

## Übungen zu Software-Qualität

Wintersemester 2012/2013

### Übungsblatt 8

#### Aufgabe 8.1 (Required- $k$ -Tuples-Test, 28 Punkte)

Vergegenwärtigen Sie sich nochmals die bereits bekannte Methode `sqrHeron()` (siehe Rückseite dieses Aufgabenblattes). Sie finden die Methode auch in der JAVA-Klasse unter der folgenden Internet-Adresse:

<http://www-lehre.inf.uos.de/~sq/2012/aufgaben/blatt08/SqrHeron.java>

Erzeugen Sie für die Methode `sqrHeron()` die benötigten Testfälle, mit denen ein Test nach dem Required- $k$ -Tuples-Kriterium für  $k = 3$  durchgeführt werden kann. Gehen Sie dafür wie folgt vor:

- Geben Sie den Kontrollflussgraphen an und bestimmen Sie alle  $l$ -dr-Interaktionen mit  $2 \leq l \leq k$ . Stellen Sie sicher, dass Ihr Kontrollflussgraph die Bedingungen für den Required- $k$ -Tuples-Test einhält! (12 Punkte)
- Erzeugen Sie aus den  $l$ -dr-Interaktionen die Menge der möglichen Interaktionen-Subpfade (Schleifenkriterien beachten). Berücksichtigen Sie hierfür alle Ihre 2-dr-Interaktionen aber nur beispielhaft eine 3-dr-Interaktion. (12 Punkte)
- Bestimmen Sie die benötigten Testpfade und Eingabedaten, so dass alle Subpfade abgedeckt werden und begründen Sie jeweils kurz. (4 Punkte)

#### Aufgabe 8.2 (Datenkontext-Überdeckung, 18 Punkte)

Erzeugen Sie für die Methode `sqrHeron()` die benötigten Testfälle, mit denen ein Test nach dem Datenkontext-Überdeckungskriterium durchgeführt werden kann. Gehen Sie dafür wie folgt vor:

- Modifizieren Sie den Kontrollflussgraphen der Methode `sqrHeron()` und bilden Sie die *geordneten* elementaren Datenkontexte zu allen Knoten. Fassen Sie anschließend die elementaren Datenkontexte zu *geordneten* Datenkontexten pro Knoten zusammen. (8 Punkte)
- Fügen Sie die Subpfade, die sich aus den Datenkontexten der Knoten ergeben, zu Testpfaden zusammen und stellen Sie Ihre Lösung grafisch dar. (8 Punkte)
- Bestimmen Sie die benötigten Testpfade und Eingabedaten, so dass alle Subpfade abgedeckt werden und begründen Sie jeweils kurz. (2 Punkte)

#### Aufgabe 8.3 (Offener Frageteil, 5 Punkte)

Beantworten Sie Ihrer Tutorin bzw. Ihrem Tutor Fragen zur Veranstaltung „Software-Qualität“.

– bitte wenden –

```

01  /**
02   * Berechnung einer Naeherung der Quadratwurzel einer Zahl mit einer Folge
03   * nach dem Heronverfahren.
04   *
05   * @param zahl eine Zahl, von der die Quadratwurzel berechnet werden soll
06   * @return Quadratwurzel zur uebergebenen Zahl, -1 fuer Eingaben < 0
07   */
08  public static double sqrtHeron(double zahl) {
09      // Schranke, vordefiniert fuer einstellige Zahl
10      double epsilon = 1e-15;
11      // max. Anzahl Folgenglieder
12      final int MAXIMUM = 100000;
13      // aktuelles Folgenglied
14      double x = 0;
15      // Folgengliednummer
16      int zaehler = 1;
17
18      // liefere -1 fuer Eingaben < 0
19      if (zahl < 0) {
20          x = -1;
21      }
22
23      // zahl <= 0 laesst sich nicht berechnen
24      if (zahl > 0) {
25          // aktuelles Folgenglied mit uebergebenen Zahl initialisieren
26          x = zahl;
27
28          // Kopie der uebergebenen Zahl erstellen
29          double kopie = zahl;
30
31          // Berechnung der Schranke
32          do {
33              // solange 'kopie' noch mehr als eine Stelle vor dem Komma hat
34              // ziehe eine Stelle ab...
35              kopie = kopie / 10;
36              // ... und multipliziere Schranke mit 10
37              epsilon = epsilon * 10;
38          } while (kopie > 1);
39
40          // Berechnung der Quadratwurzel
41          do {
42              // naechstes Folgenglied berechnen
43              x = (x + zahl / x) / 2.0;
44              zaehler++;
45          } while (zaehler < MAXIMUM && Math.abs(x * x - zahl) >= epsilon);
46      }
47      return x;
48  }

```