

## Einführung in Trusted Systems WS 2007/08

### Übung 13 29. Januar 2008

#### **Aufgabe 1: Black-Box vs. White-Box**

1. Welche dieser beiden Testing-Techniken würden Sie anwenden, um möglichst viele Bugs während der Testing-Phase zu finden: Black-Box oder White-Box Testing?
2. Wo liegen die Unterschiede zwischen Black-Box und White-Box Testing?
3. Warum braucht man "Static Testing"? Ist es nicht genug die Software mittels "Dynamic Testing" zu testen?

#### **Aufgabe 2: Coverage**

1. Welche der folgenden Aussagen sind korrekt und welche nicht? Begründen Sie Ihre Antworten.
  - (a) Aus 100% *Statement coverage* folgt 100% *Branch coverage*.
  - (b) Aus 100% *Branch coverage* folgt 100% *Statement coverage*.
  - (c) Aus 100% *Path coverage* folgt 100% *Statement/Branch coverage*.
2. Gegeben sei das unten stehende Programm. Definieren Sie eine minimale Anzahl von Test Cases, um 100%
  - (a) Statement coverage
  - (b) Branch coverage
  - (c) Path coveragezu erreichen.

```
1. int foo(int p1, int p2, int p3) {  
2.     if (p1 > p2 || p2 >= p3)  
3.         p3 = p3 - p2;  
4.     if (p3 % 2 == 0)  
5.         p3--;  
6.     return p3;  
7. }
```

3. Gegeben sei der unten stehende Abschnitt aus der Definition einer Klasse in der Programmiersprache JAVA. Der Abschnitt beschreibt eine Methode dieser Klasse, welche die Attribute `current` und `returnValue` (vom Typ `Object`) und die Methode `nextEntry()` verwendet. Das Attribut `current` wird entweder mit einer Referenz zu einem `Object` oder mit `NULL` initialisiert (Initialisierung erfolgt in der Methode `nextEntry()`). Wenn die Variable `current` nach dem Aufruf der Methode `nextEntry()` ein Objekt referenziert, hat dieses Objekt ein public Attribute `Date`, welches die Methoden `getA` und `getB` exportiert.
- (a) Zeichnen Sie das Kontrollflussdiagramm der Methode `nextElement`.
  - (b) Spezifizieren Sie welche Pfade nötig sind, um 100% statement coverage zu erreichen.
  - (c) Spezifizieren Sie welche Pfade nötig sind, um 100% branch coverage zu erreichen.

```
1. public Object nextElement() {
2.     // @pre : current != null
3.     Object returnValue = current.datum.getA();
4.     this.nextEntry();
5.     while (current != null) {
6.         if(current.datum.getB() == value)
7.             break;
8.         this.nextEntry();
9.     }
10.    return returnValue;
11. }
```

**Plagiarismus** *Der Fachbereich Informatik misst der Einhaltung der Grundregeln der wissenschaftlichen Ethik großen Wert bei. Zu diesen gehört auch die strikte Verfolgung von Plagiarismus. Mit der Abgabe einer Lösung (Hausaufgabe, Programmierprojekt, Diplomarbeit, etc. ) bestätigen Sie, dass (Sie/Ihre Gruppe) (der alleinige Autor/die alleinigen Autoren) des gesamten Materials sind. Falls Ihnen die Verwendung von Fremdmaterial gestattet war, so müssen Sie dessen Quellen deutlich zitiert haben. Bei Unklarheiten zu diesem Thema finden Sie weiterführende Informationen unter <http://www.informatik.tu-darmstadt.de/Plagiarism>.*