

Übungsblatt 1

Aufgabe 1.1 (P) Hello World, IDE, MiniJava

Starten Sie eine Entwicklungsumgebung (IDE) Ihrer Wahl und machen Sie sich mit deren grundlegenden Funktionalitäten zum Programmieren (in Java) sowie der Abbildung dieser im Dateiensystem Ihres PCs vertraut:

- Erstellen eines neues Java-Projekts (\leadsto Pfad / Ordner)
- Erstellen von *Packages* innerhalb des Projekts (\leadsto Unterordner)
- Erstellen von Klassen (*class*) (\leadsto editierbare Datei im Package / Ordner) mit oder ohne enthaltenem Hauptprogramm (*main*)

Diskutieren Sie mit Ihrem Tutor etwaige von Ihrer IDE automatisch erstellte Inhalte in den editierbaren Dateien.

Binden Sie nun die MiniJava-Datei ein:

- Erstellen einer Klasse mit Zusatz „extends MiniJava“ und einem Hauptprogramm
- Download von `MiniJava.java` von der Übungswebseite
- Transferieren dieser Datei an ein Plätzchen, wo sie von der IDE gefunden wird (selber Ordner wie die Klasse, von der sie verwendet wird)

Schreiben Sie nun ein MiniJava-Programm, welches vom Benutzer zwei ganze Zahlen einliest, die Summe der Zahlen ausgibt und sich danach von alleine beendet. Führen Sie Ihr Programm innerhalb der IDE aus.

Aufgabe 1.2 (P) Java auf der Kommandozeile

Eine Entwicklungsumgebung zum Programmieren (wie z.B. *Eclipse* oder *NetBeans*) ist im Grunde nur ein besserer Text-Editor, welcher hilfreiche Funktionalität bietet (wie etwa das Arbeiten im *Debug-Modus*, um Fehler zu finden, oder das Hervorheben von Java-Schlüsselwörtern, um die Struktur des Java-Programms zu verdeutlichen). Unter dieser Oberfläche verbirgt sich im Wesentlichen ein *Compiler*, welcher Ihr Java-Programm in Computer-Befehle übersetzt, damit der Computer es ausführen kann. Per Kommandozeile ist es möglich, direkt den Compiler aufzurufen und seine Dienste in Anspruch zu nehmen. Mit dem Aufruf

```
javac Program.java
```

lässt sich der Quellcode, der in der Datei `Program.java` steht, in eine ausführbare Form bringen, den Java-Bytecode, welcher dann in die Datei `Program.class` geschrieben wird. Beim Übersetzungsvorgang wird zugleich geprüft, ob der Quellcode syntaktisch korrekt ist, d.h. ob er der Grammatik der Sprache Java entspricht. Ist dies nicht der Fall, so findet keine Übersetzung statt, sondern es werden Fehlermeldungen ausgegeben.

Um den Java-Bytecode aus der Datei `Program.class` auszuführen, ruft man nun einfach

```
java Program
```

auf der Kommandozeile auf.

Schreiben Sie den folgenden Text mit Hilfe eines Text-Editors (z.B. *Editor*, *Notepad(++)*, *Kate*, ...) und speichern Sie ihn als eine Text-Datei namens `MeinProgram.java` ab:

```
public class MeinProgram extends MiniJava {
    public static void main (String args []) {
        write("Hello World!");
    }
}
```

Wechseln Sie nun auf der Kommandozeile mit dem Befehl `cd` in das Verzeichnis, in welchem sich Ihre Datei befindet, und geben Sie die Befehle zum Übersetzen und Ausführen Ihres Programms ein. Die `MiniJava.java` muss sich im gleichen Verzeichnis wie `MeinProgram.java` befinden.

Hinweis: Evtl. müssen Sie die `PATH` und/oder `CLASSPATH` Systemvariable ändern. Eine Anleitung finden Sie dazu unter <https://docs.oracle.com/javase/tutorial/essential/environment/paths.html>

Aufgabe 1.3 (P) Probe-Abgabe

Die Hausaufgabenabgabe erfolgt über die Moodle-Homepage. Bitte beachten Sie, dass es für einige Gruppen eine extra Seite gibt. Die Seite, über die Sie Ihre Hausaufgaben abgeben können, ist nach Ihrer Einschreibung in TUMonline automatisch auf Ihrer Moodle-Startseite verlinkt. Um sich mit dem Abgabesystem vertraut zu machen, wurde die Abgabe für das 1. Aufgabenblatt eingerichtet, auch wenn es keine Hausaufgaben auf diesem Blatt gibt. Probieren Sie die Abgabe aus, indem Sie Ihr zuvor erstelltes Hello World Programm abgeben.