Übungen zur Vorlesung

Softwaretechnologie

-Wintersemester 2010/2011-

Dr. Günter Kniesel

Übungsblatt 13

Zu bearbeiten bis: 30.01.2011

Bitte fangen Sie **frühzeitig** mit der Bearbeitung an, damit wir Ihnen bei Bedarf helfen können.
Checken Sie die Lösungen zu den Aufgaben in Ihr SVN-Repository ein, Texte als Textdatei.

Aufgabe 1. *Whitebox Test* (10 Punkte)

Gegeben sei die folgende Methode `sortiere`, welche mittels Bubblesort ein Feld von Variablen des Typs `int` sortiert.

```
public int[] sortiere(int[] bestand) {                    // Anweisungsnr.
      boolean change = true;                              //  1
      if (bestand.length > 1) {                           //  2
            while (change) {                              //  3
                  change = false;                         //  4
                  for (int i = bestand.length - 1;        //  5
                              i > 0;                      //  6
                              i--) {                      //  7
                        int i1 = bestand[i];              //  8
                        int i2 = bestand[i - 1];          //  9
                        if (i1 < i2) {                    // 10
                              bestand[i] = i2;            // 11
                              bestand[i - 1] = i1;        // 12
                              change = true;              // 13
                        }
                  }
            }
      }
      return bestand;                                     // 14
}
```

a) Entwerfen Sie für die Methode `sortiere` einen Kontrollflussgraphen.

b) Geben Sie ein Feld mit Eingabewerten an, das nötig ist, um eine Anweisungsüberdeckung zu erreichen. Schreiben Sie die Reihenfolge auf, in der die Anweisungen getestet werden.

c) Erreichen Sie mit diesem Feld an Eingabewerten auch eine Verzweigungsabdeckung? Begründen Sie kurz Ihre Antwort. Falls ja, geben Sie eine Codemodifikation an, mit der die Verzweigungsabdeckung nicht mehr gegeben wäre. Falls nicht, geben Sie ein weiteres Eingabewerte-Feld an, sodass auch die übrigen Zweige überdeckt werden. Notieren Sie zu diesem neuen Testfall wieder die Reihenfolge der durchgeführten Anweisungen.

d) Wie viele Pfade gibt es? Sind alle davon möglich? (Bitte jeweils mit kurzer Begründung.)

e) Formulieren Sie ein minimales Programm, für das mindestens zwei verschiedene Testfälle notwendig sind, um eine Anweisungsüberdeckung zu erreichen.

Aufgabe 2. *Refactoring* (8 Punkte)

a) Überlegen Sie sich welche Probleme beim Verlagern einer Methode in eine andere Klasse (Move Method Refactoring) auftreten können. Beachten Sie insbesondere auch Sichtbarkeiten und Vererbung. Beschreiben Sie mindestens … verschiedene Probleme.

b) Führen Sie mit Eclipse auf einem bestehenden Programm drei verschiedene automatisierte Refactorings durch. Beschreiben Sie was Sie tun, und was das System tut.

Tipp: Wenn Sie im Java-Editor von Eclipse mit dem Cursor auf eine Klasse / Variable / Methode gehen und die rechte Maustaste drücken, wird Ihnen unter „Refactor“ ein Katalog möglicher Refactorings angeboten.

Das SWT-Team wünscht gutes Gelingen und viel Erfolg für die Klausur!