JaWA: Java with Assertions

Clemens Fischer und Dieter Meemken
Universität Oldenburg

http://theoretica.informatik.uni-oldenburg.de/~jawa

Hintergrund

Korrektheit von Programmen durch formale Spezifikation verbessern.

- Spezifikationssprache: prädikatenlogische Formeln über dem Zustandsraum
- Zusammenhang zwischen Spezifikation und Implementierung:
  - formale Verifikation
  - Testen

Programmieren mit Vertrag (Meyer)

Grundidee

- JaWA Programm: Java Programm mit zusätzlichen Zusicherungen in Form von Kommentaren.
- JaWA Präcompiler: Übersetzt ein JaWA Programm in Java. Aus Zusicherungen werden im Prinzip `if`-Abfragen.

Vorteile:

- JaWA Programm sind auch ohne Übersetzung ausführbar.
- Zusicherungen können mit `javadoc` verarbeitet werden.
- JaWA und Java können vermischt werden.
- Geringe Einarbeitungszeit

Klasseninvarianten

```
public class Stack {
  private Linkable head;        // Kopf der Liste
  private int      items;       // Anzahl der Elemente
  ...

  /** invariant items>=0; items<=max **/
}
```

Überprüfung: zu Beginn und am Ende von jedem Methodenaufruf.

Vor- und Nachbedingungen

```
public boolean IsEmpty() {
     if( items==0 ) return true;
     else return false;
    /** ensure result==(items==0); nochange **/
  }
public void Push(Object v) {
    /** require !IsFull(); v!=null **/
     ...
   /** ensure !IsEmpty(); items==old_items+1; v==Top() **/
  }
```

Check-Anweisung

```
/* check items <= max - 2 */
```

Schleifeninvarianten und -varianten

```
Linkable actual = head;
while (i<items) {
/* invariant items==old_items */
/* variant items-i */
  actual = actual.GetNext(); i++;
}
```

Vererbung

- Invarianten der Vater-Klasse werden überprüft.
- Ein Zusammenhang zwischen alten und neuen Vor- und Nachbedingungen wird nicht hergestellt.

Vertragsverletzung: Rescue

```
public void Div() {
  /** require z!=0 **/
  x =y/z;

  /* rescue catch (RuntimeCheck.AssertionException e)

     { x=1; }
  */
}
```

Vertragsverletzung: Retry

```
public void Div() {
   /* check z!=0 */
   x =y/z;
   /* rescue catch (RuntimeCheck.AssertionException e)
  { z=1; retry; }
   */
}
```

JaWA Präcompiler

Überprüfungsmodi: Was wird überprüft?

- All: alle Zusicherungen
- Preconditions: nur Vorbedingungen ($\rightarrow$ Integrationstests)
- Nothing: keine Überprüfung

Übersetzungsmodi: Wie wird auf Verletzungen reagiert?

- Contract: geprüften Ausnahme
- UncheckedContract: ungeprüfte Ausnahme
- Warning: keine Ausnahem

Fehlermeldungen: *ExceptionClass* und *WarningClass*.

Andere JaWA-Tools

- AssertMate: sehr kleiner Sprachumfang
- JContracts: kein Präprocessor (kein old)
- JavaSpec: keine Invarianten
- JPP: keine Invarianten

Ausblick

- Reimplementierung von JaWA in Java
- Mächtigkeit der Spezifikationssprache: Quantoren
- Vererbung: Überprüfung von Vor- und Nachbedingungen
- Overhead minimieren durch bessere statische Analyse

$\Rightarrow$ Parallelität