Dozenten
Prof. Dr. Thomas Vetter
Dr. Marcel Lüthi
Departement
Mathematik und Informatik
Spiegelgasse 1
CH – 4051 Basel

Assistent
Dr. Marcel Lüthi
Departement
Mathematik und Informatik
Spiegelgasse 1
CH – 4051 Basel

Tutoren / Tutorinnen
Claudia Grundke
Viktor Gsteiger
Simon Dold
Timo Steinebrunner
Alexander Rovner
Nikodem Kernbach
Lukas Stöckli

Erweiterte Grundlagen der Programmierung (45398-01)

Blatt 5 [10 Punkte]

Vorbesprechung	21. Oktober. - 25. Oktober
Abgabe	1. November

Allgemeine Hinweise

- Wir empfehlen Ihnen, dass Sie im Buch "Sprechen Sie Java" bis und mit Kapitel 11 lesen.

Voraussetzung

- Es gelten dieselben Voraussetzungen wie für Übungsblatt 1. Wenn Sie sich betreffend der Umgebung oder der automatisierten Tests noch unsicher sind, lesen Sie bitte nochmals sorgfältig das Infoblatt oder Fragen Sie die Tutoren.
- 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.

Empfohlenes Vorgehen

- Wechseln Sie in den Ordner `src/main/java`. Dort finden Sie die Dateien, in welche Sie Ihren Java Code schreiben.
- Schreiben Sie ihr Programm, kompilieren Sie dieses mit dem Java Compiler `javac` und führen Sie es mit `java` aus, wie es in der Vorlesung gezeigt wurde.
- Wenn Sie denken, dass alles in Ordnung ist, wechseln Sie zurück ins Übungsverzeichnis `uebung5` und führen da `gradlew test` aus, um zu überprüfen ob Ihre Lösung die automatisierten Tests besteht. Überprüfen Sie auch Ihren Codestil mit `gradlew checkstyleMain`. Falls Ihr Code den Vorgaben entspricht, erhalten Sie einen Bonuspunkt.

Abgabe

Ergänzen Sie die Datei email.txt mit Ihrer Unibas E-Mail Adresse. Erstellen Sie eine Zip-Datei der gesamten Übungsumgebung (also des Verzeichnisses `uebung5`) und laden Sie dieses auf Courses (https://courses.cs.unibas.ch/) hoch.

In dieser Aufgabe 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.

Anmerkung: In dieser Aufgabe können nicht alle Aufgaben durch automatisierte Tests überpüft werden. Zum Testen der Aufgaben, die Grafiken produzieren, schauen Sie ob die Grafiken den Vorgaben entsprechen.

Aufgabe 1 - Punkt [1 Punkte]

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

`Tipp:` Schreiben Sie eigene kleine Testprogramme in der main Methode, um Ihre Implementation zu testen. Wenn alles funktioniert, nutzten Sie auch die mitgelieferten Tests. Mit `gradlew test --tests PointTests` können Sie nur die Tests für diese Klasse ausführen.

Aufgabe 2 - Grafische Elemente [2 Punkte]

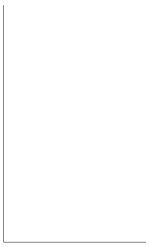
Im Verzeichnis `src/main/java` finden Sie die Klassen `Rectangle` und `PlotAxes`. Diese sollen es erlauben, durch das Angeben eines Referenzpunktes sowie der Länge und Höhe jeweils ein Rechteck respektive die Achsen eines Koordinatensystems zu zeichnen (siehe Bild unten).

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

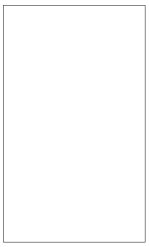
Beim Kompilieren der Programme müssen Sie die Turtle Bibliothek wie folgt mit angeben:

> `javac -cp .;jturtle-0.5.jar Rectangle.java` (Windows)

> `javac -cp .:jturtle-0.5.jar Rectangle.java` (Linux und MacOS)



Koordinatenachsen



Rechteck

Aufgabe 3 - 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.

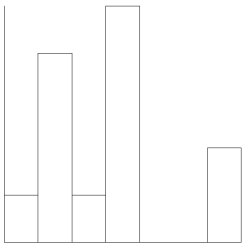
Implementieren Sie die fehlenden Methoden. Beginnen Sie dabei mit den Methoden `getMinValue` und `getMaxValue`, da Sie diese in den anderen Methoden brauchen werden. Die Kommentare im Code beschreiben genauer was die jeweiligen Methoden machen.

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

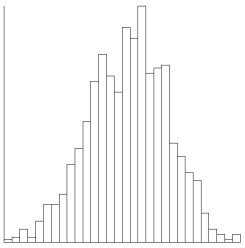
Aufgabe 4 - Plotten des Histogram [3 Punkte]

Schlussendlich implementieren Sie die fehlenden Methoden in der Klasse `HistogramPlot`. Die Klasse solle 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 5 - 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.