

## Bereich: Klassen (1)

### Radio

### Musterlösung

**Package:** de.dhbwka.java.exercise.classes

**Klasse:** Radio

```
package de.dhbwka.java.exercise.classes;

/**
 * @author DHBW lecturer
 * @version 1.0
 *
 * Part of lectures on 'Programming in Java'. Baden-Wuerttemberg
 * Cooperative State University.
 *
 * (C) 2018 by J. Sidler, T. Schlachter, C. Schmitt, W. Süß
 */
public class Radio {

    boolean on;
    int volume;
    double frequency;

    public Radio() {
        this(true, 5, 98.4);
    }

    public Radio(boolean on, int volume, double frequency) {
        this.on = on;
        if (volume < 0) {
            this.volume = 0;
        } else if (volume > 10) {
            this.volume = 10;
        } else {
            this.volume = volume;
        }
        this.setFrequency(frequency);
    }

    public void turnOn() {
        this.on = true;
    }

    public void turnOff() {
        this.on = false;
    }

    public void incVolume() {
        if (on && volume<10)
            volume++;
    }

    public void decVolume() {
        if (on && volume>0)
            volume--;
    }
}
```

```
public void setFrequency(double freq) {
    if (freq >= 85.0 && freq < 110.0)
        this.frequency = freq;
    else
        this.frequency = 99.9; // standard freq.
}

@Override
public String toString() {
    return "Radio " + (on ? "an" : "aus")
        + "; Lautstärke " + volume
        + "; Frequenz " + frequency + " MHz";
}

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);
}
}
```

## Bereich: Klassen (1)

### Zweidimensionaler Punkt

### Musterlösung

**Package:** de.dhbwka.java.exercise.classes

**Klasse:** Point

```
package de.dhbwka.java.exercise.classes;

/**
 * The Point class represents locations in a two-dimensional coordinate space
 * (x,y). The coordinates are specified in double precision.
 *
 * @author DHBW lecturer
 * @version 1.0
 *
 * Part of lectures on 'Programming in Java'. Baden-Wuerttemberg Cooperative
 * State University.
 *
 * (C) 2018 by J. Sidler, T. Schlachter, C. Schmitt, W. Süß
 */
public class Point {

    // Attributes (encapsulated)
    private double x, y;

    /**
     * Constructs and initializes a point at the origin (0, 0) of the coordinate
     * space.
     */
    public Point() {
        this(0.0, 0.0);
    }

    /**
     * Constructs and initializes a point at the specified (x,y) location in the
     * coordinate space.
     *
     * @param x x coordinate
     * @param y y coordinate
     */
    public Point(double x, double y) {
        super();
        this.x = x;
        this.y = y;
    }

    /**
     * Returns the x coordinate of this Point.
     *
     * @return x coordinate of this Point
     */
    public double getX() {
        return x;
    }

    /**
     * Sets the x coordinate of this Point to the specified value.
     */
}
```

```
*
* @param x x coordinate
*/
public void setX(double x) {
    this.x = x;
}

/**
 * Returns the y coordinate of this Point.
 *
 * @return y coordinate of this Point
 */
public double getY() {
    return y;
}

/**
 * Sets the y coordinate of this Point to the specified value.
 *
 * @param y y coordinate
 */
public void setY(double y) {
    this.y = y;
}

/**
 * Returns a new Point with the coordinates of this Point mirrored (flipped)
 * at the x axis.
 *
 * @return the resulting Point
 */
public Point mirrorXAxis() {
    return new Point(x, -y);
}

/**
 * Returns a new Point with the coordinates of this Point mirrored (flipped)
 * at the y axis.
 *
 * @return the resulting Point
 */
public Point mirrorYAxis() {
    return new Point(-x, y);
}

/**
 * Returns a new Point with the coordinates of this Point mirrored (flipped)
 * at the origin (0, 0) of the coordinate space.
 *
 * @return the resulting Point
 */
public Point mirrorOrigin() {
    return new Point(-x, -y);
}

/**
 * Returns the distance of this Point from the Point given.
 *
```

```
* @param p the Point of which the distance from this Point shall be
* calculated
* @return distance of this Point from the the origin (0, 0)
*/
public double getDistance(Point p) {
    double dx = x - p.x;
    double dy = y - p.y;
    return Math.sqrt(dx * dx + dy * dy);
}

/**
 * Returns the distance of this Point from the the origin (0, 0).
 */
public double getMagnitude() {
    return getDistance(new Point(0.0, 0.0));
}

/**
 * Returns a string representation of this Point.
 *
 * @return string representation of this Point
 */
@Override
public String toString() {
    return "Punkt (" + x + ", " + y + ")";
}

/**
 * main method, only for test purposes of class Point.
 *
 * @param args start parameters (not used)
 */
public static void main(String[] args) {
    Point pointA = new Point(4.0, 2.0);
    System.out.println("A: " + pointA);
    System.out.println("Betrag: " + pointA.getMagnitude());
    Point pointB = new Point(-1.0, -1.0);
    System.out.println("B: " + pointB);
    System.out.println("Abstand: "
        + pointA.getDistance(pointB));
    pointA = pointA.mirrorOrigin();
    System.out.println("A': " + pointA);
    System.out.println("Abstand: "
        + pointA.getDistance(pointB));
}
}
```

## Bereich: Klassen (1)

### Bankkonto

### Musterlösung

**Package:** de.dhbwka.java.exercise.classes

**Klasse:** Account

```
package de.dhbwka.java.exercise.classes;

/**
 * @author DHBW lecturer
 * @version 1.0
 *
 * Part of lectures on 'Programming in Java'. Baden-Wuerttemberg
 * Cooperative State University.
 *
 * (C) 2018 by J. Sidler, T. Schlachter, C. Schmitt, W. Süß
 */
public class Account {

    private int number;
    private String holder;
    private int balance;
    private int limit; // limit is a positive value

    public Account() {
        this(-1, "n/n", 0, 0);
    }

    public Account(int number, String holder, int balance, int limit) {
        super();
        this.number = number;
        this.holder = holder;
        this.balance = balance;
        this.limit = limit;
    }

    public int getBalance() {
        return balance;
    }

    public void processDeposit(int amount) {
        if (amount > 0)
            this.balance += amount;
    }

    public void processPayment(int amount) {
        if (amount > 0 && balance-amount >= -limit)
            this.balance -= amount;
        else
            System.err.println("Deckung nicht ausreichend!");
    }
}
```

```
@Override
public String toString() {
    return "Konto Nr. " + number
        + " (" + holder + "), "
        + "Stand: " + balance + " ct, "
        + "Limit " + limit + " ct";
}

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);
}
}
```