

Methoden, Klassen, Vererbung

Aufgabe 1

Schaut die Wiederholungsfolien an, überlegt euch Fragen zu all dem was euch nicht 100%ig klar ist und fragt in der nächsten Vorlesung oder sprecht mich einfach an.

Aufgabe 2 - Nachrichten

Schreibt eine Klasse **Message** mit der Nachricht als String, dem Empfänger und dem Absender (Objektvariablen). Diese Variablen sollen *private* sein. Schreibt einen Konstruktor, der als Parameter ebenfalls die Nachricht, den Empfänger und den Absender erwartet.

Damit man von außen auf die Daten wieder zugreifen kann, schreibt für alle Objektvariablen Getter-Methoden.

Überschreibt die `toString()`-Methode: *public String toString ()*

Legt in einer anderen Klasse zwei Nachrichten an und gebt diese auf der Konsole aus.

Aufgabe 3

Programmiert entweder Aufgabe 3a oder 3b.

a) Hochschule

Schreibt eine Klasse **Person**, die einen *Vor-* und einen *Nachnamen* hat. Im Konstruktor sollen der Vor- und der Nachnamen erwartet werden. Erstellt Getter für beide Variablen und überschreibt die `toString()`-Methode.

Erstellt eine Klasse **Student**, die von der Klasse **Person** erbt. Ein Student soll zusätzlich eine *Matrikelnummer (String)* und eine *Kennung (String)* haben. Der Student soll im Konstruktor den Vor- und Nachnamen, die Matrikelnummer und die Kennung übergeben bekommen. Schreibt Getter für die Matrikelnummer und die Kennung.

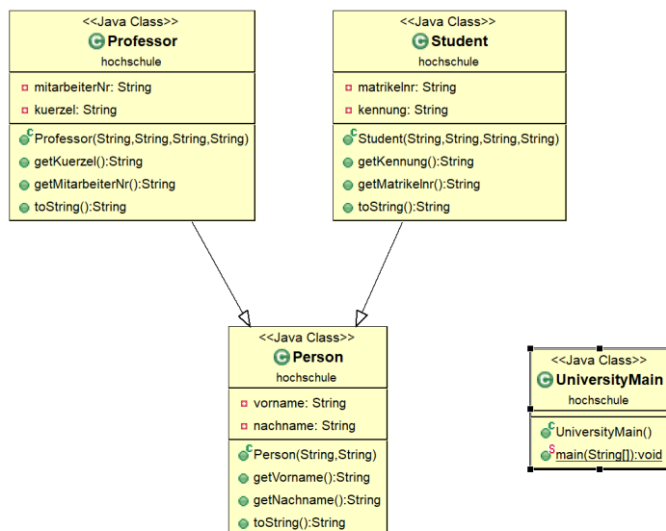
Überschreibt auch hier die `toString`-Methode und gebt alles aus.

Erstellt eine Klasse **Professor**, die von der Klasse **Person** erbt. Ein Prof soll zusätzlich eine *Mitarbeiternummer (String)* und ein *Kürzel (String)* haben. Der Prof soll im Konstruktor den Vor- und Nachnamen, die Mitarbeiternummer und das Kürzel übergeben bekommen. Schreibt Getter für die Mitarbeiternummer und das Kürzel.

Überschreibt auch hier die `toString`-Methode und gebt alles aus.

Erstellt nun in einer `main`-Methode ein Array von *Person* (oder eine `ArrayList<Person>` s.u.), in das ihr Studierende und Professoren speichert.

Geht danach das Array (die `Arrayliste`) durch und fragt jedes Objekt, ob es eine Instanz von Prof oder Student ist *if (personen [i] instanceof Student)* und gebt die Objekte auf der Konsole aus.



b) Sport

Schreibt eine Klasse **Person**, die einen *Vor-* und einen *Nachnamen* hat. Im Konstruktor soll der Vor- und der Nachnamen erwartet werden. Erstellt Getter für beide Variablen.

Erstellt eine Klasse **Spieler**, die von der Klasse **Person** erbt. Ein Spieler soll zusätzlich eine *Position (String)* und eine *Variable anzahlTore (int)* haben. Der Spieler soll im Konstruktor den Vor- und Nachnamen übergeben bekommen. Schreibt Getter für die Anzahl der Tore und die Position, sowie eine *schießtTor()*-Methode, die die Anzahl der geschossenen Tore um 1 erhöht. Schreibt eine Setter-Methode für die Position.

Überschreibt auch hier die *toString*-Methode und gibt alles aus.

Erstellt eine Klasse **Trainer**, die ebenfalls von der Klasse **Person** erbt. Ein Trainer soll zusätzlich eine *Mitgliedsnummer (int)* haben. Der Trainer soll zwei Konstruktoren haben, einen der den Vor- und Nachnamen fordert und beim zweiten Konstruktor soll zusätzlich die Mitgliedsnummer angegeben werden.

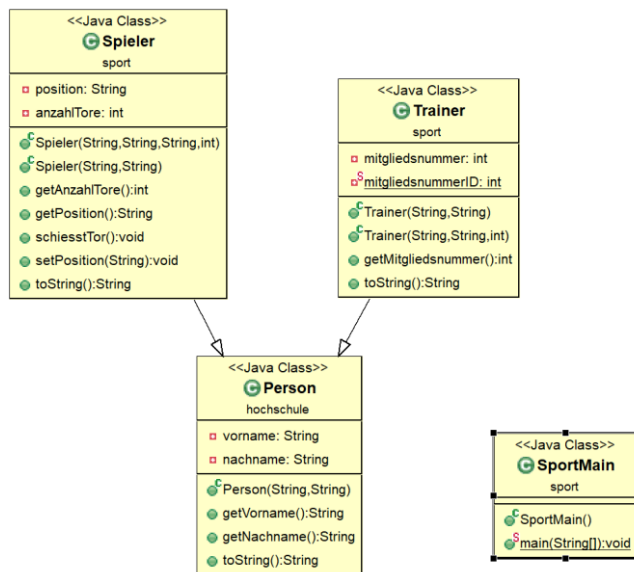
Wird der Konstruktor mit Vor- und Nachnamen aufgerufen, soll intern eine Mitgliedsnummer generiert werden. Dafür braucht die Klasse eine Klassenvariable *ID*, die automatisch hochzählt und die Mitgliedsnummer entsprechend setzt.

Schreibt einen Getter für die Mitgliedsnummer.

Überschreibt auch hier die *toString*-Methode und gibt alles aus.

Erstellt nun in einer *main*-Methode ein Array von **Person** (oder eine `ArrayList<Person>` s.u.), in das ihr **Trainer** und **Spieler** speichert. Geht danach das Array (die `ArrayListe`) durch und fragt jedes Objekt, ob es eine Instanz von **Trainer** oder **Spieler** ist und gibt die Objekte auf der Konsole aus.

Tipp: Auch hier hilft euch *instanceof* weiter.



TIPP

Ihr könnt zum Lösen wie gewohnt ein Array verwenden oder ihr probiert eine ArrayListe aus.

Array

```
Person [] persons = new Person [50];
```

```
persons [i] = new Person („Walter“, „Franke“);
```

```
persons.length
```

```
persons [i]
```

ArrayListe

```
ArrayList<Person> persons = new ArrayList<>();
```

```
persons.add(new Person („Walter“, „Franke“));
```

```
persons.size()
```

```
persons.get(i)
```

Dazu kann man abfragen, ob in der Liste ein bestimmtes Objekt enthalten ist:

```
if (persons.contains(personX))
```

Have Fun ☺