

## Übungseinheit 20. - 24. November 2017

### Themen:

- Klasse String
- Klasse ArrayList
- null

### Aufgabe 1

Schreiben Sie eine Klasse Student mit dem Konstruktor

```
public Student(String name, String matNr, int note)
```

und den folgenden gib-Methoden:

```
public String gibName(),  
public String gibMatNr(),  
public int gibNote().
```

Verfassen Sie auch eine Klasse Uebungsgruppe mit dem Konstruktor

```
public Uebungsgruppe(),
```

der eine Übungsgruppe mit leerer Studentenliste erzeugt. Implementieren Sie auch die folgenden Methoden:

```
public boolean addStudent(Student einStudent)
```

fügt der Studentenliste einen neuen Studenten hinzu, wobei zu überprüfen ist, ob der Student in der Liste schon enthalten ist (liefert true, wenn das Hinzufügen erfolgreich ist, ansonsten false). Zwei Studenten gelten als gleich, wenn sie die gleiche Matrikelnummer haben. Die Methode

```
public ArrayList<Student> getNotenliste()
```

liefert eine ArrayList, die jeden Studenten mit positiver Note enthält.

### Aufgabe 2

Schreiben Sie eine Klasse Mitglied mit dem Konstruktor

```
public Mitglied(String name, String adresse, int bezahlt)
```

und den Methoden

```
public String gibName(),  
public String gibAdresse(),  
public int gibBezahlt() und  
public void setzeBezahlt(int betrag),
```

die die entsprechenden Informationen zurückgeben bzw. neu definieren. Schreiben Sie weiters eine Klasse Verein mit dem Konstruktor

```
public Verein(),
```

der einen Verein ohne Mitglieder erzeugt. Implementieren Sie auch die folgenden Methoden:

```
public boolean addMitglied(Mitglied einMitglied)
```

fügt einMitglied der Mitgliederliste des Vereins hinzu, wenn dieses Mitglied in der Mitgliederliste noch nicht enthalten ist (liefert true wenn das Hinzufügen erfolgreich ist, ansonsten false). Zwei Mitglieder gelten als gleich, wenn sie denselben Namen und dieselbe Adresse haben.

```
public ArrayList<Mitglied> getNichtBezahlt(int beitrags)
```

liefert eine ArrayList mit jenen Mitgliedern des Vereins zurück, die ihren Mitgliedsbeitrag noch nicht zur Gänze bezahlt haben (beitrags - bezahlt > 0).

### Aufgabe 3

Schreiben Sie eine Klasse Buch mit dem Konstruktor

```
public Buch(String titel, String autor)
```

und den Methoden

```
public String gibTitel(), public String gibAutor(),
```

die die entsprechenden Informationen zurück geben.

Schreiben Sie weiters eine Klasse Bibliothek mit dem Konstruktor

```
public Bibliothek(),
```

der eine leere Bibliothek ohne Bücher erzeugt. Implementieren Sie in der Klasse Bibliothek auch die folgenden Methoden:

```
public void addBuch(String titel, String autor)
```

fügt der Bibliothek ein neues Buch mit dem angegebenen Titel und Autor hinzu, das nicht entlehnt ist.

```
public Buch entlehne(String titel, String autor)
```

versucht, ein Buch aus der Bibliothek zu entleihen. Wenn es in der Bibliothek ein Buch mit diesem Titel und Autor gibt, das nicht entlehnt ist, dann wird dieses Buch entlehnt und die Methode liefert das Buch zurück. Ansonsten ist das Entleihen nicht möglich, und die Methode liefert null.

```
public boolean retuniere(Buch einBuch)
```

gibt ein entlehntes Buch in die Bibliothek zurück. Wenn das Buch nicht aus der Bibliothek entlehnt war, liefert die Methode `false`, ansonsten `true`.

*Hinweis:* Definieren sie in der Klasse `Buch` ein weiteres Attribut und entsprechende Zugriffsmethoden, das anzeigt, ob das Buch verliehen ist oder nicht. Ob es sich bei `b1` und `b2` um dasselbe Buch handelt, kann mittels `b1 == b2` überprüft werden. Beachten Sie auch, dass es zu einem Titel und Autor mehrere Exemplare des Buches in der Bibliothek geben kann.

## Aufgabe 4

Schreiben Sie eine Klasse `Beleg` mit dem Konstruktor

```
public Beleg(int nr, String artikel, String verkaeufer, double preis,
             int anzahl)
```

`nr` ist eine eindeutige Belegnummer; `artikel` enthält die Bezeichnung, unter der dieser Artikel verkauft wird; `verkaeufer` ist der Name des Verkäufers; `preis` ist der Verkaufspreis des Artikels pro Stück; `anzahl` gibt an, wieviele Artikel pro Beleg verkauft wurden.

