

Softwarekonstruktion – Übung 6

6 Whitebox-Testen II

6.1 Grenze-Inneres-Überdeckung

Das gegebene Programmsegment soll mit einem Kontrollfluss-Testverfahren getestet werden:

```
1 while ( (a<b) OR (c<d) ) {  
2     a:= a + 1;  
3     c:= c + 1;  
4 }
```

1. Geben Sie einen minimalen Satz von Testdaten an, der für die Bedingung der while-Schleife die minimal bestimmende Mehrfachbedingungsüberdeckung erfüllt.
2. Geben Sie einen minimalen Satz von Testdaten an, der die Grenze-Inneres-Überdeckung erfüllt.

Gegeben sei nun ein Programmsegment und sein aus einem Refactoring entstandenes Gegenstück.

```
1 factorial= counter;          1 /* Refactoring */  
2 counter--;                  2 factorial=1;  
3 while ( counter > 0 ) {     3 do {  
4     factorial= factorial * counter; 4     factorial= factorial * counter;  
5     counter--;              5     counter--;  
6 }                             6 } while ( counter > 0 )  
7 System.out.println(factorial); 7 System.out.println(factorial);
```

3. Geben Sie jeweils Testfälle für eine Grenze-Inneres-Überdeckung an. Worin unterscheiden sich die beiden Testfälle? Wodurch kommt der Unterschied zu Stande?

Auch das folgende Programmsegment soll mit einem Kontrollfluss-Testverfahren getestet werden:

```
1  x=a; y=b;
2
3  /* ggT (klassisch): */
4  while (a!= b) {
5      if (a>b) { a= a-b; }
6      else {b= b-a; }
7  }
8
9  /* ggT (modern): */
10 tmp = 0;
11 while (y != 0) {
12     tmp = x % y;
13     x = y;
14     y = tmp;
15 }
```

4. Geben Sie für den Testdatensatz $\{(8, 2), (5, 5), (6, 3)\}$ den Grenze-Inneres-Überdeckungsgrad an.
5. Geben Sie einen minimalen Satz von Testdaten an, der die Grenze-Inneres-Überdeckung erfüllt.

6.2 Kontrollflussbezogenes Testen – Testfälle II

Das gegebene Programmsegment soll mit einem Kontrollfluss-Testverfahren getestet werden:

```
1 read(x, y, z)
2 if ( (x>0) AND (y>0) AND (z>0) ) { x:=x*y*z }
3 else { x:=y }
4 if ( x+y+z>0 ) { z:=-x }
```

6.2.1 *Geben Sie eine minimale Testmenge für eine Anweisungsüberdeckung an.*

6.2.2 *Geben Sie eine minimale Testmenge für eine einfache Bedingungsabdeckung an.*

6.2.3 *Geben Sie eine minimale Testmenge für eine Mehrfachbedingungsabdeckung an.*

6.2.4 *Geben Sie eine minimale Testmenge für eine minimal bestimmende Mehrfachbedingungsabdeckung an.*

6.2.5 *Geben Sie eine minimale Testmenge für eine Pfadabdeckung an.*

6.3 Kontrollflussbezogenes – Refactoring

Bei einem Refactoring wurde das Programmsegment aus Aufgabe 6.2 wie folgt geändert:

```

1 read(x,y,z)
2 if ( x>0 ) {
3     if ( y>0 ) {
4         if ( z>0 ) { x:=x*y*z }
5         else { x:=y }
6     }
7     else { x:=y }
8 }
9 else { x:=y }
10 if ( x+y+z>0 ) { z:=-x }
  
```

Das Programmsegment soll nun erneut getestet werden:

- 6.3.1 *Geben Sie eine minimale Testmenge für eine Anweisungsüberdeckung an. Vergleichen Sie Ihr Ergebnis mit dem Ergebnis von Aufgabe 6.2.1 und begründen Sie gegebenenfalls den Unterschied.*
- 6.3.2 *Geben Sie eine minimale Testmenge für eine einfache Bedingungsabdeckung an. Vergleichen Sie Ihr Ergebnis mit dem Ergebnis von Aufgabe 6.2.2 und begründen Sie gegebenenfalls den Unterschied.*
- 6.3.3 *Geben Sie eine minimale Testmenge für eine Mehrfachbedingungsabdeckung an. Vergleichen Sie Ihr Ergebnis mit dem Ergebnis von Aufgabe 6.2.3 und begründen Sie gegebenenfalls den Unterschied.*
- 6.3.4 *Geben Sie eine minimale Testmenge für eine minimal bestimmende Mehrfachbedingungsabdeckung an. Vergleichen Sie Ihr Ergebnis mit dem Ergebnis von Aufgabe 6.2.4 und begründen Sie gegebenenfalls den Unterschied.*

Hausübung

Die Hausübung findet online statt!

Alle nötigen Informationen (beachten Sie insbesondere die Abgabefrist) sind der Webseite zur Vorlesung zu entnehmen:

<https://www-secse.cs.tu-dortmund.de/secse/pages/teaching/ws14-15/swk/>

bzw.

https://www-secse.cs.tu-dortmund.de/secse/pages/teaching/ws14-15/swk/index_de.shtml#onlineuebungen