

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 neuen Java-Projekts (\rightsquigarrow Pfad / Ordner)
- Erstellen von *Packages* innerhalb des Projekts (\rightsquigarrow Unterordner)
- Erstellen von Klassen (*class*) (\rightsquigarrow 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. *Vim*, *Notepad(++)*, *Kate*, ...) und speichern Sie ihn als eine Text-Datei namens `MeinProgram.java` ab:

```
1 public class MeinProgram extends MiniJava {  
2     public static void main (String args[]) {  
3         write("Hello World!");  
4     }  
5 }
```

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>

Die Hausaufgabenabgabe erfolgt über Moodle; beachten Sie, dass es mehrere Moodle-Seiten gibt (um Ihre und unsere Verwirrung zu erhöhen). Bitte geben Sie Ihren Code als UTF8-kodierte Textdatei(en) mit der Dateiendung `.java` ab. Geben Sie **keine** Projektdateien Ihrer Entwicklungsumgebung ab. Geben Sie **keinen** kompilierten Code ab (`.class`-Dateien). Geben Sie Ihren Code **nicht** als Archiv (z.B. als `.zip`-Datei) ab. Achten Sie darauf, dass Ihr Code kompiliert. Hausaufgaben, die sich nicht im vorgegebenen Format befinden, werden nur mit Punktabzug oder gar nicht bewertet.

Aufgabe 1.3 (H) Programmieren mit Herz und Pinguin [2 Punkte]

In der Tutorübung haben Sie gelernt, Ihre Entwicklungsumgebung zu benutzen, um ein einfaches *Hello World*-Programm zu erstellen. In dieser Aufgabe sollen Sie Ihr Programm wie folgt erweitern.

1. Erweitern Sie das Programm aus der Übung derart, dass es hinter *Hello World!* ein Herz (♥) ausgibt.
2. Finden Sie einen Pinguin im Unicode-Zeichensatz. Verstecken Sie den Pinguin zusätzlich in der Ausgabe Ihres Programms.

Hinweis: Sollten die Unicode-Symbole in Ihrem Programm nicht korrekt angezeigt werden, müssen Sie Ihren Editor höchstwahrscheinlich zunächst auf UTF-8 umstellen.

Lösungsvorschlag 1.3

1.

```
1 public class HelloWorld extends MiniJava {  
2     public static void main(String[] args) {  
3         write("Hello World! ♥");  
4     }  
5 }
```

2. Wie oben, nur mit Pinguin. Latex mag keine Pinguine (mögen Pinguine Latex?).

Korrekturhinweise

- Wie gut der Pinguin versteckt ist, spielt natürlich keine Rolle für die Bewertung.
- Die Angabe war ungenau formuliert in Bezug darauf, dass natürlich Herz / Pinguin in den Code sollten und nicht nur in die Ausgabe. Wer also die Symbole durch Escape-Sequenzen ausdrückt, bekommt (leider) auch alle Punkte; bitte aber im Kommentar darauf hinweisen, wie es gedacht war.