

Grundlagen der Programmierung

Marcel Lüthi
Andreas Morel-Forster
HS 23

Universität Basel
Fachbereich Informatik

Übung 7

Voraussetzung

- Ein JDK ist installiert.
- Installierte IDE, Visual Studio Code sowie die Plugins für Java
- Wenn Sie die Vorlesung verpasst haben, dann empfehlen wir Ihnen die Unterlagen anzuschauen.
- Die Zip-Datei, die auch dieses Übungsblatt enthält, muss entpackt werden. Es enthält die gesamte Übungsumgebung inklusive der automatisierten Tests. Schreiben Sie ihre Lösungen in die dafür vorgesehenen Dateien, wie in der jeweiligen Übungsaufgabe angegeben.

In der ersten Aufgabe werden Sie den Umgang mit dem Compiler sowie Kommandozeilenparameter üben.

Im zweiten Teil des Übungsblattes entwickeln wir unser eigenes Programm um Histogrammplots zu erstellen (Beispiele solcher Plots finden Sie am Ende dieses Übungsblatts). Dabei lernen Sie, wie Sie mit Klassen und Objekten zu arbeiten und wie Sie komplexe Funktionalität aus einfachen Teilen zusammensetzen können. In den ersten Teilaufgaben entwickeln Sie die einzelnen Teile. In der letzten Aufgabe werden Sie diese dann zusammensetzen.

Aufgabe 7.1 (Berechnung des BMI)

Schreiben Sie ein Java-Programm, das den BMI berechnet. Nutzen Sie dazu die bereits vorbereitete Datei `BMI.java`, die Sie im Verzeichnis `uebung007/src/main/java` finden.

Übergeben Sie dem Programm als ersten Parameter die Grösse (in cm) und als zweiten Parameter das Gewicht (in kg). Dabei soll ein Aufruf

```
java BMI 188 88
```

den BMI anhand der Formel

$$\frac{\text{Gewicht}[kg]}{\text{Grösse}[m]^2}$$

berechnen und die folgende Ausgabe erzeugen:

```
Ihr BMI betraegt: 24.898143956541425
```

Hinweis: Die Parameter der Kommandozeile werden dem Programm als String-Array mit Namen `args` in der `main`-Methode übergeben. Um aus den an ihr Programm übergebenen Parametern Zahlen vom Typ `Double` zu erhalten, können Sie folgenden Code benutzen:

```
double a = Double.parseDouble(args[0]);  
double b = Double.parseDouble(args[1]);
```

Im Terminal kompilieren Sie das Programm im Verzeichnis `uebung007/src/main/java` mit folgendem Befehl:

```
javac BMI.java
```

Im Terminal führen Sie das Programm im Verzeichnis `uebung007/src/main/java` mit folgendem Befehl aus:

```
java BMI 188 88
```

Aufgabe 7.2 (Die Klasse Punkt, 1 Punkt)

Im Verzeichnis `src/main/java/` finden Sie die Klasse `Point`, welche einen Punkt im Raum durch die Raumkoordinaten (x, y) repräsentiert. Implementieren Sie die fehlenden Teile dieser Klasse.

Tipp: Schreiben Sie eigene kleine Testprogramme in der `main`-Methode, um Ihre Implementation zu testen. Wenn alles funktioniert, nutzen Sie auch die mitgelieferten Tests. Diese können Sie wie in der Vorlesung gezeigt, direkt aus VSCode oder aus einem Terminal mittels des Kommandos `.\gradlew test` (`./gradlew test` auf MacOS) ausführen.

Aufgabe 7.3 (Grafische Elemente, 2 Punkte)

Im Verzeichnis `src/main/java` finden Sie die Klassen `Rectangle` und `PlotAxes`. Diese sollen es erlauben, durch das Angeben eines Referenzpunktes sowie der Länge und Höhe jeweils ein Rechteck respektive die Achsen eines Koordinatensystems zu zeichnen (siehe Bild unten).

Implementieren Sie die fehlenden Methoden und testen Sie ihr Programm. Nutzen Sie dazu die statische Methode `TurtleUtils.setTurtlePosition` um die Position vom Turtle zu setzen, welche in der Klasse `TurtleUtils` implementiert ist.

Beim Kompilieren der Programme müssen Sie die Turtle Bibliothek wie folgt mit angeben:

```
> javac -cp .;jturtle-0.7.jar Rectangle.java (Windows (Eingabeaufforderung))  
> javac -cp ".;jturtle-0.7.jar" Rectangle.java (Windows (Powershell))  
> javac -cp .:jturtle-0.7.jar Rectangle.java (Linux und MacOS)
```

Entsprechend ist das Kommand zum Aufruf nun

```
> java -cp .;jturtle-0.7.jar Rectangle (Windows (Eingabeaufforderung))
```

```
> java -cp ".;jturtle-0.7.jar" Rectangle (Windows (Powershell))
```

```
> java -cp .:jturtle-0.7.jar Rectangle (Linux und MacOS)
```



Koordinatenachsen



Rechteck

Aufgabe 7.4 (Histogram, 4 Punkte)

Im Verzeichnis `src/main/java/` finden Sie die Klasse `Histogram`. Der Zweck dieser Klasse ist es, ein Array von Werten entgegenzunehmen und zu zählen, wie viele der Werte in Klassen (genannt Bins) fallen. Die Klassen werden durch Intervalle fester Länge definiert.

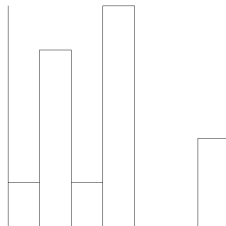
Implementieren Sie die fehlenden Methoden. Beginnen Sie dabei mit den Methoden `getMinValue` und `getMaxValue`, da Sie diese in den anderen Methoden brauchen werden. Die Kommentare im Code beschreiben genauer was die jeweiligen Methoden machen.

Tipp: Schreiben Sie eigene kleine Testprogramme in der Main-Methode um Ihre Implementation zu testen. Wenn alles funktioniert nutzen Sie auch die mitgelieferten Tests.

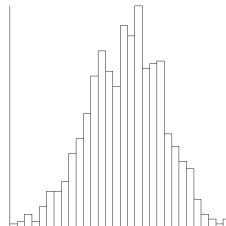
Aufgabe 7.5 (Plotten des Histogram, 3 Punkte)

Schlussendlich implementieren Sie die fehlenden Methoden in der Klasse `HistogramPlot`. Die Klasse sollte die Klassen `Histogram`, `Rectangle` und `PlotAxes` nutzen, um für ein gegebenes Array von Zahlen das Histogram zu plotten. Testen Sie ihr Programm. Der in der Main-Methode mitgegebene Testcode sollte Plots generieren, die ungefähr wie diese im Bild aussehen.

Tipp: Falls Sie die vorige Aufgabe nicht vollständig lösen konnten, können Sie zur Bearbeitung dieser Aufgabe die Klasse `MockHistogram` nutzen, welche Sie im selben Verzeichnis finden. Diese gibt einfach für jede Methode feste Werte zurück. Sie sollten damit das erste der beiden Histogramme reproduzieren können.



Einfacher Testfall



Komplexer Testfall

Aufgabe 7.6 (Farbige Grafiken, 0 Punkte)

Diese Aufgabe ist freiwillig. Können Sie das Programm so anpassen, dass die Balken der Histogramme farbig gezeichnet werden? Nutzen Sie dazu die Methode `fill` von `Turtle`.

Abgabe Erstellen Sie eine Zip-Datei der gesamten Übungsumgebung (also des Verzeichnisses `uebung7`) und laden Sie dieses auf Adam hoch.