

## Übungen zu Einführung in die Informatik: Programmierung und Software-Entwicklung

### Aufgabe 1-1

### Arbeiten mit Java ohne IDE

Präsenz

In dieser Aufgabe sollen Sie ausprobieren, wie Java-Programme ohne Unterstützung durch eine IDE (Integrated Development Environment) erstellt, kompiliert und ausgeführt werden. Um diese Aufgabe bearbeiten zu können, müssen Sie Java installiert haben, wie in der Installationsanweisung für Ihre Plattform (Windows oder Mac) besprochen wird.

- Verändern Sie das Programm **Begrueßung** aus der Installationsanleitung Windows/Mac so, dass Sie selbst begrüßt werden. Speichern Sie die Datei ab. Führen Sie das Programm aus, **ohne** es zu kompilieren. Was wird in der Eingabeaufforderung ausgegeben und warum?
- Kompilieren Sie Ihr verändertes Programm und führen Sie es aus. Was wird jetzt in der Eingabeaufforderung ausgegeben und warum?
- Stellen Sie wieder den Ursprungszustand Ihres Programms her, d.h. das Programm soll nun wieder **Hallo Welt!** ausgeben. Verändern Sie Ihr Programm, indem Sie die Anführungszeichen weglassen, d.h. Ihr Programm sollte nun folgendermaßen aussehen:

```
1 public class Begrueßung {
2     public static void main(String[] args) {
3         System.out.println(Hallo Welt!);
4     }
5 }
```

Warum können Sie Ihr Programm jetzt nicht mehr kompilieren? Wie nennt man diese Art von Fehler?

- Fügen Sie die Anführungszeichen wieder in das Programm ein, ersetzen Sie jetzt aber das Wort **Welt** durch **Wlt**, d.h. Ihr Programm sollte folgendermaßen aussehen:

```
1 public class Begrueßung {
2     public static void main(String[] args) {
3         System.out.println("Hallo Wlt!");
4     }
5 }
```

Können Sie Ihr Programm jetzt kompilieren? Warum enthält dieses Programm trotzdem einen Fehler und wie nennt man diesen Fehler?

### Aufgabe 1-2

### Arbeiten mit Java mit IDE (Eclipse)

Präsenz

In dieser Aufgabe sollen Sie die Vorteile einer IDE (Integrated Development Environment) wie Eclipse gegenüber der Benutzung eines einfachen Editors erfahren. Um diese Aufgabe bearbeiten zu können, müssen Sie Java und Eclipse installiert haben, wie in der Installationsanweisung für Ihre Plattform (Windows oder Mac) besprochen wird.

- Verändern Sie das Programm **Begrueßung** aus der Installationsanleitung Windows/Mac in Eclipse so, dass Sie selbst begrüßt werden. Speichern Sie die Datei ab. Führen Sie das Programm erneut in Eclipse aus. Was wird in der Console ausgegeben und warum?

*Hinweis:* Vergleichen Sie die Ausgabe mit der Ausgabe von Aufgabe 1-1a.

- b) Stellen Sie wieder den Ursprungszustand Ihres Programms her, d.h. das Programm soll nun wieder **Hallo Welt!** ausgeben. Verändern Sie Ihr Programm, indem Sie die Anführungszeichen weglassen, d.h. Ihr Programm sollte nun folgendermaßen aussehen:

```
1 public class Begruessung {
2     public static void main(String[] args) {
3         System.out.println(Hallo Welt!);
4     }
5 }
```

Wie und wo stellt Eclipse überall dar, dass in Ihrem Programm ein Fehler enthalten ist?

- c) Fügen Sie die Anführungszeichen wieder in das Programm ein, ersetzen Sie jetzt aber das Wort **Welt** durch **Wlt**, d.h. Ihr Programm sollte folgendermaßen aussehen:

```
1 public class Begruessung {
2     public static void main(String[] args) {
3         System.out.println("Hallo Wlt!");
4     }
5 }
```

Kann Eclipse hier einen Fehler finden? Erklären Sie Ihre Antwort.

- d) Stellen Sie wieder den Ursprungszustand Ihres Programms her, d.h. das Programm soll nun wieder **Hallo Welt!** ausgeben. Fügen Sie anschließend nach jedem Wort (außer nach **"Hallo**) einen Zeilenumbruch ein und speichern Sie die Datei ab. Ihr Programm sollte dann folgendermaßen aussehen:

```
1 public
2 class
3 Begruessung {
4     public
5     static
6     void
7     main(
8     String[]
9     args) {
10    System.
11    out.
12    println(
13    "Hallo Welt!"
14    );
15 }
16 }
```

Können Sie Ihr Programm jetzt noch kompilieren und ausführen? Erklären Sie Ihre Antwort!

- e) Klicken Sie nun im Package Explorer rechts auf die Datei **Begruessung.java** und wählen Sie im Kontextmenü **Source -> Format**. Was passiert mit Ihrem Source Code und warum ist das sinnvoll?