Einführung in die Programmierung

by André Karge

Übung - Introduction

Organisation

Heute

- Organisatorisches
- Java Installation
- erstes Programm schreiben
- IDE installieren
- Java Projekt erstellen
- Programmstrukturen
- Datentypen / Zahlendarstellungen

Übungsleitung

- Übungsleiter: M. Sc. André Karge
- Büro: Digital Bauhaus Lab (DBL)
  Bauhausstr. 9a - Raum 308
- Mail: andre.karge[at]uni-weimar.de
- Webseite: Intelligente Softwaresysteme

Übungsstruktur

Übungstermine

- Wöchentliche Übungstermine immer Donnerstags hier im Lint-Pool
- 2 Übungseinheiten
- Übungseinheit A: Donnerstag 15:15 - 16:45 Uhr
- Übungseinheit B: Donnerstag 17:00 - 18:30 Uhr

Übungsstruktur

Übungstermine

- Wöchentliche Übungstermine immer Donnerstags hier im Lint-Pool
- 2 Übungseinheiten
- Übungseinheit A: Donnerstag 15:15 - 16:45 Uhr
- Übungseinheit B: Donnerstag 17:00 - 18:30 Uhr

Teams

- jeweils max. 6 Teams pro Übungseinheit
- 2-3 Personen pro Team
- Merkt euch eure Teamnummer und gebt sie bei jeder Abgabe mit an

Übungsstruktur

Informationen

- Ihr könnt zu jeder Zeit Fragen stellen
- Erste Anlaufstelle, falls ihr irgendwo nicht weiter kommt, ist das Buch "Java ist auch eine Insel"
- Tipp: lest Kapitel 1-2 und versucht die Beispiele zu programmieren

Übungsstruktur

Abgabeinformationen

- Textdokumente **immer** als **.pdf** Datei
- Quellcode **immer** als **.java** Datei(en)
- Angabe der Gruppennummer in **allen** abgegebenen Dateien
  - im Quellcode als Kommentar am Anfang der Dateien
- Wenn mehrere Dateien abgegeben werden (Bspw. eine PDF und eine/mehrere Quellcode Dateien):
  - **Komprimieren!** als zip | rar | tar.gz | ...
- Benennt dieses Archiv nach folgendem Schema:
  programmierung_uebungXX_gruppeYY.ZZZ
  Wobei **XX** die jeweilige Übungsnummer, **YY** die jeweile Gruppennummer und **ZZZ** die jeweile Komprimierungsendung ist

Übungsstruktur

Abgabeinformationen

- es reichen die Quellcodedateien - Bitte gebt keine kompilierten Dateien ab - damit kann ich nichts anfangen (das heißt keine .class Dateien und keine kompletten Projektverzeichnisse)
- Schickt eure Lösungen immer **vor** Ablauf der Frist an andre.karge[at]uni-weimar.de
- Prüft euren Code ein letztes mal bevor ihr eure Abgabe macht - **Wenn der Code nicht läuft gibt es Punktabzüge!**
- **Nicht-Einhaltung der Abgabekriterien führen auch zu Punktabzügen!**

Übungsstruktur

Übungen

- Es wir über das Semester ca. 12 Belege geben
- Zudem muss jede Gruppe einmal die Lösungen eines Beleges vorstellen
- Gruppen können sich ab Veröffentlichung des neuen Beleges darauf bei mir anmelden - Hier gilt das "*first come - first serve*"-Prinzip
- Wenn sich keine Gruppe anmeldet werde ich zufällig eine der übrigen Gruppen auswählen

Zulassung

- **Mindestens 50% aller Punkte der Belege + erfolgreiche Vorstellung**

Java

Java

Installation - Windows

- Webseite von Oracle besuchen und dort Java JDK-8 Developer Version (Java SE Development Kit 8) herunter laden (link)
- JDK installieren
- Java PATH Variable setzen:
  - Start $\rightarrow$ "Umgebungsvariablen" eingeben $\rightarrow$ "Umgebungsvariablen bearbeiten" anklicken
  - "Erweitert"-Tab auswählen und Umgebungsvariablen-Button anklicken
  - Variable "JAVA_HOME" anlegen und zum JDK Verzeichnis zeigen lassen
  - bspw.: $C:\mathit{ProgrammFiles}\backslash\mathit{Java}\backslash\mathit{jdk}-8$
  - PATH Variable suchen, editieren $\rightarrow \%\mathit{Java\_Home}\backslash\mathit{bin}\%$ anfügen

Java

Installation - Linux

- Mit Paketmanager installieren
- bspw. mit Ubuntu:
- ```
  sudo apt update
  sudo apt install openjdk-8-jre openjdk-8-jdk openjdk-8-source
  ```
  ```

Java Installation überprüfen

Windows

Start → cmd

Linux

Terminal öffnen

