



Praktikum

Formale Entwicklung objektorientierter Software

Übungsblatt 10

Aufgabe 1

Verifizieren Sie mit KeY in Ihrer Implementierung von `ExamDataBase` die Einhaltung der von der abstrakten Oberklasse geerbten Methodenspezifikationen für die Methoden `addStudent` und `getNumWithGrade`. Beweisen Sie jeweils sowohl den `normal_behavior`- als auch den `exceptional_behavior`-Kontrakt. Die Einhaltung der `assignable`-Klauseln müssen Sie nicht beweisen.

Konstrukturen müssen dafür sorgen, dass das von ihnen initialisierte Objekt seine Invarianten erfüllt. Überprüfen Sie mit KeY, ob das für jeden Konstruktor der `ExamDataBase` implementierenden Klasse der Fall ist (dazu `use all applicable invariants` anschalten und die Beweisverpflichtung `Class specification for class ...` auswählen).

Behandeln Sie in den zu verifizierenden Methoden und Konstruktoren vorkommende Methodenaufrufe grundsätzlich durch Verwendung der Kontrakte der aufgerufenen Methode (Regel `Use Method Contract`; Vorgehen wie in Blatt 7, Aufgabe 2 beschrieben). Dies gilt besonders für Aufrufe zu selbstdefinierten Hilfsmethoden. Verifizieren Sie auch für die aufgerufenen selbstdefinierten Hilfsmethoden mit KeY, dass sie ihre Kontrakte erfüllen.

Hinweise:

- Stellen Sie vor dem Beweisen sicher, dass in KeY unter “Options” - “Taclet Libraries” die Bibliothek “stringRules.key” angeschaltet ist.
- Es bietet sich an, die Beweise für die selbstdefinierten Hilfsmethoden als erstes durchzuführen. Achten Sie darauf, dass die von Ihnen geschriebenen Kontrakte für diese Methoden aussagekräftig genug sind, sonst bekommen Sie später Probleme in den Beweisen, in denen diese Kontrakte verwendet werden. Vergessen Sie insbesondere nicht, geeignete `assignable`-Klauseln anzugeben.
- Die Schwierigkeit besteht sonst im Wesentlichen darin, für die vorkommenden Schleifen geeignete Invarianten, `assignable`-Klauseln und Varianten zu finden. Diese sind meist ähnlich aufgebaut wie in Blatt 7, Aufgabe 1.
- In der Schleifeninvariante der Methode `getNumWithGrade` wird es notwendig sein, vom JML-Konstrukt `\num_of` Gebrauch zu machen. Formulieren Sie `\num_of`-Ausdrücke in der Form `(\num_of int i; a<=i && i<=b; p)`, d.h. so, dass der Laufbereich der Variablen als eine Konjunktion von zwei `<=` Ausdrücken beschrieben wird. Das wird Ihnen das Beweisen deutlich erleichtern, da die Behandlung der aus so geformten `\num_of`-Ausdrücken resultierenden Summenterme in KeY besonders gut automatisiert ist.
- Stellen Sie für die Beweise zu `getNumWithGrade` die Strategioption “Query treatment” auf “expand”. Für die anderen Beweise genügt “none”. Die Option “Quantifier treatment” sollte immer auf “No Splits with Progs” stehen.

- In den Beweisen kommen Aufrufe zu “impliziten” Methoden vor. Diese sind an ihren in spitze Klammern eingeschlossenen Namen (z.B. `<init>`) zu erkennen und werden von KeY benutzt, um die Java-internen Vorgänge beim Erzeugen von Objekten zu modellieren. Behandeln Sie diese Aufrufe mit der Regel `methodBodyExpand`.
- Die Korrektheit Ihrer Implementierungen vorausgesetzt, sollten Ihre Beweise abgesehen vom manuellen Behandeln der Methodenaufrufe automatisch geschlossen werden können.

Abgabe bis 29.02.

Es braucht pro Gruppe nur *eine* Lösung abgegeben werden.
 Die Abgabe der Übungsblätter erfolgt mit dem SVN System. Dazu legen Sie die abzugebenden Dateien im SVN ab und kopieren sie mit SVN in den Unterordner *abgabe/*`<nr>` wie in Aufgabe 2 auf Blatt 1 beschrieben.
 Einige Aufgaben verlangen eine schriftliche Bearbeitung, diese ist dann je nach Komplexität als ASCII, html, ps- oder pdf-Dokument abzugeben. Auf *keinen* Fall im MS Word `doc`-Format.

Praktikums-Webseite: <http://i12www.ira.uka.de/~engelc/lehre/keypraktWS0708/>

Dr. Thomas Käußl: Zi. 207, Tel. 608-6286, E-Mail: kaeufl@ira.uka.de
Christian Engel: Zi. 106, Tel. 608-4338, E-Mail: engelc@ira.uka.de
Benjamin Weiß: Zi. 309, Tel. 608-4324, E-Mail: bweiss@ira.uka.de