



Informatik I – Eprog HS13

Übung 1

1 Aufgabe: Hello World

1.1 Lernziele

1. SDK installieren können.
2. Klassen mit Texteditor erstellen und anschliessend kompilieren können.
3. Programm ausführen können.

1.2 Aufgabenstellung

a) SDK Installation

Installieren Sie das [aktuelle Java SE Development Kit \(JDK\)](#)¹ (Hinweis: Mac OS X beinhaltet bereits ein JDK) sowie einen Texteditor mit Syntaxhighlighting.² Kompilieren Sie dann das nachfolgende Programm auf der Kommandozeile und führen Sie es anschliessend aus.

Vorgehen

1. Falls Sie bei der Installation des JDKs Schwierigkeiten haben, hilft Ihnen das bereitgestellte [Tutorial](#) weiter.
2. Kopieren Sie die untenstehenden Klassen in je ein separates Textfile und speichern Sie dieses unter dem `<Klassennamen>.java` ab (z. B. `HelloWorld.java`). Benutzen Sie dazu einen der vorgeschlagenen Texteditoren.
3. Kompilieren Sie die soeben erstellten Klassen und führen Sie anschliessend den `TestDriver`³ `HelloWorldTest` aus. Bei Unklarheiten empfiehlt es sich u. U., das Online Modul der Aufgabe 4 vorzuziehen.

¹aktuelle Version ist JDK 7 Update 40

²z. B. [JEdit](#), [Notepad++](#), [Vim](#), [Emacs](#), [TextWrangler](#), etc.

³Unter `TestDriver` verstehen wir eine Klasse, die eine `main()`-Methode (Einstiegspunkt jedes Programms) implementiert und deren Hauptzweck die Erstellung (Instanzierung) und das Testen von Objekten ist

```
1 public class HelloWorld {
2
3     /**
4     * Simply prints "Hello World" to the console.
5     */
6     public void printHelloWorld() {
7         System.out.println("Hello World!");
8     }
9 }
```

Listing 1: HelloWorld Klasse

```
1 public class HelloWorldTest {
2
3     /**
4     * Entry point for a Java program. Creates a HelloWorld object and
5     * calls printHelloWorld() on it.
6     */
7     public static void main(String[] args) {
8         HelloWorld myHelloWorldObject = new HelloWorld();
9         myHelloWorldObject.printHelloWorld();
10    }
11 }
```

Listing 2: HelloWorld TestDriver

b) Programmanpassung

Lassen Sie das Programm einen anderen Text als „Hello World“ ausgeben.

2 Aufgabe: Cuboid

2.1 Lernziele

1. Programmcode lesen und verstehen können.
2. Programm mit Hilfe eines TestDrivers testen können.
3. Programmlogik ändern können.

2.2 Aufgabenstellung

a) Cuboid erstellen

Schreiben Sie den untenstehen Code ab. Die Variablen `length`, `width` und `height` stehen dabei für die Länge, Breite und Höhe eines Quaders. `printVolume()` sollte das Volumen dieses Quaders berechnen und auf der Konsole ausgeben. Erstellen Sie einen TestDriver, welchen Sie verwenden, um dieses Verhalten zu testen.

```
1  /**
2   * Represents a cuboid.
3   */
4  public class Cuboid {
5      private int length = 50;
6      private int width = 30;
7      private int height = 10;
8
9      /**
10     * Calculates the volume of this cuboid
11     * and prints the result to the console.
12     */
13     public void printVolume() {
14         System.out.print("The volume of this cuboid is ");
15         System.out.println(length * width * height);
16     }
17 }
```

Listing 3: Cuboid Klasse

b) Programmanpassung

Nachdem das Programm läuft, können Sie dieses erweitern. Versuchen Sie es dahingehend anzupassen, dass es fähig ist, analog zum Volumen, die Oberfläche des Quaders korrekt auszugeben. Hinweis: Sowohl Klasse als auch TestDriver sind anzupassen.

3 Aufgabe: Codeverständnis

3.1 Lernziele

1. Verständnis von primitiven Datentypen, die Auswertung arithmetischer Ausdrücke und die Typkonversion vertiefen.
2. Codelesen und Codeverständnis trainieren.

3.2 Aufgabenstellung

a) Typkonversion

Werte welcher Datentypen können einer Variablen eines anderen Datentyps zugewiesen werden, ohne dass eine expliziter Type Cast durchgeführt werden muss? Vervollständigen Sie folgende Zuweisungs-Kompatibilitäts-Tabelle.

byte	->	->	->	->	double
char	->	->	->	->	

b) Arithmetische Ausdrücke und Rechenoperationen

1. Vervollständigen Sie Tabelle 1.

Mathematischer Ausdruck	Ausdruck in Java
$rate^2 + delta$	
$\frac{10-a}{bc}$	abc - 3
=	

Table 1: Arithmetische Ausdrücke in Java

2. Zeigen Sie drei verschiedene Wege auf, um 1 von einer `int`-Variablen zu subtrahieren.

c) Code Snippets

Welche der nachfolgenden acht Code Snippets sind syntaktisch korrekt und wie lautet in diesem Fall die Ausgabe? Begründen Sie Ihre Antwort und erklären Sie insbesondere unerwartete Ergebnisse.

```
1 int x = 10;
2 int y = 20;
3 System.out.println(x + y++);
4 System.out.println(y);
```

Listing 4: Snippet 1

```
1 byte x = 120;
2 byte y = 10;
3 x = (byte) (x + y);
4 System.out.println(x);
```

Listing 5: Snippet 2

```
1 short x = 128;
2 byte y = (byte)x;
3 System.out.println(y);
```

Listing 6: Snippet 3

```
1 byte x;
2 short y = 150;
3 x = y;
4 System.out.println(x);
```

Listing 7: Snippet 4

```
1 short x;
2 byte y = 100;
3 x = y;
4 System.out.println(x);
```

Listing 8: Snippet 5

```
1 char x = 'x';
2 System.out.println((long)x);
3
4 x = 65;
5 System.out.println(x);
6
7 x = '7';
8 System.out.println((int)x);
```

Listing 9: Snippet 6

```
1 System.out.println(1/2);
2 System.out.println(1/2d);
3 System.out.println((double)(1/2));
4 System.out.println((double)1/2);
5 System.out.println(1.0/2);
```

Listing 10: Snippet 7

```
1 double x = 7;
2 double y = 30;
3 System.out.println(x / (y % ++x));
```

Listing 11: Snippet 8

4 Aufgabe: Online Modul

4.1 Lernziele

1. Den grundlegenden Softwareentwicklungs-Prozess verstehen.
2. Grundbegriffe der Softwareentwicklung erlernen.
3. Ein einfaches Java-Programm schreiben, kompilieren und ausführen können.
4. Einfache Kompilierungsfehler verstehen und beheben können.

4.2 Aufgabenstellung

1. Arbeiten Sie das *Module 1* im [OLAT](#) durch. Melden Sie sich dazu für den Kurs *INF_11_HS Informatik I + Programmierung für Mathematik* an und klicken Sie dann im Menu auf den Link *eLearning*.
2. Lösen Sie nach dem Selbststudium den kurzen Selbsttest um Ihr neues Wissen zu überprüfen.