

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:

Nachname, Vorname:	Delalic, Salmir
Matrikelnummer:	0947919
Studienkennzahl:	033 533
E-Mail Adresse:	Delalic.salmir@gmail.com

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, bei zu großen val und pow, wird der Speicherbereich von long überschritten was zu einer radikalen Grenzziehung führt. Das Ergebnis wird in diesem Fall 0

3.2. Beispiel 2

Nein, wenn der Parameter „a“ zu groß gewählt wird, überschreitet „res“ die Integer-Grenze und kriegt ein negatives Vorzeichen aufgrund der Darstellung.

3.3. Beispiel 3

Nein, bei der Division kann es zum Abschneiden der letzten Kommastellen die über den Speicherbereich gehen, kommen. Diese wiederum verursachen eine Ungenauigkeit für die nachfolgende Multiplikation und die beiden „a“ floats unterscheiden sich ganz gering, aber doch voneinander was „false“ liefert.

5. Code Coverage (6 Punkte)

5.1. Anweisungsüberdeckung

Zweig n_2 - n_3 wird nie ausgeführt, wenn $\text{doSomeStuff}(a \leq 0, b \geq 0, c < 0)$.

Zweig n_1 - n_3 wird nie verlassen, wenn $\text{doSomeStuff}(a \leq 0, b \leq 0, c < 0)$.

5.2. Zweigüberdeckung

$\text{doSomeStuff}(1,2,3) \rightarrow n_1, n_5, n_7, n_{\text{final}}$

$\text{doSomeStuff}(1,2,0) \rightarrow n_1, n_5, n_6, n_{\text{final}}$

$\text{doSomeStuff}(1,2,-1) \rightarrow n_1, n_2, n_3, n_1, n_5, n_7, n_{\text{final}}$

$\text{doSomeStuff}(1,-2,-1) \rightarrow n_1, n_2, n_3, n_1, n_2, \dots$ endlos

$\text{doSomeStuff}(-1,2,-1) \rightarrow n_1, n_1, n_1, \dots$ endlos

5.3. Mehrfach-Bedingungsüberdeckung

n_1 true $\rightarrow c < 0$

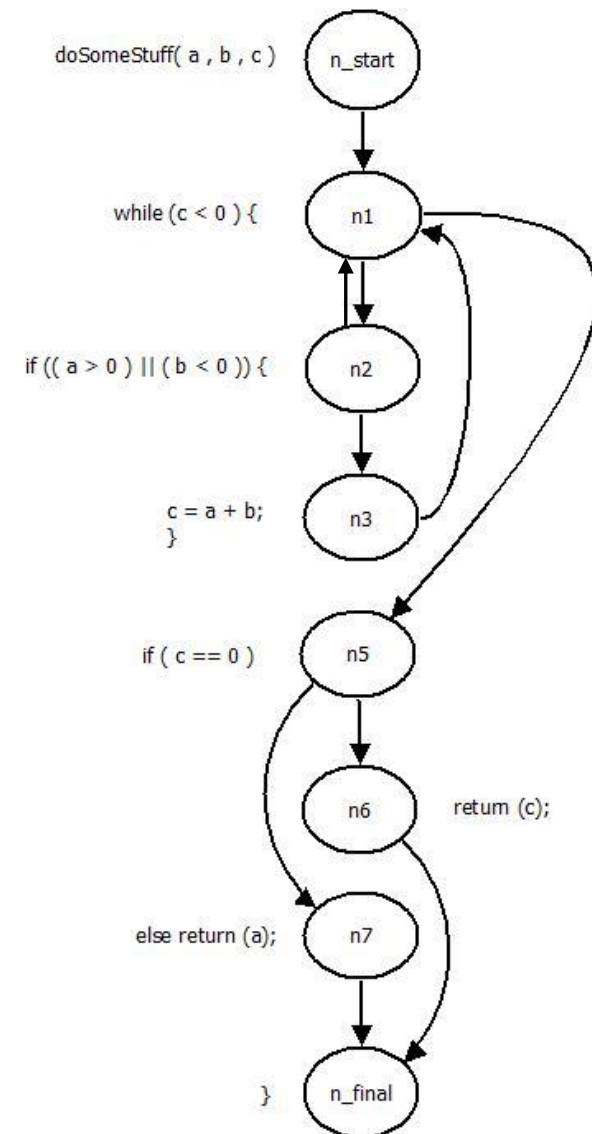
n_1 false $\rightarrow c \geq 0$

n_2 true $\rightarrow c < 0 \ \&\& \ \text{if}(a > 0 \ || \ b < 0)$

n_2 false $\rightarrow c < 0 \ \&\& \ \text{if}(a \leq 0 \ || \ b \geq 0)$

n_5 true $\rightarrow c == 0$

n_5 false $\rightarrow c > 0$



6. Theoriefragen (3 Punkte)

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

Wann ist der Einsatz welcher Coverage sinnvoll?.

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

1. Testdoubles müssen eingesetzt werden, wenn die zu testende Klasse eine Internetverbindung braucht (grabber).
Mit einem Mock kann so ein Datensatz aus dem Internet „gedoublet“ werden und die weitere Interaktion getestet.
2. Wenn die zu testende Software später erweitert oder an einem anderen System gekoppelt wird, welches der Entwickler/Tester zu diesem Zeitpunkt nicht hat und so Rückgabewerte simulieren zu können.

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

Stubs sind Doubles einer Klasse oder eines Objekts, die einen vordefinierten (oder erwarteten) Rückgabewert haben, welcher für das eigentliche Testen sinnvoll ist.

Mockups dagegen, sind Doubles wie Stubs, die aber auch übernommene Werte überprüfen und so direkt in den Test miteinfließen. zB werden Methodenspezifische Überprüfungen von Parametern aus der zu mockenden Klasse übernommen.