

Aufgabe 1.1 (P) Java JDK

Das Java Development Kit (*JDK*) stellt alle notwendigen Tools und Programme bereit, die die Entwicklung, das Debugging, Überwachen und Ausführen von Java Anwendungen ermöglicht. In dieser Aufgabe sollen Sie sich daher zunächst mit dem JDK vertraut machen.

- Laden Sie zuerst die aktuelle JDK Version¹ von Oracle herunter und installieren Sie diese auf Ihrer Maschine. Achten Sie dabei darauf, dass die JDK-Version kompatibel ist mit Ihrem Betriebssystem (Windows², Mac OS X, Linux) sowie dessen Architektur (32 oder 64 Bit).
- Zum Test Ihrer Java Installation führen Sie bitte folgende Schritte aus:
 - Mit dem Übungsblatt verteilen wir das Beispielpogramm `HelloWorld.java`. Speichern Sie dieses Programm auf Ihrem Computer.
 - Kompilieren Sie das Programm auf der Konsole mit dem Java Compiler. Der Befehl dazu lautet:

```
javac HelloWorld.java
```
 - Der Java Compiler übersetzt den Java-Quelltext `HelloWorld.java` (Dateiendung `.java`) in JVM Bytecode `HelloWorld.class` (Dateiendung `.class`). Die Datei mit dem Bytecode kann dann durch die Java-Laufzeitumgebung (JVM) mit folgendem Befehl ausgeführt werden:

```
java HelloWorld
```
 - **Achtung:** Der `java`- und `javac`-Befehl befindet sich im *bin*-Verzeichnis Ihrer Java-Installation.
- Machen Sie sich Gedanken zu folgenden Fragestellungen:
 - Worin liegt der wesentliche Unterschied zwischen `.java`- und `.class`-Dateien?
 - Weshalb benötigt man einen (Java)-Compiler?
 - Welche Vorteile ergeben sich dadurch?
 - Warum wird vom Java-Compiler nicht direkt Maschinen-Code erzeugt?

¹<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

²Hilfe zum Setzen der PATH-Variablen und Windows: <https://www.java.com/de/download/help/path.xml>

Aufgabe 1.2 (P) Eclipse

Eclipse ist eine integrierte Entwicklungsumgebung (IDE), welche durch die Bereitstellung eines besonderen Texteditors sowie Einbindung des Compilers und Debuggers die Entwicklung von Software-Anwendungen vereinfacht. Mittlerweile werden von Eclipse eine Vielzahl von verschiedenen Programmiersprachen unterstützt; eine davon ist Java. In dieser Aufgabe sollen Sie die grundlegenden Funktionen von Eclipse kennenlernen, sodass Sie in der Lage sind, Java-Anwendungen eigenständig in Eclipse zu schreiben, zu kompilieren und auszuführen.

- Installieren Sie sich dazu die aktuelle Version der Eclipse Standard Edition (*Eclipse SimRel 2018-09*³).
- Schreiben Sie eine Java-Anwendung in Eclipse, welche die Zeichenkette „Die PGdP-Veranstaltung ist eine Java-Klasse für sich.“ auf der Konsole ausgibt. Gehen Sie dabei wie folgt vor:
 - Legen Sie in Eclipse ein neues Projekt mit dem Titel *MeinPGdP* an.
 - Legen Sie im Projekt eine neue Klasse *MeinePGdPKlasse* an.
 - Implementieren Sie die Methode `public static void main(String[] args)` in der Klasse *MeinePGdPKlasse*.
 - Geben Sie die Zeichenkette in der *main*-Methode auf der Konsole aus. Nutzen Sie dazu die `System.out.println()`-Methode. Eine detaillierte Beschreibung wie diese Methode zu nutzen ist finden Sie in der Dokumentation zur Java-API⁴.
- Kompilieren und führen Sie Ihre Java Anwendung in Eclipse aus.

Hinweise: Zur Durchführung dieser Aufgabe ist es empfehlenswert, sich folgende frei verfügbare Tutorials anzuschauen:

- <http://www.vogella.com/articles/Eclipse/article.html>
- <http://www.del-net.com/download/Eclipse31Einfuehrung.pdf>

Die Hausaufgabenabgabe erfolgt über Moodle. Bitte geben Sie Ihren Code als UTF8-kodierte (ohne BOM) 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.

³<http://www.eclipse.org/downloads/>

⁴<https://docs.oracle.com/en/java/javase/11/>

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 Herzchen (♥) ausgibt.
2. Finden Sie einen Pinguin im Unicode-Zeichensatz. Verstecken Sie den Pinguin zusätzlich in der Ausgabe Ihres Programms.

Hinweis: Sowohl Herz als auch Pinguin müssen als einzelnes Symbol im Quelltext des Programms vorkommen. Sollten die Unicode-Symbole in Ihrem Programm nicht korrekt angezeigt werden, müssen Sie Ihren Editor höchstwahrscheinlich zunächst auf UTF-8 umstellen.