

Konzepte der Programmierung

Lösungsskizzen – Übungsblatt 7

Aufgabe 7.1

(a) Sichtbarkeiten und Lebensdauer

- **x** (Deklaration Z. 3): Innerhalb der ganzen Klasse lebendig (Z. 2 bis 31), überall sichtbar außer in Zeilen 6-16 und 26-29
- **name** (Deklaration Z. 4): Innerhalb der ganzen Klasse lebendig, überall sichtbar außer in Methode **fak()** also den Zeilen 18-31 und in Zeile 14/15. Jedoch erfolgt darauf ein qualifizierter Zugriff in Zeile 22
- Parameter **x** (Deklaration Z. 6): Sichtbar und lebendig im Rumpf der Methode **setzeName()** (Z. 6-16)
- **name** (Deklaration Z. 14): Nur sichtbar und lebendig in Zeile 14 und 15 (bis zum Ende des **else**-Blocks)
- Parameter **name** (Deklaration Z. 18): Sichtbar und lebendig im Rumpf der Methode **fak()** (Z. 18-31)
- **sum** (Deklaration Z. 19): Sichtbar und lebendig ab der Deklaration bis zum Ende der Methode (Z. 19-31)
- **x** (Deklaration Z. 26): Sichtbar und lebendig im Rumpf der for-Schleife (Z. 27-29)
- **a** (Deklaration Z. 28): Sichtbar und lebendig nur in Zeile 28 und 29

- (b) Die globale Variable **x** wird in der Methode **setzeName()** (Z.6-16) durch den gleichnamigen Parameter **überdeckt** und in Zeile 27-29 durch den for-Schleifen-Zähler.

Die globale Variable **name** wird überdeckt durch die lokale Variable in Zeile 14 und 15 und durch den gleichlautenden Übergabeparameter der Methode **fak()** in Z. 16-31.

Globale Variablen können jedoch durch *qualifizierten Zugriff* (**this.<variablenName>**) auch in Überdeckungsbereichen gelesen und somit verändert werden. Dies ist bei Klassenfeldern ebenfalls möglich durch den qualifizierten Zugriff mithilfe des Klassennamens (**<klassenname>.<statischerVariablenName>**), bei Objektfelder jedoch nur mit **this**.

- (c) Es handelt sich um eine **rekursive** Methode, d. h. sie ruft sich selbst im Rumpf auf.
- (d) An der Sichtbarkeit und Lebensdauer der Variablen ändert sich nichts.

Aufgabe 7.2

siehe BlueJ Projekt

Aufgabe 7.3

- (a) siehe BlueJ Projekt

- (b) • **reinige()** hat zwei Seiteneffekte, der Wasserstand wird reduziert und es erfolgt eine Ausgabe

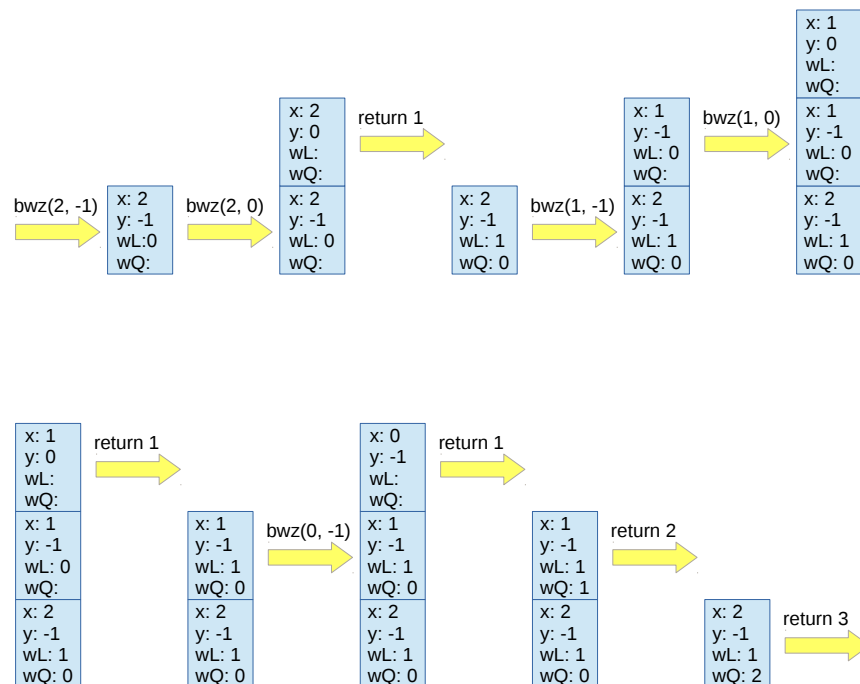
- `schalteEin()/schalteAus()`: zwei Seiteneffekte: Veränderung des Feldes `istAn` und Methodenaufruf mit Seiteneffekt
- `makeKaffee()`: Seiteneffekte: Konsolenausgaben und Veränderung der Objektfelder
- Prüfmethode: Für den Fall, dass eine Ressource erschöpft ist, gibt es als Seiteneffekt die Konsolenausgabe. Würde diese fehlen, wären die Methoden seiteneffektfrei.

Aufgabe 7.4

siehe BlueJ Projekt

Aufgabe 7.5 (Veranschaulichung der Rekursion)

(a) Folge von Aufrufstapeln für `berechneWegezahl(2, -1)`:



(b) Ausführungsprotokoll für `berechneWegezahl(3,-2)`:

```
bwz(3, -2)
  bwz(3, -1)
    bwz(3, 0)
    bwz(2, -1)
      bwz(2, 0)
      bwz(1, -1)
        bwz(1, 0)
        bwz(0, -1)
  bwz(2, -2)
    bwz(2, -1)
      bwz(2, 0)
      bwz(1, -1)
        bwz(1, 0)
        bwz(0, -1)
    bwz(1, -2)
      bwz(1, -1)
        bwz(1, 0)
        bwz(0, -1)
      bwz(0, -2)
```

(c) Die Methode ist:

- nicht linear, da sie mehr als einmal aufgerufen wird
- direkt, da sich selbst und nicht eine andere Methode aufruft
- nicht schlicht, da der rekursive Aufruf nicht die letzte Anweisung ist die ausgeführt wird.