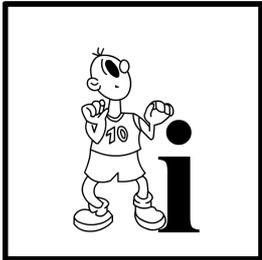


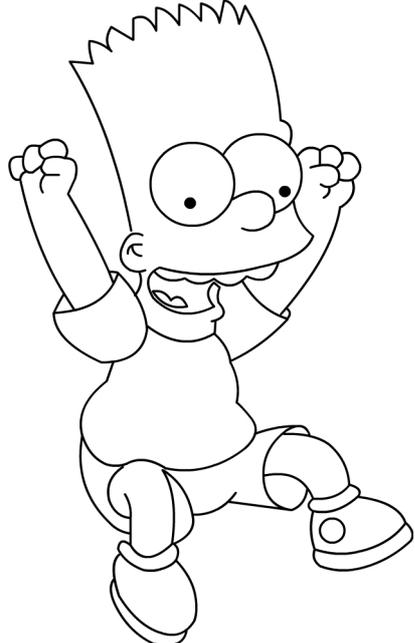


Idee, Zweck, Systembeschreibung:



Für den Sportunterricht brauchen Personen einen *Body-Mass-Index-Rechner (BMIREchner)*. Hierfür soll es möglich sein, auf einer *Benutzeroberfläche (Hauptfenster)* nach der Eingabe des *Körpergewichts (in kg)* und der *Körpergröße (in m)*, den *bmi (Body-Mass-Index)* einer *Person zu berechnen*. Die Berechnung soll erfolgen, wenn man die Schaltfläche „*BMI berechnen*“, *bedient (anklickt)*. Eine weitere Schaltfläche *Eingaben löschen* soll dafür sorgen, dass alle gemachten Eingaben *gelöscht* werden. Der *berechnete BMI (bmi)* soll auf der *Benutzeroberfläche (Hauptfenster)* *angezeigt* werden. In der erweiterten Version soll zudem die Möglichkeit bestehen durch Eingabe des *Geschlechts (männlich oder weiblich)* und des *Alters (in Jahren)*, das *Intervall [BMI Min, BMI Max]* und die daraus abgeleitete *Klassifikation zu bestimmen*. Beide Angaben (*Intervall und Klassifikation*) sollen auf der *Benutzeroberfläche (Hauptfenster)* *angezeigt* werden, wenn man die Schaltfläche *BMI interpretieren anklickt*.

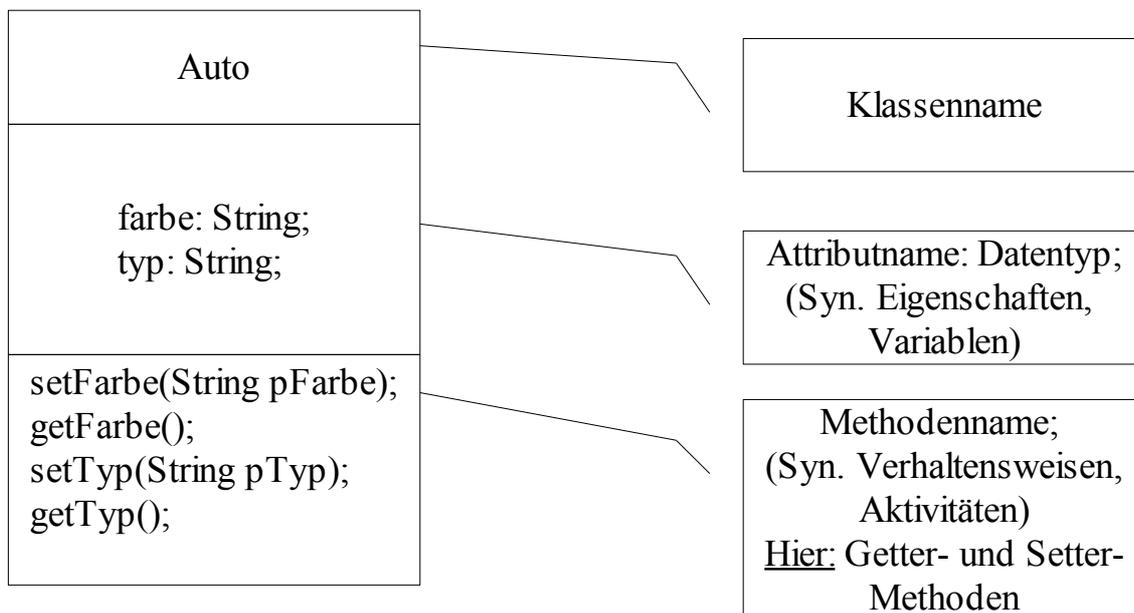
Anwendungsfälle:

