

ÜBUNGS-BLOCK 5

AUFGABEN



Aufgabe 1:

In dem Video zu den komplexen Datentypen aus dem Video-Block 5 haben wir die Klasse „Color“ gesehen. Du weißt jetzt, wo du dir weiterhelfen kannst, wenn du zu dem Umgang mit Ressourcen aus der Java-Bibliothek Fragen hast.

Schreibe ein Programm, dass:

1. In einer Variablen auf ein Color-Objekt der Farbe gelb referenziert
2. Die Farbkanäle dieses Objekts auf der Konsole ausgibt, im Format:
`<R> - <G> - `
3. Anschließend den Farbton etwas dunkler macht
4. Am Ende nochmals die Farbkanäle auf der Konsole ausgibt

Weitere Aufgaben auf der nächsten Seite!

Aufgabe N-1:

Sieh dir mal folgenden Code an:

```
public static void main(String[] args) {
    int x = 0;
    int y = 0;
    initializeXY(x, y);
    System.out.println("x=" + x + ", y=" + y);
}

private static void initializeXY(int x, int y) {
    x = 5;
    y = 8;
}
```

Wenn du in den Videos gut aufgepasst hast, dann solltest du wissen, dass die Ausgabe hier x=0, y=0 lautet, und **nicht** x=5, y=8, wie es der Programmierer hier wohl erwartet hatte.

Der Grund: Java übergibt nicht die Variablen selbst in eine Methode, sondern *kopiert* lediglich deren *Werte* in neue Variablen - nämlich die, die in der Parameterliste der Methode deklariert sind. Wie du weißt nennt sich dieser Übergabemechanismus Call By Value.

Das Gegenstück dazu nennt sich *Call By Reference* - dabei werden tatsächlich die Referenzen auf die ursprünglichen Variablen übergeben, und neue Zuweisungen zu diesen Variablen haben einen nachhaltigen Effekt, auch außerhalb der Methode. Der Programmierer, der diesen Code geschrieben hat, dachte wohl, dass Java mit Call By Reference arbeiten kann, so wie es in anderen Programmiersprachen (z.B. C) der Fall ist - stimmt aber nicht, es gibt nur Call By Value in Java!

Deine Aufgabe:

Finde einen Weg, um nun *trotzdem* den Effekt des Call By Reference in Java zu "simulieren". Es geht also darum, dass in obigem Beispiel dann tatsächlich die Werte 5 und 8 ausgegeben werden, und das unter der Bedingung, dass:

- 1) weiterhin die initializeXY-Methode genutzt wird, um die Werte zu setzen
- 2) Die initializeXY Methode nach wie vor void bleibt, also nichts zurückgibt!

Das einzige, was du an der Methode verändern darfst - und das ist nun der **Tipp** den ich dir gebe: Du darfst die Parameterliste der Methode verändern! Entsprechend darfst/musst du dann natürlich auch in der main-Methode Änderungen vornehmen.

Diese Aufgabe ist ganz schön schwierig, aber ich bitte dich darum, dir Zeit zu nehmen und darüber nachzudenken, wie man den gewünschten Effekt irgendwie realisieren könnte. Denke "out of the box" - verbeiße dich nicht in den gegebenen Code. Es gibt auch soetwas wie "außerhalb dieses Codes". Werde kreativ! Viel Spaß beim Grübeln :)

PS: Okay, *einen* Tipp gibt's doch noch: Was war nochmal der Unterschied zwischen einer Zuweisung und einer De-Referenzierung...

Aufgabe 2:

Gegeben sind folgende Klassen:

```
public class KlasseA {

    private String sa;

    public KlasseA(String sa) {
        setSa(sa);
    }

    public String getSa() {
        return sa;
    }

    public void setSa(String sa) {
        this.sa = sa;
    }
}

public class KlasseB {

    private KlasseA a;
    private String sb;

    public KlasseB(KlasseA a) {
        setA(a);
        setSb(a.getSa());
    }

    public String getSb() {
        return sb;
    }

    public void setSb(String sb) {
        this.sb = sb;
    }

    public void setA(KlasseA a) {
        this.a = a;
    }
}
```

Sowie das folgende Programm:

```
public class GarbageCollection {
    public static void main(String[] args) {

        KlasseA a = new KlasseA(new String("A1"));
        // 1
        KlasseB b = new KlasseB(new KlasseA(new String("A2")));
        // 2
        a = null;
        // 3
        b.setSb("B");
        // 4
        b.setA(null);
        // 5
    }
}
```

Im Programm ist jede zweite Zeile mit einem Kommentar versehen. Für jeden Zeitpunkt, an dem die Ausführung des Programms an einer kommentierten Zeile angelangen würde, sollst du eine Skizze des Speicher- und Referenzmodells zu diesem Zeitpunkt malen.

Orientiere dich an den Skizzen die ich dazu in den Videos gezeichnet habe. Die Skizzen soll für diesen Zeitpunkt also jeweils aufzeigen:

- 1) Welche Objekte es im Hauptspeicher gibt
- 2) Welche Variablen es gibt
- 3) Welche Referenzen (Zeiger von Variablen auf Objekte) es gibt

Anmerkung: Es soll für diese Aufgabe davon ausgegangen werden, dass der Garbage Collector auf Hochtouren arbeitet und ein Objekt, sobald es zur Räumung freigegeben ist, *sofort* aus dem Speicher löscht.