

Arbeiten mit JUnit 3.8

– mit der Windows Eingabeaufforderung bzw. der Shell auf Linux/MacOS –

Achtung! Blöcke der Form

```
$ irgendein_kommando  
Ausgabe des Kommandos
```

bedeuten, dass man `irgendein_kommando` auf der Eingabeaufforderung bzw. Shell eingeben muss (ohne `$`), mit `Enter` abschickt und daraufhin eine Ausgabe ähnlich der gegebenen erhalten sollte.

Voraussetzungen

- OpenJDK installiert (Version 8, 11 oder 12)
- JUnit 3.8.2 ZIP heruntergeladen und entpackt

JUnit 3.8

Ist das JUnit-Archiv entpackt, erhalten wir folgende Dateien/Verzeichnisse:

- README.html
- cpl-v10.html
- doc/
- javadoc/
- junit/
- **junit.jar**
- src.jar

junit.jar ist für uns wichtig – darin befinden sich alle kompilierten Klassen des JUnit-Frameworks. Wir notieren uns den **kompletten Pfad** zu dieser Datei für später. Typische Orte, je nach Betriebssystem, könnten sein:

- Windows: `C:\Users\JavaJoe\Downloads\junit3.8.2\junit.jar`
- Linux: `/home/javajoe/Downloads/junit3.8.2/junit.jar`
- MacOS: `/Users/JavaJoe/Downloads/junit3.8.2/junit.jar`

Den ersten Test kompilieren

Wir legen eine Datei `FirstTest.java` an, um zu schauen wie wir unsere Installation zum Laufen bekommen:

```
import junit.framework.TestCase;

public class FirstTest extends TestCase {
    public void testMathOperation() {
        assertEquals(1 + 1, 2);
    }
}
```

(Der Inhalt wird in der Übung klar werden.)

Der erste Versuch der Kompilierung schlägt fehl:

```
$ javac FirstTest.java
FirstTest.java:2: error: package junit.framework does not exist
import junit.framework.TestCase;
                   ^
[...]
```

D.h., der Java-Compiler kann die Klassen aus dem JUnit-Framework nicht finden. Wir können dem Compiler aber den Pfad zu dieser *Klassenbibliothek* mitteilen, indem wir den *Classpath* mit der Option `-cp` anpassen. Der Classpath ist eine Liste von Orten, an denen nach solchen Bibliotheken gesucht werden soll:

Linux und MacOS:

```
$ javac -cp ./pfad/zur/junit.jar FirstTest.java
```

Windows:

```
$ javac -cp .;C:\pfad\zur\junit.jar FirstTest.java
```

Jetzt funktioniert es, und es sollte die Datei `FirstTest.class` im Verzeichnis aufgetaucht sein.

HINWEIS: In beiden Fällen wurde `.` (= das Arbeitsverzeichnis) an den Anfang des Classpath gesetzt. Das Verzeichnis `.` ist der Standardwert für die `-cp` Option und sollte nicht vergessen werden wenn diese explizit gesetzt wird.

INFO: Das Trennzeichen für die Listeneinträge ist plattformabhängig, unter Windows ist es `;` und unter Linux/MacOS `:`.

Den ersten Test ausführen

JUnit 3.8 bringt mehrere sogenannte *Runner* mit: Klassen mit `main()` -Methode, die wir direkt

dem `java` -Kommando als auszuführende Klassen mitgeben können. Zum Beispiel:

```
$ java -cp .:pfad/zur/junit.jar junit.textui.TestRunner FirstTest
.  
Time: 0,001  
  
OK (1 test)
```

Als Argument haben wir dem `TestRunner` den Namen unserer Testklasse, `FirstTest`, mitgegeben.

Neben `junit.textui.TestRunner` gibt es auch einen Runner mit graphischer Oberfläche: `junit.swingui.TestRunner`. Dessen Aufruf funktioniert analog zu obigem Aufruf.

In der Übung wird mit den `TestSuite`s noch eine weitere Variante vorgestellt, wie Tests ausgeführt werden können.