

Übung 7: Kopplung und Kohäsion



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Software Engineering
WS 2018/19 - Dr. Michael Eichberg

Abgabe

Die Übung wird als *sbt*-Projekt (<http://www.scala-sbt.org>) bereitgestellt. *sbt* kann verwendet werden, um Java (und Scala) Projekte einfach auszuführen und zu testen. Installieren Sie *sbt* auf Ihrem Rechner und stellen Sie sicher, dass ein Java SDK (min. Java 8) installiert ist. Überprüfen Sie, dass der Java Compiler *javac* auf der Kommandozeile ausführbar ist, auch wenn Sie sich nicht im Verzeichnis mit der ausführbaren Datei befinden. Wenn nicht, passen Sie Ihren *PATH* entsprechend an, Hinweise wie dies bei Ihrem Betriebssystem möglich ist, finden Sie im Internet.

IntelliJ erlaubt das Importieren von *sbt*-Projekten direkt, dafür muss allerdings das Scala Plugin installiert sein. Um ein Eclipse-Projekt zu erzeugen, kann *sbt eclipse* im Projektverzeichnis (das Verzeichnis, das die Datei *build.sbt* enthält) ausgeführt werden, danach kann das Projekt mit Datei > Importieren > Vorhandene Projekte in Arbeitsbereich importiert werden. Wenn Sie eine *main*-Methode geschrieben haben, können Sie Ihr Programm mit *sbt run* ausführen, Tests können Sie unter *src/test/java/* anlegen und mit *sbt test* ausführen.

Das Ausführen des Kommandos *sbt* im Projektverzeichnis startet den interaktiven Modus von *sbt*. Im interaktiven Modus können Kommandos ausgeführt werden, ohne jedes Mal *sbt* eingeben zu müssen.

Um Ihre Lösung abzugeben, melden Sie sich zunächst unter

<https://submission.st.informatik.tu-darmstadt.de/course/se18>

an und erzeugen Sie ein *submission token*. Wenn Sie den interaktiven Modus von *sbt* verwenden, führen Sie dann folgendes Kommando aus:

```
submit <ihreTUID> <submissionToken>
```

wobei *<ihreTUID>* Ihre eindeutige Identifikationsnummer an der TU Darmstadt (**nicht Ihre Matrikelnummer!**) und *<submissionToken>* das zuvor generierte Token ist. Geben Sie die spitzen Klammern nicht mit an, der Befehl sollte etwa so aussehen: *submit ab01cdef 01234567*. Wenn Sie nicht den interaktiven Modus verwenden, muss das Kommando in Anführungszeichen gesetzt werden, also *sbt "submit <ihreTUID> <submissionToken>"*. Sie können (innerhalb der Abgabefrist) beliebig oft eine Lösung einreichen, allerdings wird nur die zuletzt eingereichte Lösung bewertet. Die letzte Lösung, die ein Gruppenmitglied eingereicht hat, wird zur Bewertung für die ganze Gruppe herangezogen. Koordinieren Sie sich daher in Ihrer Gruppe, wer Ihre gemeinsame Lösung einreicht.

Stellen Sie sicher, dass Sie das zur Verfügung gestellte Template nutzen und die Namen und Signaturen der vorgegeben Klassen und Methoden nicht verändern sowie dass von Ihnen hinzugefügte Klassen und Methoden die geforderten Namen und Signaturen verwenden. Ändern Sie außerdem nichts an der vorgegebenen Datei *build.sbt*. Andernfalls wird das System Ihre Lösung nicht bewerten können. Beachten Sie, dass der Zugriff auf die Abgabepattform nur im **internen Netz der Universität** möglich ist. Für einen Zugriff von außerhalb benötigen Sie daher eine VPN-Verbindung. Überprüfen Sie, ob Ihre Abgabe erfolgreich war, indem Sie sie von der Abgabepattform herunterladen. Überprüfen Sie dabei auch, ob alle Dateien in der Abgabe enthalten sind.

Einführung

In dieser Übung befassen Sie sich mit Kopplung und Kohäsion. Bearbeiten Sie die folgenden drei Aufgaben und erstellen Sie eine einzelne PDF-Datei mit Ihren Antworten. Legen Sie diese Datei in den Ordner *solution* in dem Template, das Ihnen zur Verfügung gestellt wurde, und verwenden Sie zur Abgabe wie oben angegeben *sbt submit*.

Problem 1 Kopplung**6P**

Im Template zu dieser Übung finden Sie ein kleines Projekt aus mehreren Klassen. Betrachten Sie die Klasse `Navigation`:

- Bestimmen Sie die Kopplung der Klasse (d.h. alle Klassen zu denen eine Abhängigkeit besteht)
- Bewerten Sie die Kopplung der Klasse im Hinblick auf Wiederverwendbarkeit in einem anderen Projekt.

Problem 2 Kohäsionsmetriken**5P**

Betrachten Sie wieder die Klasse `Navigation`. Bestimmen Sie den Mangel an Kohäsion nach folgenden Metriken, wobei der Konstruktor *nicht* beachtet werden soll.

- Die LCOM-Metrik nach Li und Henry
- Die LCOM*-Metrik nach Henderson-Sellers (berücksichtigen Sie hierfür nur Instanzattribute)

Geben Sie jeweils nachvollziehbar an, wie Sie zu Ihrem Ergebnis gekommen sind (z.B. Rechenweg).

Was ist, im Allgemeinen, der mögliche Wertebereich der beiden Metriken? Welchen Einfluss hat das auf die Vergleichbarkeit zwischen verschiedenen Klassen?

Problem 3 Kohäsion**5P**

Bewerten Sie nun, auch unter Berücksichtigung ihrer Ergebnisse aus Problem 2, die Kohäsion der Klasse `Navigation`. Welche Konsequenzen ergeben sich daraus? Welche Aussagekraft hat die LCOM-Metrik nach Li und Henry für die Methoden `toString` und `printRoute`? Was ist der Grund dafür? Nehmen Sie geeignete Anpassungen am Projekt aus dem Template vor, um bessere Kohäsion zu erreichen.