

Übungsblatt 2

Abgabedatum: 31.10.2021

Die Abgabe Ihrer Lösungen erfolgt vor Ablauf der Abgabefrist digital über die Moodle-Plattform. Erstellen Sie dazu ein PDF-Dokument, das die Lösungen Ihrer schriftlichen Aufgaben enthält. Laden Sie dieses PDF-Dokument und den erarbeiteten Java-Code (.java-Dateien) mit den in den Aufgaben vorgegebenen Namen bei Moodle hoch. Bitte laden Sie die Dateien einzeln hoch, Dateiarhive (z.B. .zip-Dateien) werden nicht akzeptiert.

Sie können maximal **(5 Punkte)** mit diesem Übungsblatt erreichen.

Aufgabe 1 (Zahlensysteme)

0,5 Punkte

1. Legen Sie eine neue Klasse `Zahlensysteme.java` an und fügen Sie folgenden Code ein.

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

        // a)
        System.out.println("a = " + 15);
        System.out.println("b = " + 015);

        // b)
        System.out.println(0.25 * 100 == 25);
        System.out.println(0.362 * 100 == 36.2);

        // c)
        System.out.println(0xAFFE);
        System.out.println(10 + 0b10 + 010 + 0x10);

        // d)
        System.out.println("float " + (987654321f - 987654320f) + " vs int " +
            (987654321 - 987654320));
    }
}
```

Führen Sie das Programm aus und erläutern Sie, warum für a und b unterschiedliche Werte ausgegeben werden.

2. Erläutern Sie, warum der erste Ausdruck `0.25 * 100 == 25` „true“ der Ausdruck `0.362 * 100 == 36.2` allerdings „false“ zurückliefert.
3. Rechnen Sie den Wert $[AFFE]_{16}$ händisch ins Dezimalsystem um. Geben Sie ihren Rechenweg an. Lösen Sie die Aufgabe $10 + [10]_2 + [10]_8 + [10]_{16}$, indem Sie zunächst alle Werte in Dezimalzahlen umrechnen. Geben Sie wieder ihren Rechenweg an.

4. Wie lässt sich erklären, dass nicht beide Rechnungen bei d (987654321f - 987654320f und 987654321 - 987654320) auf 1 kommen?

Aufgabe 2 (Subtraktion)

0,5 Punkte

1. Schreiben Sie ein Java-Programm (Subtraktion.java), das zwei Zufallszahlen a und b mit den Wertebereichen $a \in \{50, 51, 52, \dots, 100\}$ und $b \in \{1, 2, 3, \dots, 50\}$ erzeugt und anschließend den Text „Was ist die Subtraktion von $\ll \text{Wert von } a \gg$ und $\ll \text{Wert von } b \gg$?“ ausgibt. Für $\ll \text{Wert von } a \gg$ bzw. $\ll \text{Wert von } b \gg$ soll dabei der tatsächliche Wert von a bzw. b ausgegeben werden.

Nutzen Sie zum Erzeugen von Zufallszahlen die Funktion `Math.random()`. Die genannte Funktion erzeugt einen `double`-Wert der größer gleich 0 und kleiner 1 ist. Überlegen Sie sich, wie Sie diesen `double`-Wert verwenden können, um Variablen mit entsprechenden Wertebereichen zu erzeugen.

2. Erweitern Sie das Programm aus Aufgabe a so, dass der Nutzer nach der Ausgabe des Textes „Was ist die Subtraktion von $\ll \text{Wert von } a \gg$ und $\ll \text{Wert von } b \gg$?“ eine Zahl eingeben kann. Das Programm soll nach Eingabe einer Zahl `true` ausgeben falls diese Zahl die Subtraktion aus a und b ist und `false`, wenn dies nicht der Fall ist. Nutzen Sie dazu den Ausdruck `a == b`, der für die Eingaben a und b einen Booleschen-Wert zurück gibt, der bei Zahlen angibt, ob die Werte der beiden Zahlen gleich sind (`true`) oder nicht (`false`).
3. Testen Sie das Programm mehrfach mit richtigen und falschen Eingaben.

Aufgabe 3 (Zinseszins)

1,5 Punkte

