

Übungen zur Vorlesung Softwaretechnologie

-Wintersemester 2008/2009-
Dr. Günter Kniesel, Daniel Morales

Übungsblatt 12

Abgabetermin am 25. Januar, 23:59 Uhr

Bitte fangen Sie **frühzeitig** mit der Bearbeitung des Zettels an, damit wir Ihnen bei Bedarf helfen können!!!! Lösen Sie die Aufgaben mit Together 2007 in UML2.0, wenn nichts anderes angegeben ist. Fragen zu Übungsaufgaben respektive zur Vorlesung können Sie auf der Mailingliste swt-tutoren@iai.uni-bonn.de, bzw. swt-vorlesung@iai.uni-bonn.de stellen.

Aufgabe 1. *Blackbox Test* (9 Punkte)

Gegeben sei folgende Schnittstelle:

```
interface StringAnalyzer {  
    String containedCharacters (String input);  
}
```

Die Methode `containedCharacters` erhält einen beliebigen String als Eingabeparameter. Die Methode untersucht, welche ASCII-Buchstaben aus [n-t, N-T] (also Buchstaben von „n“ bis „t“ in Groß- und Kleinschreibung) im String `input` vorkommen, und gibt diese in Form eines sortierten Strings zurück. Falls derselbe Buchstabe mehrmals im Eingabestring vorkommt, so soll der Rückgabestring nur einen Repräsentanten dieses Buchstabens enthalten. Der Rückgabestring sollte alphabetisch sortiert sein. Falls ein Buchstabe im Eingabestring in Groß- und in Kleinschrift vorkommt, so steht der Großbuchstabe vor dem entsprechenden Kleinbuchstaben. Alle anderen Zeichen werden ignoriert.

- Überlegen Sie, welche Fehlerarten auftreten können.
- Geben Sie für jede Fehlerart wenigstens einen Eingabestring und das erwünschte Ergebnis an.
- Schreiben Sie eine Java-Klasse `StringAnalyzerTest`, mit der Sie unter Zuhilfenahme von *JUnit* die Methode `containedCharacters` aus *MyAnalyzer.class* testen. *MyAnalyzer* implementiert das Interface `StringAnalyzer`. Beides ist auf der SWT-Homepage herunterzuladen.
Hinweis: Informationen zum Einbinden von externen Klassen und Bibliotheken finden Sie im technischen Hinweis auf der Vorlesungsseite.

Aufgabe 2. Whitebox Test (8 Punkte)

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

```

public int[] sortiere(int[] bestand) { // Anweisungs-
    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
}

```

- Entwerfen Sie für die Methode `sortiere` einen Kontrollflussgraphen.
- 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.
- Erreichen Sie mit diesem Feld an Eingabewerten auch eine Verzweigungsabdeckung? Begründen Sie kurz Ihre Antwort. Falls nicht, dann geben Sie ein weiteres Feld an, sodass auch die übrigen Zweige überdeckt werden. Notieren Sie zu diesem neuen Testfall wieder die Reihenfolge der durchgeführten Anweisungen.
- Wie viele Pfade gibt es? Kurze Begründung.

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

Aufgabe 3. *Junit und Refactorings* (7 Punkte)

- a) **(4 Punkte)** Überlegen Sie sich welche Probleme beim Verlagern einer Methode in eine andere Klasse (Move Method Refactoring) und beim Extrahieren von Programmcode aus einer Methode (Extract Method Refactoring) auftreten können.

Tip: beachten Sie auch Sichtbarkeiten und Vererbung!

- b) **(3 Punkte)** Erkunden Sie auch die Refactoring-Fähigkeiten von Eclipse. Schreiben Sie dafür ein kleines Programm, und führen Sie mit Eclipse drei verschiedene automatisierte Refactorings durch. Beschreiben Sie was Sie tun, und was das System tut.

Tip: Wenn Sie im Java-Editor von 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.