

Programmieren Tutorium 6

Wer? Florian Tobias Schandinat

Wo? 50.34, Raum -108

Wann? jeden Montag 17:30-19:00

Material online

http://github.com/schandinat/programmieren_ws10

Primärziel

Konzept der objekt-orientierten Programmierung verstehen und es in Form von (Java) Programmen umsetzen können

Diese Veranstaltung richtet sich auch an StudentInnen, die noch nie in ihrem Leben programmiert haben!

Primärziel

Konzept der objekt-orientierten Programmierung verstehen und es in Form von (Java) Programmen umsetzen können

Diese Veranstaltung richtet sich auch an StudentInnen, die noch nie in ihrem Leben programmiert haben!

Sekundärziel

Google (Suchmaschine) verstehen und nachbauen

Übungsbetrieb

14 täglich

6 Übungsblätter, insgesamt ca. 120 Punkte

Unbenoteter Übungsschein: ≥ 60 Punkte

Anmeldung zum Schein unter <https://studium.kit.edu/>

Abgabe: **Praktomat, montags bis 13:00**

<https://praktomat.info.uni-karlsruhe.de/>

Übungsbetrieb

14 tägig

6 Übungsblätter, insgesamt ca. 120 Punkte

Unbenoteter Übungsschein: ≥ 60 Punkte

Anmeldung zum Schein unter <https://studium.kit.edu/>

Abgabe: **Praktomat, montags bis 13:00**

<https://praktomat.info.uni-karlsruhe.de/>

Disclaimer

Jetzt ausfüllen und abgeben!

Demo

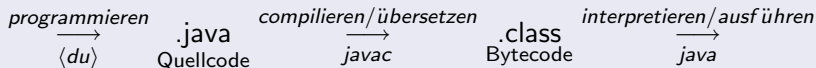
- Niemand schreibt ab!

- Niemand schreibt ab!
- Im Tutorium redet immer nur eine Person

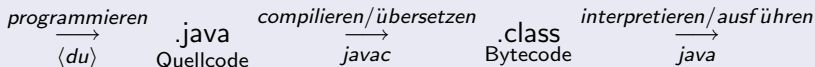
- **Niemand schreibt ab!**
- Im Tutorium redet immer nur eine Person
- Fragen sofort stellen
Das Tutorium ist dafür da um eure Fragen zu beantworten!

Habe ich etwas vergessen?

Von der Quelle bis zur Mündung



Von der Quelle bis zur Mündung



Hinweise

- Quellcode wird in Textdateien (plaintext) gespeichert
- Immer nur eine Klasse pro Datei
- $\langle \text{Dateiname} \rangle = \langle \text{Klassenname} \rangle . \text{java}$
- Nur Klassen mit einer main-Methode sind ausführbar

HelloWorld.java

```
1  /**
2   * This class implements a little sample program that prints "Hello World!"
3   */
4  class HelloWorld {
5      /**
6       * This method prints "Hello World!"
7       *
8       * @param args unused
9       */
10     public static void main(String[] args) {
11         System.out.println("Hello World!");
12     }
13 }
```

Hello World!

HelloWorld.java

```
1  /**
2   * This class implements a little sample program that prints "Hello World!"
3   */
4  class HelloWorld {
5      /**
6       * This method prints "Hello World!"
7       *
8       * @param args unused
9       */
10     public static void main(String[] args) {
11         System.out.println("Hello World!");
12     }
13 }
```

Wie compiliert man das und führt es ggf. hinterher aus?

Hello World!

HelloWorld.java

```
1  /**
2   * This class implements a little sample program that prints "Hello World!"
3   */
4  class HelloWorld {
5      /**
6       * This method prints "Hello World!"
7       *
8       * @param args unused
9       */
10     public static void main(String[] args) {
11         System.out.println("Hello World!");
12     }
13 }
```

Wie compiliert man das und führt es ggf. hinterher aus?

javac HelloWorld.java

Hello World!