Überlegen Sie, nachdem Sie die gesamte Aufgabenstellung verstanden haben, über welche Methoden und Attribute diese Klasse verfügen sollte!

Schreiben Sie weiters eine Klasse `Buchhaltung` mit dem Konstruktor

```
public Buchhaltung(),
```

der eine leere Liste von Belegen erzeugt. Implementieren Sie auch die folgenden Methoden:

```
public boolean buche(Beleg eintrag)
```

fügt den Beleg `eintrag` der `Buchhaltung` hinzu, wenn dieser Beleg in der `Buchhaltung` noch nicht enthalten ist (liefert `true` wenn das Hinzufügen erfolgreich ist, ansonsten `false`). Hinweis: Zwei Belege gelten als gleich, wenn sie dieselbe `nr` haben!

```
public ArrayList<Beleg> getVieleBilligeEintraege(int viele, double billig)
```

liefert eine `ArrayList` mit allen Belegen, bei denen die Stückzahl  $\geq$  `vielen` und der Verkaufspreis  $<$  `billig` ist, zurück.

```
public Beleg pruefeZufaelligen()
```

liefert einen zufälligen Beleg aus der Liste der Belege zurück. Falls kein Beleg existiert, soll `null` zurückgegeben werden.

```
public boolean storniereBeleg(int nr)
```

storniert den Beleg mit der Belegnummer `nr`. Wenn dieser Beleg nicht existiert oder schon storniert wurde, liefert die Methode `false` zurück. Andernfalls wird der Beleg als storniert vermerkt und `true` zurückgegeben.

```
public String findeBilligstenArtikel()
```

gibt den Namen eines Artikels zurück, der einen geringsten Stückpreis hat. Die Methode soll die leere Zeichenkette zurückgegeben, falls kein Beleg in der Buchhaltung existiert. (Haben mehrere Artikel den selben geringsten Stückpreis, kann aus dieser Menge ein beliebiger Artikel zurückgeliefert werden.)

## Aufgabe 5

Schreiben Sie die Klasse `Tagebuch` mit dem Konstruktor

```
public Tagebuch(String name)
```

der ein Tagebuch mit dem gegebenen Namen anlegt. Am Anfang enthält das Tagebuch keine Einträge. Schreiben sie ebenfalls die Methoden:

```
public void addEintrag(String eintrag)
```

fügt einen neuen Eintrag `eintrag` dem Tagebuch hinzu, wenn der Eintrag ungleich `null` ist und eine Mindestlänge von 10 Zeichen hat.

```
public boolean ersetzeEintrag(String alterEintrag,  
                             String neuerEintrag)
```

ersetzt den existierenden Tagebucheintrag `alterEintrag` mit `neuerEintrag` und liefert `true` zurück. Falls es keinen Tagebucheintrag mit `alterEintrag` gibt, dann liefert die Methode `false` zurück und das Tagebuch wird nicht geändert.

```
public double durchschnittsLaengeEintraege()
```

berechnet die Durchschnittslänge der Tagebucheinträge.

```
public ArrayList<String> gibLetzteEintraege(int n)
```

liefert die letzten `n` hinzugefügten Einträge im Tagebuch zurück. Wenn weniger als `n` Tagebucheinträge enthalten sind, dann sollen alle Einträge zurückgeliefert werden. Falls es keine Einträge gibt, dann liefert die Methode eine leere Liste zurück.

Sie können den jeweiligen Klassen weitere Methoden und/oder Attribute hinzufügen, wenn Ihnen das sinnvoll oder notwendig erscheint.

## Aufgabe 6

Schreiben Sie eine Klasse `Heizungssteuerung` mit folgenden Eigenschaften: Es gibt einen Konstruktor

```
public Heizungssteuerung(int minTemp, int maxTemp),
```

der eine Heizungssteuerung erzeugt, die eine Heizung zwischen einer minimalen und einer maximalen Temperatur steuert. Zu Beginn soll die Temperatur auf die Minimaltemperatur gestellt werden. Die Methoden

```
public void waermer(),
```

```
public void kuehler()
```

erhöhen bzw. verringern die Temperatur um eine voreingestellte Schrittweite. Die Schrittweite ist zu Beginn auf 5 gesetzt. Wenn durch das Erhöhen der Temperatur die Maximaltemperatur

überschritten würde, soll die Temperatur auf den Maximalwert gesetzt werden. Analoges gilt für das Verringern der Temperatur. Die Methode

```
public int getTemperatur()
```

liefert die aktuell eingestellte Temperatur. Mit der Methode

```
public void setSchrittweite(int schrittweite)
```

kann die Schrittweite auf einen neuen Wert gesetzt werden. Stellen Sie sicher, dass die Schrittweite nur verändert wird, wenn der übergebene Wert `schrittweite` größer als 0 ist.