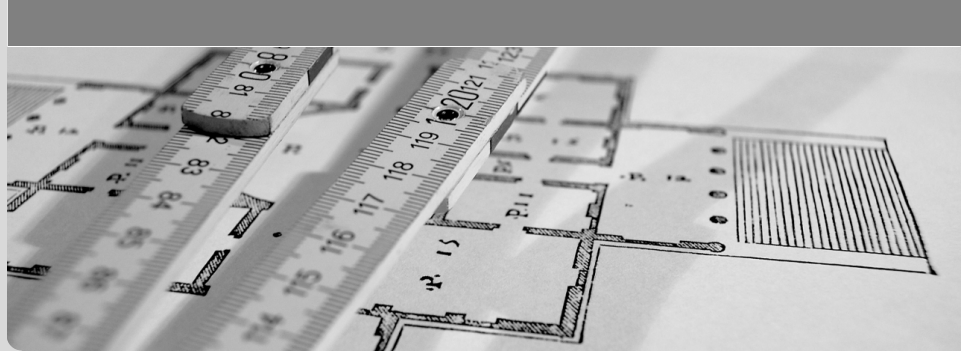


Programmieren

1. Tutorium

Organisatorisches, Java, Editoren, Hello World, Datentypen, Klassen, Objekte

Robin Rüde | 2. November 2014



1 Begrüßung

2 Organisatorisches

- Accounts/Webseiten
- Aufbau des Moduls

3 Java

- Installation
- Editoren
- HelloWorld / Kompilieren
- Datentypen
- Variablen
- Operatoren
- Klassen und Objekte

- Dieses Tutorium:
 - Tutorium zu Programmieren
 - Tut-Nr: 5
 - jede Woche Montag 11:30 - 13:00
- Ich:
 - Robin Rüde
 - 19 Jahre alt
 - Info im 3. Semester
 - Programmiererfahrung: C++/C#/C/Java/Python/Webentwicklung/etc
- Ihr?

2

Organisatorisches

`http://studium.sexy`

- wenn keinen Account (u-Kürzel) schnell zum SCC Service Desk
- Ilias-Kurs „Programmieren“ beitreten
 - <https://ilias.studium.kit.edu>
 - PW: progws1415
 - Verlinkt auf Vorlesungsseite und Praktomat
- Einloggen im Praktomat
 - https://praktomat.cs.kit.edu/2014_WS
 - VPN nötig! (siehe Wiki im Ilias)
 - automatische Plagiatsprüfung (JPlag): sehr, sehr zuverlässig
- Vorlesungsaufzeichnungen auf Youtube
- Einverständniserklärung abgeben (jetzt)

- Vorlesung / Übung + Tutorium zum Üben/Wiederholen
- 5 LP / ECTS
- Übungsblätter Pflicht, Übungsschein nötig für Klausur -> Anmeldung `campus.studium.kit.edu` sobald freigeschaltet (s. Vorlesung)
- Klausur Pflicht: 2 Aufgaben über jeweils 4 Wochen, Abgabe im Praktomaten
- bis zum 2. Semester Prüfung antreten, bis zum 3. Semester bestehen (Teil der Orientierungsprüfung)

- Sechs Übungsblätter mit je 20 Punkten => 120 Punkte insgesamt
- 50% (60 Pkt) zum bestehen
- Bewertung nach
 - Methodik (Modellierung, Dokumentation, Lesbarkeit)
 - Funktionalität (Korrektheit, Stabilität)

3

Java

■ Windows

- Download der JDK 7 von <http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html>
- Architektur: x86 = 32bit, x64 = 64bit
- Java-Binaries zu PATH hinzufügen

■ Linux

- `sudo apt-get install openjdk-7-jdk`

■ Mac

- Download von oracle.com

- **nicht:** MS Word, LibreOffice
- Texteditoren mit Syntaxhervorhebung:
 - Windows: Notepad++
 - Linux: gedit, vim, ...
 - Mac: z.B. TextWrangler
- anfangs **KEINE** IDEs wie Eclipse!

```
public class HelloWorld {  
    // Die main-Methode wird beim Starten  
    // als erstes ausgeführt  
    public static void main(String[] args) {  
        System.out.println("Hello World!");  
    }  
}
```

```
public class HelloWorld {  
    // Die main-Methode wird beim Starten  
    // als erstes ausgeführt  
    public static void main(String[] args) {  
        System.out.println("Hello World!");  
    }  
}
```