```
java -version
> openjdk version "1.8.0_222"
> OpenJDK Runtime Environment (build 1.8.0_222-b05)
> OpenJDK 64-Bit Server VM (build 25.222-b05, mixed mode)

javac -version
> javac 1.8.0_222
```

Java Einführung

Erstes Java Programm

- Texteditor eurer Wahl öffnen
  - Notepad, Notepad++, Gedit, Visual Studio Code, Sublime Text, Atom, Vim, Emacs, ...
- Neue Datei anlegen "HelloWorld.java"
- Quellcode schreiben

Erstes Java Programm

Einstiegspunkt - Die Main Methode

```
public class HelloWorld {
  public static void main(String[] args) {
    // Hier kommt Quellcode hin
    System.out.println("Hello World");
  }
}
```

Erstes Java Programm

Kompilieren per Hand

- Man kann den Java Compiler per Hand aufrufen um Java Dateien in Bytecode übersetzen zu lassen
- Ist das, was passiert, wenn man in einer IDE den Compile Button drückt

```
javac HelloWorld.java  # Compiling der Java Datei
java HelloWorld  # Ausführen des Bytecodes
> Hello World
```

Programmierwerkzeuge

Integrated Development Environment

- IDE = Editor + Compiler + Debugger
- Für Java:
  - Eclipse (link)
  - IntelliJ (link)
- wir werden für die Übung IntelliJ verwenden

Quelle: http://www.tutego.de/seminare/java-schulung/eclipse-seminar.html

Quelle: https://www.zoschke.com/intellij-idea

IntelliJ

Projekterstellung

- neues Command-Line Projekt erstellen
- neue Class-Datei im default Package erstellen
- Quellcode schreiben
- Programm ausführen

live Demo

Programmstrukturen

Standardgerüst

- jedes Programm muss eine main-Methode implementieren um es ausführen zu können

```
public static void main(String[] args) {

}
```

Programmstrukturen

Standardgerüst

- jedes Programm muss eine main-Methode implementieren um es ausführen zu können

```
public static void main(String[] args) {

}
```

- Java Dateien müssen den selben Namen haben wie die Klasse, die sie implementieren:
  - Datei *HelloWorld.java* implementiert die Klasse *public class HelloWorld {}*

Programmstrukturen

Standardgerüst

- jedes Programm muss eine main-Methode implementieren um es ausführen zu können

```
public static void main(String[] args) {

}
```

- Java Dateien müssen den selben Namen haben wie die Klasse, die sie implementieren:
  - Datei *HelloWorld.java* implementiert die Klasse *public class HelloWorld {}*
- Klassenbezeichner werden in der Regel **Groß** geschrieben
- Methoden- und Variablenbezeichner werden in der Regel **klein** geschrieben

Programmstrukturen

Standardsyntax

- Kommentare
  - Enzeilig ab // ...
  - Mehrzeilig ab /* ... bis */
- Klassen und Methoden haben einen Body, welcher mit geschweiften Klammern geöffnet { und geschlossen } wird
- Methoden bestehen aus einer Verkettung von Anweisungen, wobei jede Anweisung mit einem Semikolon ; terminiert wird

Programmstrukturen

Standardsyntax

- Kommentare
  - Enzeilig ab // ...
  - Mehrzeilig ab /* ... bis */
- Klassen und Methoden haben einen Body, welcher mit geschweiften Klammern geöffnet { und geschlossen } wird
- Methoden bestehen aus einer Verkettung von Anweisungen, wobei jede Anweisung mit einem Semikolon ; terminiert wird

```
public class HelloWorld {
  public static void main(String[] args) {
    int a = 1;              // Anweisung 1
    int b = 2;              // Anweisung 2
    int c = a + b;          // Anweisung 3
    System.out.println(c); // Anweisung 4
  }
}
```

Übungsblatt

- Online: nächste Woche Montag ca. 16:00
- Abgabe: zwei Wochen darauf am Montag 11:00
- **Zu spät abgegebene Belege werden mit 0 Punkten bewertet!**
- Nicht-Einhaltung der Abgabekriterien führt auch zu Punktabzügen!

Übungsblatt

Inhalt

- Einer- und Zweierkomplement berechnen
- Standardoperatoren
- Klassen-Theorie

Übungsblatt

Inhalt

- Einer- und Zweierkomplement berechnen
- Standardoperatoren
- Klassen-Theorie

nächste Übung

- nächste Woche: Reformationstag - keine Übung

Nächste Übung

- Grundlagen zu Datentypen, Anweisungen und Methoden, Einer-Zweierkomplemente und Operatoren
- Fragerunde des ersten Belegs

Fragen?