

## Übungsblatt 7 – Ausgabe am 08.06.2020

Abgabe bis 15.06.2020, 12 Uhr, per Mail

### Aufgabe 7.1: Kontroll- und Datenflussanalyse.

Gegeben sei die folgende (fehlerhaft implementierte) Methode:

```
/**
 * Berechnet Anzahl der Biere, die mit einer gegebenen Menge
 * Wasser, Malz und Hopfen gebraut werden kann.
 * @param gewAnzahl gewünschte Anzahl, die gebraut werden soll
 * @param wasser verfügbares Wasser in L
 * @param malzUndHopfen verfügbare Menge Malz und Hopfen in kg
 * @return Anzahl der Biere, die gebraut werden können
 */

1 public static int berechneAnzahl(int gewAnzahl, double wasser,
   double malzUndHopfen) {
2   int gebrauteBiere = 0;
3   String biername;
4   boolean genugGebraut = false;
5   for(; gebrauteBiere < gewAnzahl; gebrauteBiere++) {
6     if(wasser > 0.0 && malzUndHopfen > 0.0) {
7       wasser -= 10;
8       malzUndHopfen -= 0.1;
9     } else {
10      genugGebraut = true;
11      break;
12    }
13  }
14  System.out.println(gebrauteBiere + " " + biername + " können
   gebraut werden");
15  return gebrauteBiere;
16 }
```

- a) Erstellen Sie den *Kontrollflussgraphen* für die Methode „**berechneAnzahl**“. Vergeben Sie Knotennamen jeweils entsprechend der zugehörigen Zeilennummern.
- b) Versehen Sie die Knoten des Kontrollflussgraphen mit den *Datenfluss-Attributen*  $d(x)$ ,  $c(x)$ ,  $p(x)$  und  $u(x)$  für alle Parameter und lokalen Variablen. Ordnen Sie die Datenfluss-Attribute eines Knotens entsprechend der Auswertungsreihenfolge von links nach rechts an.
- c) Suchen Sie im Beispiel nach Datenfluss-Anomalien und listen Sie diese in einer Tabelle wie folgt auf:

Variable	Typ	stark?	Pfad
malzUndHopfen	du	nein	01-02-03-04-05-14-15
...	...	...	...

- d) Erstellen Sie den *Datenflussgraphen* und versehen Sie alle Datenflusskanten mit den entsprechenden Variablennamen.
- e) Erweitern Sie den Datenflussgraphen zu einem *Abhängigkeitsgraphen*.
- f) Listen Sie die Menge der Knoten und Kanten auf, die den *Vorwärts-Slice* für Knoten 10 bilden.