

P1 - Serie 03

(1) Klasse `Thermometer`

Programmieren Sie eine Klasse `Thermometer`, welche einen einfachen Fieberthermometer modelliert. Die Klasse soll eine Temperatur in Celsius als einzige Instanzvariable speichern. Der Konstruktor soll diese Instanzvariable standardmässig auf 37.0 Grad setzen. Schreiben Sie eine Methode `increase`, welche die Temperatur um 0.1 Grad erhöht und einen Getter für die Temperatur. Zudem definieren Sie eine Methode `reset`, welche die Temperatur wieder auf 37.0 zurücksetzt.

Schreiben Sie eine zweite Klasse `ThermometerTest`, in der Sie zwei Objekte vom Typ `Thermometer` instanzieren und deren Methoden ausführlich testen.

Klasse `Thermometer`

```
1  public class Thermometer {
2
3      private double temperature;
4
5      public Thermometer() {
6          this.temperature = 37.0;
7      }
8
9      public void increase() {
10         this.temperature += 0.1;
11     }
12
13     public double getTemperature() {
14         return this.temperature;
15     }
16
17     public void reset() {
18         this.temperature = 37.0;
19     }
20 }
```

Aufgabe 2: Car -Klasse

(2) Klasse `Car`

Programmieren Sie eine Klasse `Car`, welche die Marke, das Modell und den Jahrgang des Fahrzeuges modelliert. Der Konstruktor soll diese drei Instanzvariablen gemäss Parameterübergabe initialisieren – zudem schreiben Sie Getter und Setter für alle Instanzvariablen und eine `toString` Methode für eine einzeilige Repräsentation von `Car` Objekten. Schliesslich definieren Sie eine Methode `isAntique`, welche einen boolean zurückgibt, der anzeigt ob das Auto aktuell älter ist als 45 Jahre.

In einer zweiten Klasse `Garage` instanzieren Sie drei `Car` Objekte und testen alle programmierten Methoden.

```
import java.time.LocalDate;

public class Car {

    private String brand;
    private String model;
    private int year;
    private boolean isAntique;
    private final int ANTIQUE_AGE = 45;

    public Car(String brand, String model, int year) {
        this.brand = brand;
        this.model = model;
        this.year = year;
        this.isAntique = checkIfAntique();
    }

    public String getBrand() {
        return this.brand;
    }

    public String getModel() {
        return this.model;
    }

    public int getYear() {
        return this.year;
    }

    public boolean getIsAntique() {
```

```

        return this.isAntique;
    }

    public void setBrand(String brand) {
        this.brand = brand;
    }

    public void setModel(String model) {
        this.model = model;
    }

    public void setYear(int year) {
        this.year = year;
        this.isAntique = checkIfAntique();
    }

    private boolean checkIfAntique() {
        int currentYear = LocalDate.now().getYear();
        if (currentYear - this.year > ANTIQUE_AGE) {
            return true;
        } else {
            return false;
        }
    }

    public boolean isAntique() {
        return this.isAntique;
    }

    public String toString() {
        return this.brand + " " + this.model + " " + this.year + " Is Antique: " + this.isAntique;
    }
}

```

(3) Cargo und Box-Klassen

Schreiben Sie eine Klasse `Cargo`, welche ein Stückgut mit Länge, Breite, Höhe und einem Namen modelliert (z.B. 30, 44, 65, "Kaffeemaschine"). Schreiben Sie einen Konstruktor, Getter und Setter für alle Instanzvariablen und eine Methode `toString`.

Schreiben Sie eine Klasse `Box`, die Instanzvariablen für die Länge, Breite und Höhe einer Box enthält. Zusätzlich enthält die Klasse `Box` eine Instanzvariable `full` vom Typ `boolean`, die angibt, ob die Box gefüllt ist oder nicht, sowie eine Instanzvariable `cargo` vom Typ `Cargo`. Der Konstruktor setzt die Länge, Breite und Höhe einer Box gemäss Parametern – neu instanziierte `Box` Objekte sollen standardmässig leer sein. Definieren Sie einen zweiten Konstruktor ohne Parameter, der eine Standard-Box mit Länge, Breite und Höhe 1 generiert. Zusätzlich definieren Sie eine Methode `getCapacity`, die das Volumen der Box berechnet und zurückgibt.

Schliesslich schreiben Sie eine Methode `addCargo`, welche ein Objekt vom Typ `Cargo` als Parameter entgegennimmt. Falls dieses Stückgut gemäss Länge, Breite und Höhe in die Box passt, passen Sie die Variable `full` und die Instanzvariable `cargo` an und geben `true` zurück (andernfalls `false`).

Testen Sie die Klasse `Box`, indem Sie in einer weiteren Klasse `BoxTest` drei `Box` Objekte instanziiieren, manipulieren und ausgeben.

