Bereich: Klassen (1)

Radio	Schwierigkeit: ★☆☆☆☆
Package: `de.dhbwka.java.exercise.classes`	**Klasse:** `Radio`

Aufgabenstellung:

a) Implementieren Sie eine Klasse `Radio` mit folgenden Attributen:

- `on`, ob das Radio an (`true`) oder aus (`false`) ist
- `volume`, wie laut spielt das Radio Musik?
  (Die Lautstärke soll nur im Bereich von 0 bis 10 liegen und ganzzahlig sein)
- `frequency`, gibt die Frequenz des gewählten Senders an
  (Erlaubter Frequenzbereich ist von 85.0 bis 110.0)

b) Implementieren Sie zu der Klasse zwei Konstruktoren der Art:

- `Radio()` mit sinnvollen Startwerten
- `Radio(boolean on, int volume, double frequency)`

c) Zu der Klasse Radio sollen folgende Methoden implementiert werden:

- `void incVolume(),void decVolume()`
  Diese Methoden sollen die Lautstärke um 1 Einheit ändern (nur möglich im Zustand an).
- `void turnOn(),void turnOff()`
  Diese Methoden sollen den Zustand des Attributs eingeschaltet (`on`) ändern.
- `void setFrequency(double frequency)`
  Diese Methode soll eine Frequenz speichern. Ist die gewählte Frequenz außerhalb des erlaubten Frequenzbereichs, soll die Frequenz 99.9 gewählt werden.
- `public String toString()`
  Diese Methode soll Informationen über den internen Zustand als String zurückgeben. Sie soll eine Zeichenkette der Form "Radio an: Freq=98.4, Laut=2" zurückgeben.

d) Testen Sie Ihre Klasse mit der `main()`-Methode oder aus einer anderen Klasse heraus!

Beispiel für eine Testmethode:

```
public static void main(String[] args) {
	Radio radio = new Radio(false, 7, 93.5);
	System.out.println(radio);
	radio.turnOn();
	System.out.println(radio);
	radio.incVolume();  radio.incVolume();
	System.out.println(radio);
	radio.incVolume();
	radio.incVolume();
	System.out.println(radio);
	radio.decVolume();
	System.out.println(radio);
	radio.setFrequency(97.8);
	System.out.println(radio);
	radio.setFrequency(112.7);
	System.out.println(radio);
	radio.turnOff();
	System.out.println(radio);
```

```
}
```
Zugehörige Ausgabe:
```
Radio aus; Lautstärke 7; Frequenz 93.5 MHz
Radio an; Lautstärke 7; Frequenz 93.5 MHz
Radio an; Lautstärke 9; Frequenz 93.5 MHz
Radio an; Lautstärke 10; Frequenz 93.5 MHz
Radio an; Lautstärke 9; Frequenz 93.5 MHz
Radio an; Lautstärke 9; Frequenz 97.8 MHz
Radio an; Lautstärke 9; Frequenz 99.9 MHz
Radio aus; Lautstärke 9; Frequenz 99.9 MHz
```

Bereich: Klassen (1)

Zweidimensionaler Punkt	Schwierigkeit: ★★☆☆☆
Package: `de.dhbwka.java.exercise.classes`	**Klasse:** `Point`

Aufgabenstellung:

Die Klasse `Point` soll einen Punkt in einem zweidimensionalen Koordinatensystem repräsentieren. Ein Punkt besteht aus folgenden Komponenten:

- X-Koordinate
- Y-Koordinate

a) Legen sie die Attribute dieser Klasse fest!

b) Schreiben Sie sinnvolle Konstruktoren für die Klasse!

c) Erzeugen Sie `getter`- und `setter`-Methoden für die Attribute und setzen Sie die Modifikatoren für Methoden und Attribute so, dass das Prinzip der Kapselung für die Attribute gilt, d.h. auf die Attribute soll von außerhalb der Klasse Punkt nur über die `getter`- bzw. `setter`-Methoden zugegriffen werden können!

d) Schreiben Sie Methoden mit dem folgenden Verhalten zu dieser Klasse:

- `public String toString()`
- Abstand des Punktes zum Ursprung
- Spiegelung des Punktes an der X-Achse (Ergebnis soll ein neuer Punkt sein!)
- Spiegelung des Punktes an der Y-Achse (Ergebnis soll ein neuer Punkt sein!)
- Spiegelung des Punktes am Ursprung (Ergebnis soll ein neuer Punkt sein!)
- Abstand des Punktes zu einem anderen Punkt.

e) Schreiben Sie ein Hauptprogramm, das alle Methoden ausgiebig testet!

Hinweise:

- Wählen Sie „sprechende“ Namen für Ihre Methoden!
- Der Typ des Rückgabewertes einer Methode kann auch ein Referenztyp sein und deshalb auch vom Typ `Point`.

Beispiel für eine Testmethode:

```
public static void main(String[] args) {
    Point pointA = new Point(4.0, 2.0);
    System.out.println("A: " + pointA);
    Point pointB = new Point(-1.0, -1.0);
    System.out.println("B: " + pointB);
    System.out.println("Abstand A-B: "
            + pointA.getDistance(pointB));
    pointA = pointA.mirrorOrigin();
    System.out.println("A': " + pointA);
    System.out.println("Abstand A'-B: "
            + pointA.getDistance(pointB));
}
```

Zugehörige Ausgabe:

```
A: Punkt (4.0,2.0)
B: Punkt (-1.0,-1.0)
Abstand A-B: 5.830951894845301
A': Punkt (-4.0,-2.0)
Abstand A'-B: 3.1622776601683795
```

Bereich: Klassen (1)

Bankkonto	Schwierigkeit: ★★☆☆☆
Package: `de.dhbwka.java.exercise.classes`	**Klasse:** `Account`

Aufgabenstellung:

Schreiben Sie eine Klasse `Account`, die ein Bankkonto realisiert.
Attribute für das Bankkonto sind die Kontonummer, der Name des Kontoinhabers, der Kontostand sowie ein Limit, bis zu dem das Konto überzogen werden darf.

a) Erstellen Sie einen oder mehrere geeignete Konstruktoren!

b) Erstellen Sie Instanzenmethoden für folgende Operationen:

- `String`-Darstellung des Bankkontos
- Einzahlung
- Auszahlung
- Abfrage des Kontostandes

Machen Sie dabei auch Plausibilitätsprüfungen, z.B. kann nur ein positiver Betrag eingezahlt werden.

Beispiel für Testmethode und zugehörige Ausgabe:

```
public static void main(String[] args) {
    Account account = new Account(4711, "Donald Duck", 500, 1000);
    System.out.println(account);
    account.processDeposit(200);
    System.out.println(account);
    account.processPayment(400);
    System.out.println(account);
    account.processPayment(2000);
    System.out.println(account);
}
```

```
Konto Nr. 4711 (Donald Duck), Stand: 500 ct, Limit 1000 ct
Konto Nr. 4711 (Donald Duck), Stand: 700 ct, Limit 1000 ct
Konto Nr. 4711 (Donald Duck), Stand: 300 ct, Limit 1000 ct
Deckung nicht ausreichend!
Konto Nr. 4711 (Donald Duck), Stand: 300 ct, Limit 1000 ct
```