

Praktikum

Formale Entwicklung objektorientierter Software

Übungsblatt 3: ESC/Java2

Aufgabe 12 — Einführung in ESC/Java2

ESC/Java2 ist ein Werkzeug, das die Korrektheit von Java-Programmen bzgl. einer JML-Spezifikation überprüfen kann. Im Unterschied zum Testen mit RAC oder JMLUnit führt ESC/Java2 eine *statische* Verifikation durch, d.h., der Quellcode wird untersucht, ohne das Programm tatsächlich auszuführen. Vorteile dieses Ansatzes sind vor allem, dass keine Testfälle geschrieben werden müssen, und dass die Prüfung *alle* möglichen Eingabewerte und alle Ausführungspfade durch das Programm berücksichtigen kann, nicht nur diejenigen, die durch Testfälle abgedeckt sind.

Lesen Sie sich die auf der Webseite verfügbaren Informationen zu ESC/Java2 durch. Nehmen Sie dabei insbesondere folgende Punkte zur Kenntnis:

- *Modularität* der Verifikation: ESC/Java2 prüft jede Methode einzeln. Ruft eine Methode eine andere auf, wird zur Behandlung dieses Methodenaufrufs nur die *Spezifikation* der aufgerufenen Methode benutzt, niemals ihre Implementierung. Es ist Aufgabe des Benutzers, dafür zu sorgen, dass diese Spezifikation aussagekräftig genug ist, um die Korrektheit der aufrufenden Methode daraus abzuleiten.
- Fehlende *Vollständigkeit*: Es kommt vor, dass ein eigentlich korrektes Programm von ESC/Java2 als fehlerhaft beanstandet wird, weil das Werkzeug nicht in der Lage ist, die Korrektheit zu beweisen. (Da das Problem der Programmverifikation unentscheidbar ist, muss es solche Fälle schon aus theoretischer Sicht geben.)
- Fehlende *Korrektheit*: Um die Benutzbarkeit möglichst einfach zu halten, macht ESC/Java2 auch bei der Korrektheit Kompromisse. Es gibt also fehlerhafte Programme, die das Werkzeug fälschlich als korrekt klassifiziert.
- Voreinstellung für `non_null/nullable`: In JML ist seit einiger Zeit “`non_null`” die Voreinstellung, aber ESC/Java2 hat diesen Wandel nicht ganz nachvollzogen. In der von uns verwendeten Version 2.0.5 mit Option `-nonNullByDefault` scheint `non_null` zwar als Voreinstellung für Methodenparameter und -rückgabewerte verwendet zu werden, nicht aber für Felder. Geben Sie im Zweifelsfall alle `non_null/nullable`-Annotationen explizit an.

Sie können ESC/Java2 mit einem Befehl wie “`escj MyClass.java`” auf eine Klasse anwenden. Mit Hilfe der Option `-routine` können Sie auch nur einzelne Methoden prüfen.

Aufgabe 13 — Bag und Amount

Laden Sie von der Praktikums-Webseite die Dateien `Bag.java` und `Amount.java` herunter und lösen Sie die in den Kommentaren beschriebenen Aufgaben. Ziel ist es, den Quellcode mit JML-Spezifikationen zu annotieren und eventuell vorhandene Bugs zu beseitigen, bis ESC/Java2 schließlich keine Warnungen mehr ausgibt. Bei `Amount.java` sind vorher noch einige in natürlicher Sprache gegebene Eigenschaften als JML-Klasseninvarianten zu formalisieren.

In beiden Fällen soll die fertige Spezifikation keine `assume`-Anweisungen enthalten. Verwenden Sie auch keine Optionen wie `NoWarn`, mit denen Warnungen unterdrückt werden.

Zum Schluss: Glauben Sie, dass Sie *alle* Probleme gefunden haben? Wie sicher sind Sie sich, dass der Code jetzt korrekt ist? Haben Sie Ideen, wie JML oder ESC/Java2 verbessert werden könnten?

Aufgabe 14 — Klausuren

Wenden Sie ESC/Java2 auf Ihre Lösung zu Aufgabe 3 von Blatt 1 an.

- (a) Findet das Werkzeug den Fehler, den Ihre `main`-Methode aufzeigt? Wenn ja: Bei der Analyse welcher Methode wird der Fehler erkannt, und warum dort?
- (b) Findet das Werkzeug noch weitere Fehler in dem Programm? Welche?

Technischer Hinweis

Das Program ESC/Java 2.0.5 können Sie herunterladen und selbst installieren (URL steht auf der Homepage zum Praktikum). Sie finden es aber auch auf unseren Poolrechnern und können es starten, in dem Sie dort den Befehl `escj` verwenden. *Vorsicht*: ESC/Java ist mit manchen neueren Java-Installationen (v.a. Java 1.7) nicht verträglich.

Abgabe bis Dienstag, 27.11.

Abgabe (als Java-, ASCII- oder PDF-Dateien, oder tar-Archiv solcher) per E-Mail an Matthias Ulbrich. Es braucht pro Gruppe und Aufgabe nur *eine* Lösung abgegeben werden. Bitte dokumentieren Sie Ihre Lösungen ausreichend und seien Sie darauf vorbereitet, sie auf Nachfrage zu erklären.

Praktikums-Webseite: <http://formal.iti.kit.edu/teaching/keypraktWS1213/>

<i>Daniel Bruns:</i>	R. 223, Tel. 608-45268, E-Mail: bruns@kit.edu
<i>David Faragó:</i>	R. 308, Tel. 608-47322, E-Mail: farago@ira.uka.de
<i>Christoph Gladisch:</i>	R. 223, Tel. 608-45268, E-Mail: gladisch@ira.uka.de
<i>Christoph Scheben:</i>	R. 106, Tel. 608-44338, E-Mail: scheben@ira.uka.de
<i>Matthias Ulbrich:</i>	R. 106, Tel. 608-44338, E-Mail: mulbrich@ira.uka.de