



Aufgabe 0.6 (1 Punkt)

Geben Sie das Lösungswort des Quiz aus der Lernnotiz an.

Aufgabe 12 (1 Punkt)

Beantworten Sie folgende Frage schriftlich. Mit dem vorgestellten Datenfluss-Ansatz kann geprüft werden, ob die Deklaration einer Variablen überflüssig ist, wie?

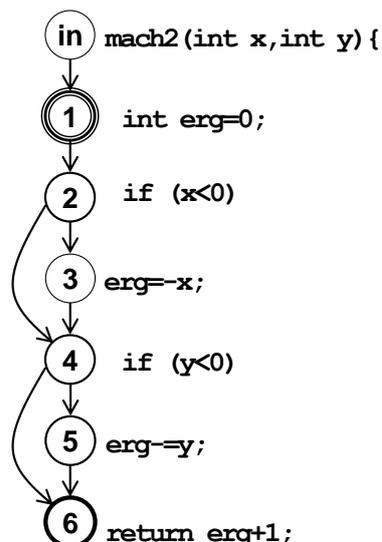
Aufgabe 13 (3 Punkte)

Markieren Sie den Graphen auf der linken 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

- all defs-Kriterium
- all p-uses-Kriterium
- all c-uses-Kriterium
- 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 (1 Punkt)

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 (Branches) für EclEmma. 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 37 steht in <https://kleuker.iui.hs-osnabrueck.de/querschnittlich/Prog1 WiSe24 Blatt11.pdf>.

| | | | | |
|-------------|---------|----|---|----|
| ● schritt() | 100,0 % | 30 | 0 | 30 |
|-------------|---------|----|---|----|

Aufgabe 15 (3 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 nachfolgenden Tabelle berechnet werden.

| März 2014 | | | | | | |
|-----------|----|----|----|----|----|----|
| Mo | Di | Mi | Do | Fr | Sa | So |
| 24 | 25 | 26 | 27 | 28 | 1 | 2 |
| 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 17 | 18 | 19 | 20 | 21 | 22 | 23 |
| 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| 31 | 1 | 2 | 3 | 4 | 5 | 6 |



Die Tests finden Sie im Projekt
qsAufgabeTestFirstDauer auf der
Veranstaltungsseite.

- a) Schreiben Sie die Methode `dauer(.)`, nutzen Sie dazu das Paket `java.time`, z. B. mit den Klassen `LocalDate` und `Duration`, das seit Java 8 Bestandteil von Java ist. Vermeiden Sie die Nutzung von Schleifen oder Streams.
- 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.

| 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 |

