



# Programmierung für Mathematik HS10

## Übung 1

### 1 Aufgabe: Hello World

#### 1.1 Lernziele

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

#### 1.2 Aufgabenstellung

##### a) MATH-Account

Folgen Sie den Anweisungen von 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](http://student1.math.uzh.ch) oder [student2.math.uzh.ch](http://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 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 ist es aber besser Leer- und Sonderzeichen, wie 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 TestDriver<sup>1</sup> `HelloWorldTest` aus. Geben Sie dazu

```
java HelloWorldTest
```

in die Kommandozeile ein.

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

**Listing 1:** HelloWorld Klasse

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

**Listing 2:** HelloWorld TestDriver

## b) Programmanpassung

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

---

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

## 2 Aufgabe: OOP

### 2.1 Lernziele

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

### 2.2 Aufgabenstellung

1. Lesen Sie sich den folgenden Code genau durch:

```
1 public class Dog {
2     private String coatColor;
3     private int age;
4
5     public void setData(String myCoat, int myAge) {
6         coatColor = myCoat;
7         age = myAge;
8     }
9
10    public void bark() {
11        System.out.println("wuff!");
12    }
13 }
```

```
1 public class TestDriver {
2     public static void main(String[] args) {
3
4         Dog myDog = new Dog();
5
6         myDog.setData("black", 3);
7         myDog.bark();
8
9     }
10 }
```

2. Entscheiden Sie, ob es sich bei den folgenden Programmbestandteilen um eine *Klasse*, ein *Objekt*, eine *Instanzvariable*, oder um eine *Methode* handelt:
  - (a) Dog
  - (b) coatColor
  - (c) age
  - (d) setData
  - (e) bark
  - (f) TestDriver
  - (g) main
  - (h) myDog

## 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     public void add(int number1, int number2) {
3         System.out.println("The sum of " + number1 + " and " +
4             number2 + " is " + (number1 + number2));
5     }
6
7     public void subtract(int number1, int number2) {
8
9     }
10
11    public void multiply(int number1, int number2) {
12
13    }
14 }
```

Listing 3: Calculator Klasse

#### b) Programmanpassung

Nachdem das Programm läuft, können Sie dieses um zusätzliche Funktionalität erweitern. Ergänzen Sie dazu die Methoden `subtract` und `multiply` sinnvoll (das heisst, wie bei der Methode `add` soll jeweils das Ergebnis der Operation auf der Konsole ausgegeben werden). 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\_10\_HS Informatik I + Programmierung für Mathematik* an und klicken sie dann im Menu auf den Link *Selbststudium*.
2. Lösen sie nach dem Selbststudium den kurzen Selbsttest (*Self-Assessment > Week 1*) um ihr neues Wissen zu überprüfen.