

Daten ausgeben und einlesen

Lernziele:

- Sie können einfache Klassen in einer von Ihnen vertrauten Entwicklungsumgebung erstellen
- Sie können Daten ausgeben und auch Daten via Konsole einlesen
- Sie können Daten von Textfiles einlesen

1 Modul 226 und Modul 411

In diesem Modul werden wir mit der Programmiersprache Java arbeiten. Während Sie sich parallel dazu in den Modulen 226a und 226b vertieft mit dieser Sprache und der Objektorientierung beschäftigen werden, werden wir im Modul 411 Java als Mittel zum Zweck verwenden. Nach und nach werden wir das in den 226-er Modulen Gelernte hier anwenden.

Es wird sich nicht vermeiden lassen, dass zwischen den 226-er Modulen und dem Modul 411 gewisse Doppelspurigkeiten entstehen.

2 Entwicklungsumgebung

Sie können frei wählen, mit welcher Entwicklungsumgebung Sie arbeiten möchten.

- Eclipse
- IntelliJ
- ...

3 Klassen erstellen

Hier ist der Code einer kleinen Java-Klasse, welche verwendet werden kann, um verschiedene Personen zu begrüßen:

```
public class WelcomeMessage {
    public void sayHello(String name) {
        System.out.println("Welcome " + name);
    }

    public static void main(String[] args) {
        WelcomeMessage msg = new WelcomeMessage();
        msg.sayHello("Tom");
        msg.sayHello("Mirjam");
    }
}
```

Die main-Methode

Ein Java-Programm wird immer so ausgeführt, dass die `main`-Methode einer bestimmten Klasse aufgerufen wird. Diese muss genau so deklariert sein, wie Sie das im Beispiel sehen:

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

In dieser Methode (ein neuer Name für Funktion) wird dann üblicherweise ein Objekt erzeugt, in unserem Beispiel:

```
WelcomeMessage msg = new WelcomeMessage();
```

und von diesem Objekt wird dann eine Methode aufgerufen:

```
msg.sayHello("Tom");
```


5 Konsolen Ein- und Ausgabe

Ausgabe

Im obigen kleinen Beispiel haben Sie schon gesehen, wie Text auf der Konsole ausgegeben werden kann, nämlich mit der Methode `System.out.println`. Vielleicht ist Ihnen auch aufgefallen, wie Texte einfach mit dem `+`-Zeichen zusammengehängt werden können. Das Gleiche gilt auch für Zahlenwerte, wie das folgende kleine Beispiel zeigt:

```
int a = 5;
int b = 6;
System.out.println("Die Summe von " + a + " und " + b + " ist " + (a+b));
```

Wie würde wohl die Ausgabe aussehen, wenn beim Ausdruck `(a+b)` die Klammern weggelassen würden? Probieren Sie das aus und versuchen Sie das zu verstehen.!

Eingabe

Um Texte oder Zahlen von der Konsole einzulesen, müssen wir ein Objekt der Klasse `Scanner` (das ist eine Klasse aus der Standard Java-Library) erzeugen. Dieses Objekt gibt uns dann auf Aufforderung hin den nächsten Text oder die nächste Zahl:

```
Scanner scan = new Scanner(System.in);
int a = scan.nextInt();
String s = scan.next();
```

6 Übungen

Erstellen Sie ein neues Projekt und darin eine Klasse `AgeCalculator`, welche die folgende Methode enthält:

```
public void tellAge()
```

Diese Methode soll den Benutzer nach seinem Namen und seinem Jahrgang fragen und dann die folgende Meldung ausgeben:

```
Hallo Peter, du wirst in diesem Jahr 34 Jahre alt.
```

wobei natürlich der Name und das Alter den Benutzereingaben entsprechen sollen. Verwenden Sie für die Berechnung des Alters das aktuelle Jahr.

Tipp: Verwenden Sie die Klasse `SimpleDateFormat` um das Format zu bestimmen und die Klasse `Date` um das aktuelle Datum zu generieren.

Quelle: BBW Unterlagen, St.Dütsch. Überarbeitet 2020.