Mögliche Lösung

Klasse Cargo

```

public class Cargo {

    private String name;
    private int length;
    private int width;
    private int height;

    public Cargo(String name, int length, int width, int height) {
        this.name = name;
        this.length = length;
        this.width = width;
        this.height = height;
    }

    public String getName() {
        return this.name;
    }

    public int getLength() {
        return this.length;
    }

    public int getWidth() {
        return this.width;
    }
}

```

```

    public int getHeight() {
        return this.height;
    }

    public void setName(String name) {
        this.name = name;
    }

    public void setLength(int length) {
        if (length > 0) {
            this.length = length;
        }
    }

    public void setWidth(int width) {
        if (width > 0) {
            this.width = width;
        }
    }

    public void setHeight(int height) {
        if (height > 0) {
            this.height = height;
        }
    }

    public String toString() {
        return "Cargo: \tName: " + this.name + " Dimensions: " + this.length + "x" + this.width + "x" + this.height;
    }
}

```

Klasse Box

```

import java.util.ArrayList;
import java.util.Collections;

public class Box {

    private Cargo cargo;
    private boolean isFull;
    private int length;
    private int width;
    private int height;

    public Box(int length, int width, int height) {
        this.length = length;
        this.width = width;
        this.height = height;
        this.isFull = false;
    }

    public Box() {
        this(1, 1, 1); // refers to the other constructor
    }

    public int getCapacity() {
        return this.length * this.width * this.height;
    }

    //... all other getters

    public void setLength(int length) {
        if (length > 0) {
            this.length = length;
        }
    }

    //... all other setters

    public boolean addCargo(Cargo cargo) {
        if (this.isFull) {
            System.out.println("Box already full");
            return false;
        }

        /*
         * System welches sicherstellt, dass das Cargo schlaue eingepackt wird. Box kann ja gedreht werden
         * ArrayList<Integer> erstellt ein Array
         */
    } else {

```

```

        ArrayList<Integer> cargoDimensions = new ArrayList<>();
        cargoDimensions.add(cargo.getLength());
        cargoDimensions.add(cargo.getWidth());
        cargoDimensions.add(cargo.getHeight());
        ArrayList<Integer> boxDimensions = new ArrayList<>();
        boxDimensions.add(this.length);
        boxDimensions.add(this.width);
        boxDimensions.add(this.height);

        Collections.sort(cargoDimensions);
        Collections.sort(boxDimensions);

        boolean fits = true;
        for (int i = 0; i < 3; i++) {
            if (cargoDimensions.get(i) > boxDimensions.get(i)) {
                fits = false;
                break;
            }
        }

        if (fits) {
            this.cargo = cargo;
            this.isFull = true;
            System.out.println("Cargo placed in Box!");
            return true;
        } else {
            System.out.println("Box dimensions are too small for that cargo");
            return false;
        }
    }

    public void removeCargo() {
        this.cargo = null;
        this.isFull = false;
    }

    public String toString() {
        return "Box: \tDimensions: " + this.length + "x" + this.width + "x" + this.height + " Capacity: " + this.getCapacity()
            + " " + this.cargo;
    }
}

```

(4) Erweiterung der Klasse Book

Auf ILIAS (Übungen → Serie 3) finden Sie eine Datei `Book.java`. Ihre Aufgabe ist es die darin implementierte Klasse `Book` wie folgt zu erweitern:

1. Schreiben Sie einen Konstruktor sowie Getter und Setter für alle Instanzvariablen.
2. Implementieren Sie die Methode `age`, welche das Alter eines Buches (Anzahl Tage seit Erscheinungsdatum) berechnet und zurückgibt.
3. Implementieren Sie die Methode `toString`, die alle Informationen eines `Book`-Objekts als String zurückgibt. Beispiel: 123, Die Blechtrommel, Günter Grass, 1.1.1959
4. Vervollständigen Sie die Methode `input`, welche die Werte für `id`, `title`, `author` und `dateOfPublication` von der Kommandozeile vom Benutzer einliest und im jeweiligen `Book`-Objekt abspeichert. Ungültige Eingaben müssen Sie nicht abfangen.

Testen Sie sämtliche Methoden der Klasse `Book` in einer zusätzlichen Klasse `BookShelf`.

```

import java.util.Date;
import java.util.Scanner;
import java.text.*;

public class Book
{
    private int id;
    private String title;
    private String author;
    private Date dateOfPublication;

    public static final String DATE_FORMAT = "dd.MM.yyyy";

    public Book(int id, String title, String author, String dateOfPublication) {
        this.id = id;
        this.title = title;
        this.author = author;
        this.dateOfPublication = stringToDate(dateOfPublication);
    }

    public Book() {
    }
}

```

```

/** Returns the age of the book in days since publication */
public int age() {
    return (int) ((new Date().getTime() - dateOfPublication.getTime()) / (1000 * 60 * 60 * 24));
}

/** Returns a String representation of the book */
public String toString() {
    return id + ", " + title + ", " + author + ", " + dateToString(dateOfPublication);
}

/** Reads all book data from user input */
public void input() {
    Scanner scanner = new Scanner(System.in);
    System.out.print("ID: ");
    id = scanner.nextInt();
    scanner.nextLine(); // consume newline
    System.out.print("Title: ");
    title = scanner.nextLine();
    System.out.print("Author: ");
    author = scanner.nextLine();
    System.out.print("Date of publication: ");
    dateOfPublication = stringToDate(scanner.nextLine());
    scanner.close();
}

//--- Get-/Set-methods ---

//--- helper methods -- DO NOT CHANGE -----
/** Converts the Date object d into a String object */
public static String dateToString( Date d )
{
    SimpleDateFormat fmt = new SimpleDateFormat( DATE_FORMAT );
    return fmt.format( d );
}

/** Converts the String object s into a Date object */
public static Date stringToDate( String s )
{
    Date r = null;
    try {
        SimpleDateFormat fmt = new SimpleDateFormat( DATE_FORMAT );
        r = fmt.parse( s );
    } catch ( ParseException e ){
        System.err.println( e );
        System.exit(1);
    }
    return r;
}
}

```