

# Abgabedokument

## VU Software-Qualitätssicherung

Bitte beachten Sie, dass nur richtig benannte Abgabedokumente im richtigen Format bewertet werden! Welche Dateien in welchem Format abzugeben sind entnehmen Sie der Angabe.

Ihre Daten – Bitte füllen Sie alle Felder aus:

<b>Nachname, Vorname:</b>	Huter, Martin
<b>Matrikelnummer:</b>	0727981
<b>Studienkennzahl:</b>	033 533
<b>E-Mail Adresse:</b>	0727981@student.tuwien.ac.at

Auf Übung 3 können Sie bis zu 32 Punkte erreichen.

Bei der Übung handelt es sich um eine Einzelarbeit. Plagiate werden mit 0 Punkten bewertet. Weiters behalten wir es uns vor, Studenten im Verdachtsfall zu einem **Kontrollgespräch** einzuladen.

### 3. Bugsuche (6 Punkte)

#### 3.1. Beispiel 1

Nein das Ergebnis ist nicht immer das Selbe.  $X^0 = 1$  für ein beliebiges  $X$ , jedoch gibt der Algorithmus für `pow(x,0)`,  $x$  zurück.

#### 3.2. Beispiel 2

Hier haben wir es mit einem Überlauf zu tun wenn man die Funktion richtig ansteuert.

Da in der Schleife alle Zahlen von 1 bis  $a$  aufsummiert werden haben wir relativ schnell große Zahlen.

Ab 65536 ( $2^{16}$ ) ist das größer als `MAX_INT` in Java weshalb die Zahl negativ wird da ein Überlauf zustande kommt.

Ab 92682 ist es wieder positiv, und ab dann ist es alternierend in gewissen abständen.

#### 3.3. Beispiel 3

Nein das ist nicht fall, durch die Darstellung von Floats begründet, wenn ich  $a = 10^{38}$  wähle (nahe dem Maximum in Java) und  $b = 10^{-45}$  (nahe der Zahl mit der längsten Mantisse begründet durch die Anzahl der Führenden 0en).

Durch die Kalkulation mit Division und Multiplikation entsteht ein Rundungsfehler, weshalb  $a \neq (c) * b$  und deswegen wird `false` zurückgegeben

## 5. Code Coverage (6 Punkte)

### 5.1. Anweisungsüberdeckung

(A,B,0), (1,B,-1)

### 5.2. Zweigüberdeckung

{c0 Eingaben}, (-1,0,0)

### 5.3. Mehrfach-Bedingungsüberdeckung

{c1 Eingaben}

Da man durch (-1,0,0) in eine Endlosschleife kommt ist es nicht möglich beide Fälle des zweiten Ifs abzudecken. Somit kommt bei c2 kein weiterer Fall hinzu.

## 6. Theoriefragen (3 Punkte)

### 6.1. Erklären Sie den Unterschied der Coverage Kategorien c0 - c3.

**Wann ist der Einsatz welcher Coverage sinnvoll?**

#### **Anweisungsüberdeckung c0:**

Ziel ist es jede Anweisung mindestens einmal durchzuführen um Deadcode zu finden.

Es ist jedoch ein schwaches Kriterium zur Fehlerfindung da die Auswirkung von Fehlern an Testdaten gekoppelt ist.

#### **Zweigüberdeckung c1:**

Hier wird jede Verzweigung getestet, dh jedes if sollte auf true/false getestet werden und alle switches entsprechend ihrer Bedingungen. Wird oft in Kontrollfluss orientierten testen verwendet.

#### **Bedinungsüberdeckung c2:**

Ist im Prinzip die kaskadierung der c1 Tests. Das heißt, dass jede Kombination von Verzweigungen getestet wird. Folgedessen entstehen  $2^n$  mögliche Testfälle, wobei n die Anzahl der Verzweigungen darstellt.

### 6.2. Er Nennen Sie mindestens zwei Gründe für den Einsatz von Testdoubles (Mock, Stubs, etc.).

Testdoubles werden dann benutzt wenn die realen Objekte nicht zur Verfügung stehen.

Folgende Gründe können verantwortlich sein:

- Objekt nicht fertig wenn es für Test benötigt wird

- Abhängigkeiten zu Zeit und Datum bestehen bei dem Objekt.

- Aufwändig das Objekt in den benötigten Objektstatus zu versetzen.

### **6.3. Erklären Sie den Unterschied zwischen Mock Objects und Stubs.**

Stub:

Liefert vordefinierte Ergebnisse beim Aufrufen

Mock Objects:

Liefert auch vordefinierte Ergebnisse, überprüft jedoch auch die Eingabeparameter und Testet somit die Eingabeparameter des Mock Objekts