- `cd <Verzeichnis>` zum Verzeichniswechseln
- `'javac'` zum kompilieren
 - Befehl: `javac HelloWorld.java`
 - Erstellt `HelloWorld.class`
- `'java'` zum ausführen
 - Befehl: `java HelloWorld`
 - nimmt Klassennamen als Parameter, nicht Dateinamen (.class)

■ Primitive Datentypen:

byte	8 bit	(-128 bis +127)
short	16 bit	(-32000 bis +32000)
int	32 bit	Ganzzahlen (-2 Mrd bis 2 Mrd)
long	64 bit	(-viel bis +viel)
float	32 bit	ungenauere Gleitkommazahlen
double	64 bit	genaue Gleitkommazahlen
boolean		Wahrheitswerte (true oder false)
char	16 bit	Zeichen, z.B. 'a'

■ String

- für Zeichenketten
- kein primitiver Datentyp
- Klasse: `java.lang.String`
- Verkettung: `String abc = "abc" + "def";`

■ Deklaration

- `<typ> <name>;`
- z.B.: `int zaehler;`

■ Zuweisung

- `<name> = <wert>;`
- z.B.: `zaehler = 1;`

■ Deklaration + Zuweisung

- `int zaehler = 1;`
- `int zaehler2 = zaehler + 1;`

- +, /, -, * wie bekannt
- Vergleiche: $x < y$, $y > x$, $x \leq y$, $y \geq x$, $==$
 - `int x = 3 + 3;`
 - `boolean xLargerThan5 = (x > 5);`

- +, /, -, * wie bekannt
- Vergleiche: $x < y$, $y > x$, $x \leq y$, $y \geq x$, $==$
 - `int x = 3 + 3;`
 - `boolean xLargerThan5 = (x > 5);`
- Ganzzahlen und Kommazahlen sind strikt getrennt
 - `5 / 3 == 1`
 - `5 % 3 == 2` (Modulo/Divisionsrest)
 - `5.0 / 3.0 == 1.66666666666666674068153497501043602...`

- `+`, `/`, `-`, `*` wie bekannt
- Vergleiche: `x < y`, `y > x`, `x <= y`, `y >= x`, `==`
 - `int x = 3 + 3;`
 - `boolean xLargerThan5 = (x > 5);`
- Ganzzahlen und Kommazahlen sind strikt getrennt
 - `5 / 3 == 1`
 - `5 % 3 == 2` (Modulo/Divisionsrest)
 - `5.0 / 3.0 == 1.66666666666666674068153497501043602...`
- Casting (Konvertieren von Typen)
 - Implizites Upcasting (kleinere Datentypen in größere)
 - `short x = 3;`
`int y = x; // geht`
`byte z = y; // Fehler`
 - Explizites Casting: `byte z = (byte) y;`
- Überläufe

- Reale Objekte als Java-Objekte abbilden
- Java-Objekte sind durch Klassen definiert
- Klassen enthalten Attribute (Eigenschaften) und Methoden (Funktionen)
- Objekt: Instanz einer Klasse
 - Ein wirkliches Fahrzeug: mein Opel Corsa
 - Eigene Werte für Attribute: `wheelCount = 4;`
 - Methoden sind für alle Instanzen gleich

```
class Vehicle {
    int wheelCount;
    String color;
    Gear gear;

    public static void main(String[] args) {
        Vehicle v1 = new Vehicle(); // Konkretes Objekt
        v1.wheelCount = 4;
        v1.color = "schwarz";
        v1.gear = new Gear();
    }

    void accelerate(double x) {
        // accelerates for x seconds.
    }
}

class Gear {
    int yearOfManufacture;
}
```