Institut für Informatik
Jun.-Prof. Dr.-Ing. Elke Pulvermüller
B. Sc. Sergey Krutikov
B. Sc. Alexander Brandt

Universität Osnabrück, 27.11.2012
`http://www-lehre.inf.uos.de/~sq`

Testat bis Freitag, 07.12.2012, 19:00 Uhr

Übungen zu Software-Qualität

Wintersemester 2012/2013

Übungsblatt 7

Aufgabe 7.1 (Defs/Uses-Test, 40 Punkte)

Machen Sie sich mit der Methode `sqrtHeron()` auf der Rückseite des Aufgabenblattes vertraut. Sie finden die Methode auch in der JAVA-Klasse unter der folgenden Internet-Adresse:

`http://www-lehre.inf.uos.de/~sq/2012/aufgaben/blatt07/SqrtHeron.java`

a) Erzeugen Sie den Kontrollflussgraphen der Methode `sqrtHeron()` in Datenflussdarstellung (d. h. Datenflussattribute ergänzen).

b) Geben Sie eine Menge von Testfällen an, die das *All-Defs*-Kriterium erfüllen und begründen Sie kurz anhand der Hilfsmengen *dcu* und *dpu*.

c) Geben Sie eine Menge von Testfällen an, die das *All-p-Uses*-Kriterium erfüllen und begründen Sie kurz anhand der Hilfsmengen *dcu* und *dpu*.

d) Geben Sie eine Menge von Testfällen an, die das *All-c-Uses*-Kriterium erfüllen und begründen Sie kurz anhand der Hilfsmengen *dcu* und *dpu*.

e) Geben Sie eine Menge von Testfällen an, die das *All-c-Some-p-Uses*-Kriterium erfüllen und begründen Sie kurz anhand der Hilfsmengen *dcu* und *dpu*.

f) Geben Sie eine Menge von Testfällen an, die das *All-p-Some-c-Uses*-Kriterium erfüllen und begründen Sie kurz anhand der Hilfsmengen *dcu* und *dpu*.

g) Geben Sie eine Menge von Testfällen an, die das *All-Uses*-Kriterium erfüllen und begründen Sie kurz anhand der Hilfsmengen *dcu* und *dpu*.

h) Geben Sie eine Menge von Testfällen an, die das *All-du-Path*-Kriterium erfüllen und begründen Sie kurz anhand der Hilfsmengen *dcu* und *dpu*.

Erzeugen Sie die benötigten *dcu*- und *dpu*-Mengen. Geben Sie für alle Testfälle die jeweiligen Eingabedaten und Testpfade an. Stellen Sie eine eindeutige Zuordnung von den Testfällen zu den jeweils zu überdeckenden *dcu*- und *dpu*-Mengen sicher.

Aufgabe 7.2 (Offener Frageteil, 10 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  }
```