HelloWorld.java

```
1  /**
2   * This class implements a little sample program that prints "Hello World!"
3   */
4  class HelloWorld {
5      /**
6       * This method prints "Hello World!"
7       *
8       * @param args unused
9       */
10     public static void main(String[] args) {
11         System.out.println("Hello World!");
12     }
13 }
```

Wie compiliert man das und führt es ggf. hinterher aus?

```
javac HelloWorld.java
```

```
java HelloWorld
```


Hello World!

HelloWorld.java

```
1  /**
2   * This class implements a little sample program that prints "Hello World!"
3   */
4  class HelloWorld {
5      /**
6       * This method prints "Hello World!"
7       *
8       * @param args unused
9       */
10     public static void main(String[] args) {
11         System.out.println("Hello World!");
12     }
13 }
```

Wie compiliert man das und führt es ggf. hinterher aus?

```
javac HelloWorld.java
java HelloWorld
```

Ausgabe

Hello World!

Was ist (k)eine Textdatei?

.doc, .docx, .odt

Was ist (k)eine Textdatei?

.doc, .docx, .odt

sind **keine** Textdateien!

Sie enthalten neben dem eigentlich Text auch noch Formatierungen und Metainformationen!

Unter anderem deshalb programmiert man **nicht** mit Microsoft Word oder Open/Libre Office.

Was ist (k)eine Textdatei?

`.doc, .docx, .odt`

sind **keine** Textdateien!

Sie enthalten neben dem eigentlich Text auch noch Formatierungen und Metainformationen!

Unter anderem deshalb programmiert man **nicht** mit Microsoft Word oder Open/Libre Office.

`.txt, .java`

sind Textdateien.

Sie enthalten wirklich nur den angezeigten Text!

Man erstellt Sie mit einem Texteditor (nächste Folie). Zum Programmieren bevorzugt man solche die Syntax-Highlighting unterstützen.

Texteditoren

Windows: Notepad++, Notepad

Linux: gedit, Kate

Mac: TextEdit, SimpleText

...und viele andere, welchen ihr benutzt bleibt euch überlassen!

Java Development Kit (JDK)

- Sun Oracle JDK: <http://java.sun.com/javase/downloads/>
- OpenJDK: <http://openjdk.java.net/install/>

Hinweis: Unter Mac OS ist JDK standardmäßig installiert

Java API - Dokumentation

<http://download.oracle.com/javase/6/docs/api/>

Eine kleine Aufgabe für zu Hause

Aufgabe

HelloWorld.java kompilieren und ausführen

Fragen und Probleme

gerne auch per E-Mail an FlorianSchandinat@gmx.de

Klasse und Objekt

Klasse

Abstraktion von Objekten

Klasse

Abstraktion von Objekten

- Attribute
- Methoden

Klasse

Abstraktion von Objekten

- Attribute
- Methoden

Objekt

Instanz einer Klasse

Klasse

Abstraktion von Objekten

- Attribute
- Methoden

Objekt

Instanz einer Klasse

- Identität (\leftrightarrow Instanz)
- Zustand (\leftrightarrow Attribute)
- Verhalten (\leftrightarrow Methoden)

Woraus besteht ein Attribut?

Woraus besteht ein Attribut?

- Typ
- Bezeichner (Attributname)

Woraus besteht ein Attribut?

- Typ
- Bezeichner (Attributname)

Beispiele:

```
int anzahl;
```

```
int gewicht; //in kg
```

Woraus besteht ein Attribut?

- Typ
- Bezeichner (Attributname)

Beispiele:

```
int anzahl;
```

```
int gewicht; //in kg
```

Typ

- Elementare Datentypen
- Klassen

Wichtig

Es werden die Werte direkt gespeichert

Typ – Elementare Datentypen

Wichtig

Es werden die Werte direkt gespeichert

Welche gibt es und was speichern sie?

Wichtig

Es werden die Werte direkt gespeichert

Welche gibt es und was speichern sie?

