

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

Hinweise zu den Tests

Erweiterte Grundlagen der Programmierung (45398-01)

Für die Übungen der Veranstaltung *Erweiterte Grundlagen der Programmierung* werden automatische Testprogramme mitgeliefert. Hier finden Sie eine Anleitung, wie diese Tests verwendet werden können.

Am besten Lesen Sie sich dieses Blatt gründlich durch, bevor Sie das erste Übungsblatt lösen, aber nachdem Sie dieses heruntergeladen und die Zip-Datei entpackt haben.

Automatisierte Tests

Ein Test hilft Ihnen dabei festzustellen, ob ihr Programm richtig funktioniert oder ob es noch versteckte Fehler enthält, die Sie möglicherweise noch nicht entdeckt haben. Ein Test besteht aus mehreren kleinen Prüfungen die Teile der geschriebenen Programme oder Methoden auf ihre Richtigkeit prüfen. Ein Test prüft aber nicht alle möglichen Eventualitäten und dient deshalb nur als Anhaltspunkt ob eine Übungsaufgabe richtig gelöst wurde.

Um die Tests automatisiert ausführen zu können, verwenden wir die Software *Gradle*. *Gradle* ist ein sogenanntes Buildtool, welches das Bauen von Programmen (also das Kompilieren) sowie das Ausführen der Tests koordiniert.¹

Um Ihre Programme zu testen, starten Sie eine Konsole und wechseln in das Verzeichnis mit den von Ihnen heruntergeladenen Übungen. Dort finden Sie die Datei `gradlew`, mit der Sie *Gradle* starten können. Führen Sie folgendes Kommando aus:

```
> .\gradlew test (unter Windows)
> ./gradlew test (unter MacOS und Linux)
```

Dies testet jeweils alle Übungsaufgaben eines Übungsblattes und listet anschliessend auf, welche Prüfungen bestanden wurden und welche nicht. Für das erste Übungsblatt könnte die Ausgabe zum Beispiel wie folgt aussehen:

```
BMITests > testAufgabe2BerechnungRichtig PASSED
BMITests > testAufgabe2AusgabeRichtig PASSED
SquaresTest > testOutputImageTheSame FAILED
```

¹Beachten Sie, dass Gradle nicht benötigt wird um Java zu programmieren. Es vereinfacht nur die Ausführung von wiederkehrenden Kommandos. Im Hintergrund werden aber einfach wieder die Standardkommandos `javac` und `java` aufgerufen. Für diese Vorlesung müssen Sie Gradle nicht verstehen, sondern lediglich die hier vorgegebenen Kommandos ausführen können.

```
java.lang.AssertionError at SquaresTest.java:43
WelcomeTests > testAufgabe1EndetMitAusrufezeichen PASSED
WelcomeTests > testAufgabe1EnthaeltHalloUndName PASSED
WelcomeTests > testAufgabe1AufrufOk STANDARD_OUT
    Hallo Spock!
WelcomeTests > testAufgabe1AufrufOk PASSED

6 tests completed, 1 failed
```

Diese Ausgabe zeigt an, dass der Test `SquaresTest` fehlgeschlagen ist und somit Ihre Lösung für die entsprechende Teilaufgabe nicht richtig sein kann. Die Tests für alle anderen Aufgaben waren jedoch erfolgreich.

Wenn Sie interessiert sind, wie ihr Code getestet wird, können Sie sich alle Testfälle anschauen. Sie finden den Code für die Testfälle im Verzeichnis `src/test/java`. Es handelt sich dabei um normalen Java Code.

Checkstyle

Neben Korrektheit vom Code prüfen wir auch, ob die Abgabe unseren Vorgaben betreffend Programmierstil entspricht. Unsere Anforderungen werden dabei mit jeder Übung etwas grösser. Um zu überprüfen ob Ihr Programmierstil unseren Anforderungen entspricht verwenden wir das Werkzeug *Checkstyle*, welches wiederum durch *Gradle* aufgerufen wird. Um Ihr Programm auf Stilfehler zu überprüfen, geben Sie folgendes ein:

```
.\gradlew checkstyleMain
```

Falls Checkstyle Fehler findet, werden diese auf der Konsole angezeigt. Zusätzlich wird ein Bericht generiert, den Sie im Verzeichnis `build/reports/checkstyle` finden. Die Vorgaben, wie richtig formatierter Java code aussehen soll finden Sie hier:

https://checkstyle.org/sun_style.html.