1. Schreiben Sie ein Java-Programm Zinseszins.java, das das Startkapital (K_0 in €), einen Zinssatz (p in %) und die Dauer der Anlage (n in Jahren) einliest und das Kapital nach n Jahren auf 2 Nachkommastellen gerundet ausgibt. Zum Runden kann z.B. die Methode `Math.round(double a)` verwendet werden.
2. Erstellen Sie eine weitere Klasse Dauer.java die nach Eingabe des Startkapitals, des Zinssatzes und des Endkapitals die Dauer der Investition in Jahren berechnet, die mindestens benötigt wird, um das entsprechende Endkapital zu erreichen.
3. Erstellen Sie eine Klasse Zinssatz.java die nach Eingabe des Startkapitals, des Endkapitals und der Dauer den Zinssatz in Prozent berechnet.

Aufgabe 4 (Retro-Textmodus)

2,5 Punkte

In der Vorgabe finden Sie die Datei `Console.java`, welche einen Textmodus-Bildschirm simuliert, wie er in den 1980er und frühen 1990er-Jahre verbreitet war. Die Anzeige besteht aus einer Gitter von 80×25 Zellen. Jede Bildschirmzelle kann ein Zeichen darstellen und separat mit einer Vordergrundfarbe (Textfarbe) und einer Hintergrundfarbe belegt werden. Ein Textcursor markiert die aktuelle Position auf dem Bildschirm. Es existieren verschiedene Funktionen zur Textausgabe und zur Manipulation der Bildschirmfarben und Cursorposition.

1. Die Vorgabe enthält das Java-Programm `ConsoleDemo.java`, welche die Funktionalitäten von `Console` zeigt. Es muss sich im gleichen Ordner wie `Console.java` befinden. Führen Sie das Programm aus und machen Sie sich mit den den gezeigten Funktionen vertraut. Schauen Sie Sich die Dokumentation von `Console` an und experimentieren Sie.

2. Legen Sie ein neues Java-Programm `Textmodus.java` an. Nutzen Sie die Console-Funktionen zur Textausgabe und Farbmanipulation um eine (individuell gestaltete) Blütenpflanze aus farbigen Textzeichen in die Mitte des Textbildschirms zu zeichnen. Die Pflanze soll mindestens 7 Bildschirmzeilen hoch und 5 Spalten Breite umfassen, grüne Blätter haben sowie zwei verschiedene Blütenfarben enthalten. Nutzen Sie `Console.gotoXY()`, um die Pflanzenteile zu positionieren.

Hinweis: Alternativ zu Blütenpflanzen können Sie auch mehrfarbige Fische, Fahrzeuge etc. gestalten (Kreativität ist erwünscht). Wichtig ist, dass eine feste Basisfarbe und zwei weitere Dekorfarben vorhanden sind. Halten Sie auch die Vorgaben zur Mindestgröße ein.

3. Erweitern Sie Ihr Programm um eine Methode `show1(Color color1, Color color2, int row, int column)` und verschieben Sie den Code zur Darstellung der Pflanze in diese Methode. Modifizieren Sie ihren Code so, dass die Position, an denen die Pflanze angezeigt wird, durch die Parameter `row` und `column` bestimmt wird. Zudem sollen für die zwei Blütenfarben die Parameter `color1` und `color2` verwendet werden.
4. Rufen Sie `show1` in der `main`-Methode mehrfach mit unterschiedlichen Parametern auf, um mehrere unterschiedlich gefärbte Pflanzen an verschiedenen Positionen auf dem Bildschirm anzuzeigen.
5. Programmieren Sie zwei weitere Methoden `show2` und `show3` mit der gleichen Signatur (Parametern) wie `show1`. Die neuen Methoden sollen jeweils deutlich anderes gestaltete Pflanzentypen an variabler Position und mit variablen Farben darstellen. Ergänzen Sie die `main`-Methode um mehrere Aufrufe zu `show2` und `show3` mit verschiedenen Parametern für Position und Farben um die Anzeige zu einem gemischten Blumenbeet (bzw. Aquarium/Parkplatz) zu erweitern.