Prof. Dr. Stephan Kleuker
Hochschule Osnabrück
Fakultät Ing.-Wissenschaften und Informatik
- Software-Entwicklung -

Software-Qualität

Sommersemester 2024
6. Aufgabenblatt

Aufgabe 0.6 (0 Punkte)
Geben Sie das Lösungswort des Quiz aus der Lernnotiz an.

Aufgabe 12 (1 Punkt)

Beantworten Sie folgende Fragen und Aufgaben schriftlich in ganzen Sätzen oder mit Stichpunkten oder einer Graphik.

a) Mit dem vorgestellten Datenfluss-Ansatz kann geprüft werden, ob die Deklaration einer Variablen überflüssig ist, wie?
b) Gegeben sei die Methode m auf der rechten Seite, zeichnen Sie den zugehörigen Datenflussgraphen zusammen mit def, c.use, p-use, dcu und dpu.

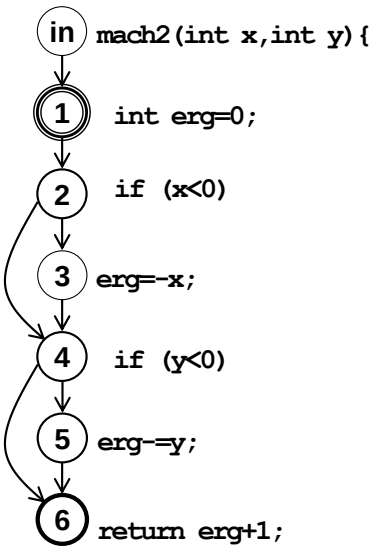
```
public int m(int x) {
  int erg = 0;
  switch (x) {
    case 0: {
      erg = 1;
      break;
    }
    case 1:
    case 2: {
      erg = 2;
      break;
    }
    default: {
      return 42;
    }
  }
  return erg;
}
```

Aufgabe 13 (3 Punkte)

Markieren Sie den Graphen auf der rechten Seite mit def(.), c-use(.), p-use(.).
Ergänzen Sie die Markierungen für dcu(.,.) und dpu(.,.). Geben Sie dann jeweils eine möglichst kleine Menge von Testfällen an, die zusammen das

a) all defs-Kriterium
b) all p-uses-Kriterium
c) all c-uses-Kriterium
d) all uses-Kriterium

erfüllen. Es reicht die Angabe von Knotenfolgen, z. B. (in, 1, 2, 3, 4, 5, 6). Der Graph befindet sich vergrößert auch auf der nächsten Seite, um z. B. Markierungen dort direkt einzutragen.



Aufgabe 14 (2 Punkte)

Gegeben sei die Methode schritt() aus der Klasse spiel.Conway aus dem Projekt qsAufgabeUeberdeckungConway von der Veranstaltungsseite. Schreiben Sie Testfälle für eine vollständige C2-Überdeckung (s. unten). Starten Sie Ihre Überlegungen aber mit sinnvollen Testfällen, die alle typischen Fälle beinhalten. Das berechnete Ergebnis muss vereinfachend für die Aufgabe nicht geprüft werden, die zugehörige Aufgabe 38 steht in http://kleuker.iui.hs-osnabrueck.de/WiSe23_Prog1/Prog1_WiSe23_Blatt11.pdf.

	Coverage	Covered Instructio...	Missed Instructions	Total Instructions
schritt()	100,0 %	100	0	100

Prof. Dr. Stephan Kleuker
Hochschule Osnabrück
Fakultät Ing.-Wissenschaften und Informatik
- Software-Entwicklung -

Software-Qualität

Sommersemester 2024
6. Aufgabenblatt

Aufgabe 15 (4 Punkte, Test First, allerdings Test von anderer Person erstellt)

Zu implementieren ist eine Klassenmethode dauer(.) in der Klasse helper.Utility, die zwei Datums-Objekte vom Typ java.util.Date übergeben bekommt und die die Dauer an Arbeitstagen zwischen diesen beiden Tagen einschließlich der jeweiligen Tage zurückgibt. Sollte das zweite Datum vor dem ersten liegen, wird eine IllegalArgumentException geworfen. Die Methode soll in einem Projektmanagement-Werkzeug genutzt werden, dabei spielen für das Datum nur Tag, Monat und Jahr eine Rolle. Beim Ergebnis soll berücksichtigt werden, dass samstags und sonntags nicht gearbeitet wird, Feiertage werden nicht beachtet. Für den Kalender auf der rechten Seite sollen damit die Ergebnisse der Tabelle auf der rechten Seite berechnet werden. Beachten Sie, dass angegebene Daten am Wochenende liegen können.

Nr.	Start	Ende	dauer
0	17.3.2014	16.3.2014	Exception
1	17.3.2014	17.3.2014	1
2	17.3.2014	21.3.2014	5
3	17.3.2014	22.3.2014	5
4	17.3.2014	23.3.2014	5
5	17.3.2014	24.3.2014	6
6	14.3.2014	17.3.2014	2
7	15.3.2014	17.3.2014	1
8	15.3.2014	15.3.2014	0
9	16.3.2014	17.3.2014	1
10	1.3.2014	30.3.2014	20
11	1.3.2014	28.2.2015	260
12	1.3.2015	29.2.2016	261

Die Tests finden Sie im Projekt qsAufgabeTestFirstDauer auf der Veranstaltungsseite.

a) Schreiben Sie die Methode dauer(.), nutzen Sie dazu das Paket java.time, das seit Java 8 Bestandteil von Java ist.
b) Prüfen Sie, ob eine vollständige C2-Überdeckung Ihres Codes durch die Testfälle erfolgt. Ist dies nicht der Fall, klären Sie warum und ergänzen Sie gegebenenfalls Testfälle.

Prof. Dr. Stephan Kleuker
Hochschule Osnabrück
Fakultät Ing.-Wissenschaften und Informatik
- Software-Entwicklung -

Software-Qualität

Sommersemester 2024
6. Aufgabenblatt