- boolean – Wahrheitswert: true, false

Wichtig

Es werden die Werte direkt gespeichert

Welche gibt es und was speichern sie?

- boolean – Wahrheitswert: true, false
- int – Ganzzahl: 1, 2, 670, -210

Wichtig

Es werden die Werte direkt gespeichert

Welche gibt es und was speichern sie?

- boolean – Wahrheitswert: true, false
- int – Ganzzahl: 1, 2, 670, -210
- float – Kommazahl (einfache Genauigkeit): 1, 0.1f, -0.5f, 1E-10f

Wichtig

Es werden die Werte direkt gespeichert

Welche gibt es und was speichern sie?

- boolean – Wahrheitswert: true, false
- int – Ganzzahl: 1, 2, 670, -210
- float – Kommazahl (einfache Genauigkeit): 1, 0.1f, -0.5f, 1E-10f
- double – Kommazahl (doppelte Genauigkeit): -0.5, 1E-50

Wichtig

Es werden die Werte direkt gespeichert

Welche gibt es und was speichern sie?

- boolean – Wahrheitswert: true, false
- int – Ganzzahl: 1, 2, 670, -210
- float – Kommazahl (einfache Genauigkeit): 1, 0.1f, -0.5f, 1E-10f
- double – Kommazahl (doppelte Genauigkeit): -0.5, 1E-50
- char – Einzelnes Zeichen: 'a', 'b', '1', '2', '-', '#'
- ...

Wichtig

Es werden lediglich Referenzen auf die Objekte gespeichert

Wichtig

Es werden lediglich Referenzen auf die Objekte gespeichert

Vordefinierte Klassen

Wichtig

Es werden lediglich Referenzen auf die Objekte gespeichert

Vordefinierte Klassen

- String – Ganze Zeichenketten: "Hello World!", "-13"

Wichtig

Es werden lediglich Referenzen auf die Objekte gespeichert

Vordefinierte Klassen

- String – Ganze Zeichenketten: "Hello World!", "-13"
- Integer – Ganzzahl (analog zu int, kann aber auch *null* sein)
- ...

sollen ... sein

- aussagekräftig (nicht: *a*, *b*, *c*, *attribut1*, *attribut2*)
- kurz und prägnant (keine Romane)
- mit einem kleinen Anfangsbuchstaben (*gewicht* statt *Gewicht*)
- bei aus mehreren Wörtern zusammengesetzten Namen die einzelnen Wortanfänge mit Großbuchstaben markiert (*linkesRad*)

sollen ...

- ergänzende Informationen liefern
- nicht das offensichtliche (Programmiersprache) beschreiben

sollen ...

- ergänzende Informationen liefern
- nicht das offensichtliche (Programmiersprache) beschreiben

3 Arten von Kommentaren

- Einzeiliger Kommentar `// ...`
- Mehrzeiliger Kommentar `/* ... */`
- JavaDoc Kommentar `/** ... */`
folgt genauen Regeln, mehr dazu ein anderes Mal

sollen ...

- ergänzende Informationen liefern
- nicht das offensichtliche (Programmiersprache) beschreiben

3 Arten von Kommentaren

- Einzeiliger Kommentar `// ...`
- Mehrzeiliger Kommentar `/* ... */`
- JavaDoc Kommentar `/** ... */`
folgt genauen Regeln, mehr dazu ein anderes Mal

Wichtig

Der Code soll auch ohne Kommentare verständlich sein!

Auto

**Wer meint, dass er/sie das 1.
Übungsblatt jetzt im Prinzip
lösen kann?**

TODO

- ① Einreichen einer Lösung für das 1. Übungsblatt im Praktomat bis **1.11.2010, 13:00**
- ② Anmelden für den Übungsschein auf <https://studium.kit.edu/> bis **31.3.2011**
- ③ HelloWorld.java compilieren und ausführen

Vielen Dank für die Aufmerksamkeit!

...und viel Spaß beim Programmieren :)