

1 Klassen und Objekte

Klasse mit Attributen und Konstruktor

Schreiben Sie eine neue Klasse *Person*, welche eine Person bei einem Gesundheitscheck widerspiegelt. Jede Person soll folgende Felder haben, welche tlw. über Konstruktor-Parameter gesetzt und über Getter ausgelesen werden können. Details dazu finden sich in der nachfolgenden Tabelle.

Attribut-Name	Beschreibung	Über Konstruktor <u>parameter</u> -setzen	Getter	Setter
vorname (String)	Vorname	JA	JA	NEIN
nachname (String)	Nachname	JA	JA	NEIN
koerpergroesse (int)	Körpergröße in cm	JA	JA	JA
aktivaetslevel (byte)	Gibt an, wie sportlich aktiv eine Person ist. Werte zwischen 1 und 4 sind zulässig	JA	JA	JA

public String rechneBMI(int gewicht)

Diese Methode berechnet den BMI anhand der Körpergröße und dem Gewicht anhand folgender Formel und gibt eine Gewichtskategorisierung zurück.

$$\text{bmi} = \text{gewicht} / ((\text{koerpergroesse} / 100.0) * (\text{koerpergroesse} / 100.0))$$

bmi	Rückgabe
< 18.5	„Untergewicht“
18.5 bis 25	„Normalgewicht“
25 bis 30	„Übergewicht“
> 30	„Adipositas“

public void printPerson(int gewicht)

Diese Methode gibt die Person mit ihrem Aktivitätslevel und BMI-Kategorisierung aus. Geben Sie für das Aktivitätslevel folgende Werte aus.

1 ... *Couch Potato*

2 ... *Gelegenheitssportler*

3 ... *Sportler*

4 ... *Profisportler*

Verknüpfen Sie den Namen, das Aktivitätslevel und die BMI-Kategorisierung zu folgender Ausgabe:

Max Mustermann, Sportler, Normalgewicht

Anmerkung: Probieren Sie diese Ausgabe alternativ auch mittels switch-case-Anweisung. Details dazu finden Sie im Folien-Satz zu Generics.

Demo-Klasse

Implementieren Sie auch eine Demo-Klasse, welche zeigt, dass die Klasse *Person* funktioniert.

2 Schleifen und Arrays

Erzeugen Sie eine neue Klasse mit dem Namen *LottoZahlenApp* und erstellen Sie darin eine main-Methode. Schreiben Sie eine Methode *public static int[] generateRandomNumbers(int size, int maxNumber)*. Die Methode soll einen int-Array mit der Länge von size erstellen und den Array mit Zufallszahlen zwischen 1 und *maxNumber* befüllen.

Eine Zufallszahl zwischen 1 und maxNumber kann wie folgt erstellt werden:

```
int number = (int) (Math.random() * maxNumber + 1);
```

Tipp:

- 1) Erstellen Sie den Array
- 2) Erstellen Sie eine for-Schleife, welche bis zu size läuft
- 3) Generieren Sie eine Zufallszahl und schreiben Sie diese in den Array auf die richtige Position
- 4) Retournieren Sie den Array

Rufen Sie in der main-Methode die Methode *generateRandomNumbers* mit den Parameter size=6 und maxNumber=45 auf und geben Sie die erstellten Zahlen auf der Konsole aus.

3 Collections

Es soll eine Einkaufsliste abgebildet werden. Eine Einkaufsliste besteht aus verschiedenen Einkäufen.

Erzeugen Sie dazu eine neue Klasse mit dem Namen Einkauf (Attribute siehe Tabelle). Erstellen Sie eine Klasse EinkaufslisteApp und darin die main-Methode.

Attribut-Name	Beschreibung	Über Konstruktor <u>parameter</u> -setzen	Getter	Setter
artikelname (String)	Der Name des gekauften Artikels	JA	JA	NEIN
preis (double)	Der Preis des gekauften Artikels	JA	JA	NEIN

Hinweis: equals(), hashCode() werden benötigt

In der Klasse „EinkaufslisteApp“ erstellen Sie folgende Methoden:

public static double sum(ArrayList<Einkauf> einkaufsliste)

Diese Methode soll den Gesamtpreis der Einkäufe retournieren.

zB: Die Einkaufsliste enthält 2 Artikel mit den Preisen 10.00 und 15.00 => 25.00 soll retourniert werden.

public static double sum(HashSet<Einkauf> einkaufsliste)

Berechnen Sie die Summe (Gesamtpreis) aller einzigartigen (unique) Artikel und geben Sie den Wert zurück

public static HashSet<Einkauf> unique(ArrayList<Einkauf> einkaufsliste)

Da eine Einkaufsliste mehrere gleiche Artikel enthalten kann, soll diese Methode alle einzelnen Artikel liefern, d.h. jeder Artikel soll nur einmal vorkommen.

Erstellen Sie in der main-Methode eine Einkaufsliste und fügen Sie der Einkaufsliste mehrere Artikel (gerne auch gleiche) hinzu. Rufen Sie dann die Methoden auf und geben Sie das Ergebnis auf der Konsole aus.