



Programmierung für Mathematik (HS13)

Übung 1

1 Aufgabe: Hello World

1.1 Lernziele

1. MATH Account erstellen und nutzen.
2. Klassen mit Texteditor erstellen und anschliessend kompilieren können.
3. Programm ausführen können.

1.2 Aufgabenstellung

a) MATH-Account

Falls Sie noch keinen MATH-Account haben, folgen Sie den Anweisungen auf der [Homepage des Institut für Mathematik](#), um einen MATH-Account zu erstellen. Loggen Sie sich anschliessend in Ihre MATH-Umgebung ein (unter [student1.math.uzh.ch](#) oder [student2.math.uzh.ch](#) die Daten vom MATH-Account eingeben, dann unter *Applications* eine der *Gnome full*-Versionen anklicken). Kompilieren Sie das Programm, welches auf der nächsten Seite abgebildet ist, auf der Kommandozeile und führen Sie es anschliessend aus.

Vorgehen

1. öffnen Sie den Texteditor jEdit (Applications > Programming > jEdit)
2. Kopieren Sie die beiden Klassen, welche auf der nächsten Seite abgebildet sind, in je eine separate Textdatei und speichern Sie diese unter dem <Klassennamen>.java ab (im Falle der HelloWorld-Klasse müsste der Dateiname HelloWorld.java lauten).
3. Kompilieren Sie die soeben erstellten Klassen: Öffnen Sie das Terminal (schwarzes Symbol oben in der Menüleiste), wechseln Sie in das entsprechende Verzeichnis, in dem die .java-Dateien gespeichert sind (z.B. `cd Desktop/Informatik1/Uebung1` eintippen und zum bestätigen [Enter] drücken. Falls ein Ordnername ein Leerzeichen enthält, muss dieser in Anführungszeichen geschrieben werden. Generell wird empfohlen, Leerzeichen, Sonderzeichen sowie Umlaute zu vermeiden.). Geben Sie nun

```
javac HelloWorldTest.java
```

in die Kommandozeile ein. Es werden automatisch beide Dateien kompiliert, da `HelloWorld.java` von `HelloWorldTest.java` verwendet wird.

4. Führen Sie den TestDriver¹ `HelloWorldTest` aus. Geben Sie dazu

```
java HelloWorldTest
```

in die Kommandozeile ein.

```
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.

¹Unter TestDriver verstehen wir eine Klasse, die eine `main()`-Methode (Einstiegspunkt eines jeden Java-Programms) implementiert und deren Hauptzweck die Erstellung (Instanzierung) und das Testen von Objekten ist

2 Aufgabe: Object-oriented Programming (OOP)

2.1 Lernziele

1. Wissen, was Klassen, Objekte, Instanzvariablen und Methoden sind.

2.2 Aufgabenstellung

1. Lesen Sie den folgenden Code genau durch und entscheiden Sie, ob es sich bei den folgenden Programmbestandteilen (siehe Listing 3 und Listing 4) um eine *Klasse*, ein *Objekt*, eine *Instanzvariable*, oder um eine *Methode* handelt:

1. Car
2. carColor
3. age
4. setCarColor
5. setCarAge
6. honk
7. TestDriver
8. main
9. myFerrari

```
1 public class Car {
2     private String carColor;
3     private int age;
4
5     public void SetCarColor(String myCarColor) {
6         carColor = myCarColor;
7     }
8
9     public void setCarAge(int myCarAge) {
10        age = myCarAge;
11    }
12
13    public void honk() {
14        System.out.println("honk!");
15    }
16 }
```

Listing 3: Car

```
1 public class TestDriver {
2     public static void main(String[] args) {
3
4         Car myFerrari = new Car();
5
6         myFerrari.setCarColor("red");
7         myFerrari.setCarAge(23);
8         myFerrari.honk();
9
10    }
11 }
```

Listing 4: TestDriver

3 Aufgabe: Calculator

3.1 Lernziele

1. Programmcode verstehen können.
2. Programm mit Hilfe eines TestDrivers testen können.
3. Ein bestehendes Programm anpassen können.

3.2 Aufgabenstellung

a) Calculator erstellen

Schreiben Sie den untenstehen Code ab. Erstellen Sie ausserdem selbstständig einen TestDriver, welchen Sie verwenden um die Methode `add` von `Calculator` zu testen.

```
1 public class Calculator {
2
3     /**
4      * Method to add number1 to number2 and
5      * print the result to the console
6      * @param number1
7      * @param number2
8      */
9     public void add(int number1, int number2) {
10
11         System.out.println("The sum of " + number1
12             + " and " + number2 + " is "
13             + (number1 + number2));
14     }
15
16     /**
17      * Method to subtract number2 from number1
18      * and print the result to the console
19      * @param number1
20      * @param number2
21      */
22     public void subtract(int number1, int number2) {
23
24     }
25
26     /**
27      * Method to multiply number1 with number2
28      * and print the result to the console
29      * @param number1
30      * @param number2
31      */
32     public void multiply(int number1, int number2) {
33
34     }
35 }
```

```
36     /**
37     * Method to divide number1 by number2 and
38     * print the result to the console
39     * @param number1
40     * @param number2
41     */
42     public void divide(int number1, int number2) {
43
44         //TODO: don't forget to handle the division by 0
45
46     }
47 }
```

Listing 5: Calculator Klasse

b) Programmanpassung

Nachdem das Programm läuft, können Sie dieses um zusätzliche Funktionalität erweitern. Ergänzen Sie dazu die Methoden `subtract`, `multiply` und `divide` sinnvoll (das heisst, wie bei der Methode `add` soll jeweils das Ergebnis der Operation auf der Konsole ausgegeben werden). Vergessen Sie nicht, den Fall „Division durch 0“ speziell zu behandeln. Testen Sie mit Ihrem TestDriver die geänderten Methoden.

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 *Self-Study Module 1* im [OLAT](#) durch. Melden Sie sich dazu für den Kurs *INF_13_HS Informatik I* 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.