Universität Karlsruhe
Institut für Theoretische Informatik
Prof. Dr. P. H. Schmitt
C. Engel, B. Weiß

04. Februar 2009

Praktikum
Formale Entwicklung objektorientierter Software
Übungsblatt 10 – Gruppe 2

Vorbemerkung

Sie haben eine Version Ihres Milliardenbankprojekts zugesandt bekommen oder werden es noch bekommen. Im Wesentlichen haben wir die GUI-Klassen entfernt, `System.out.println` Anweisungen auskommentiert, und evtl. kleine Syntaxfehler in der Spezifikation behoben. Insbesondere haben wir keine Fehler in die Implementierung eingebaut, die nicht schon vorher vorhanden waren.

Außerdem haben wir einzelne Methodenspezifikationen verändert oder eingefügt. Es kann sein, dass eine solche Spezifikation die Aufgabe der Methode nicht richtig beschreibt. Sollten Sie feststellen, dass ein Beweis nicht schließbar ist, kann der "Fehler" in der Spezifikation, der Implementierung oder in beidem liegen. Die Ursache sollte dann beseitigt und die Verifikation erfolgreich abgeschlossen werden. Nähere Hinweise finden Sie in der gruppenspezifischen Aufgabenbeschreibung unten.

Schalten Sie bitte vor dem Beweisen im KeY-System unter "Taclet libraries" den Punkt "stringRules.key" an.

Aufgabe 19 (Gruppe 2)

(a) Methode `util.MyDate::getToday()`. Beweisen Sie für Ihre beiden Verträge jeweils die "EnsuresPost"-Beweisverpflichtung (mit den Standardeinstellungen für die "assumed invariants"). Vergleichen Sie die beiden Beweise. Was fällt Ihnen auf? Woran liegt das? Hinweis: Achten Sie auf die verwendeten Klasseninvarianten!

(b) Methode `server.Zentralrechner::getKonto(Person)`. Beweisen Sie "EnsuresPost" für die beiden (von uns leicht veränderten) Verträge.

- Einer der Verträge widerspricht sich mit einem anderen (implizit vorhandenen) Spezifikationselement. Beseitigen Sie diesen Widerspruch. Falls Sie das Problem nicht gleich sehen, können Sie versuchen, den Vertrag zu verifizieren, und sich den offenen Beweis anschauen.
- Wie üblich müssen Sie für die Behandlung der Schleife zunächst eine passende Schleifeninvariante, -variante und `assignable`-Klausel angeben.
- Die Korrektheit der Methode hängt von Klasseninvarianten anderer Klassen ab. Identifizieren Sie diese Klassen und aktivieren Sie die nötigen Invarianten beim Starten der Beweise im "Contract Configurator" unter "assumed invariants", um sie für die Verifikation verwenden zu können.
- Benutzen Sie für die Beweissuchstrategie die Einstellungen "Method treatment: expand" und "Query treatment: Expand".

(c) Methode `modell.Konto.geldAbheben(Bargeld)`. Beweisen Sie "EnsuresPost" für den markierten Vertrag. Wählen Sie dazu unter "assumed invariants" die nötigen Klasseninvarianten und verwenden Sie dann die Strategieeinstellungen "Method treatment: Contracts" und "Query treatment: None".

(d) (*Ohne Abgabe*) Betrachten Sie noch einmal Ihre Implementierung der zentralen Methode `server.Zentralrechner::geldAbheben`. Erfüllt die Implementierung tatsächlich die angegebenen Verträge?

Geben Sie sowohl die geänderten .java-Dateien als auch die Beweise selbst ab.
Rückfragen zu diesen Aufgaben bitte an Benjamin Weiß.

Abgabe bis 31.03.
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://lfm.iti.uni-karlsruhe.de/keyprakt0809.php`
Christian Engel: R. 106, Tel. 608-4338, E-Mail: `engelc@ira.uka.de`
Benjamin Weiß: R. 309, Tel. 608-4324, E-Mail: `bweiss@ira.uka.de`