		
<i>Myra Bellamy ist normalgewichtig [22,27]. Ihr Körpergewicht beträgt 58 kg und Ihre Körpergröße beträgt 1.69 m. Sie ist 48 Jahre alt.</i>	<i>Red Barklay ist übergewichtig [21,26]. Sein Körpergewicht beträgt 90 kg und Ihre Körpergröße beträgt 1.75 m. Er ist 38 Jahre alt.</i>	<i>Bart Simpson ist untergewichtig [19,24]. Sein Körpergewicht beträgt 45 kg und Ihre Körpergröße beträgt 1.68 m. Er ist 19 Jahre alt.</i>

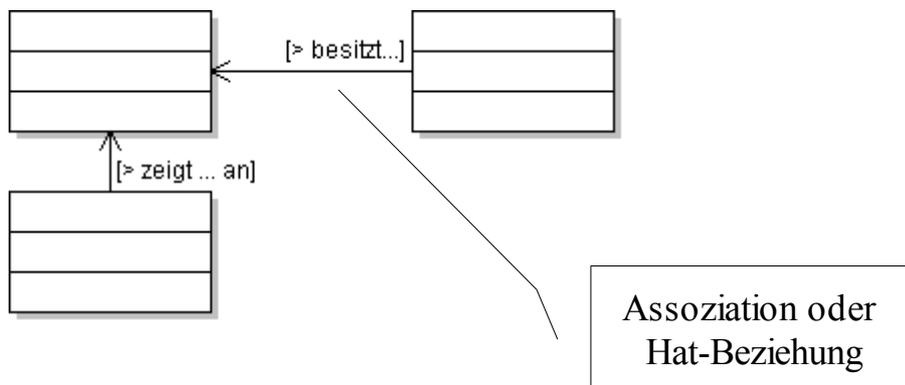


Merke: Sogenannte Getter- und Setter-Methoden¹ (Syn. Aktivität, Verhaltensweise) existieren für jede Eigenschaft. Z.B. Die Methoden `setFarbe(String pFarbe)` und `getFarbe()` für das Attribut (syn. Variable, Eigenschaft) `String farbe` der Klasse `Auto`. Diese Methoden dienen dazu, Eigenschaftswerte einzelner Objekte zu modifizieren (bearbeiten, ändern) bzw. erstmalig zu initialisieren (Wert setzen). Es sind quasi Teilhandlungen auf unterster Ebene (Hinweis: kleinschrittig denken).

Zur Erinnerung die UML-Notation einer Klasse:



Vorschlag für die System-Architektur des Body-Mass-Index-Rechners:



Hinweis: Ohne diese Verbinder (Pfeile) können die Klassen nicht miteinander kommunizieren.

¹ „get“ steht für „holen“, „set“ steht für „setzen“



BMI-Formel:

$$\text{bmi} = \text{gewicht}/(\text{groesse}^2) = \text{gewicht}/(\text{groesse}*\text{groesse})$$

Intervall-Tabelle:

Bedingungen für die <i>Methode</i> bestimmeMinUndMax(){...}		
Altersintervall	BMI Minimum	BMI Maximum
[19;24] 19 <= Alter <= 24	19	24
]24;34] Alter <= 34	20	25
]34;44] Alter <= 44	21	26
]44;54] Alter <= 54	22	27
]54;64] Alter <= 64	23	28
]64;∞[Alter > 64	24	29

Klassifikationstabelle:

Bedingungen für die <i>Methode</i> bestimmeKlassifikation(){...}		
Klassifikation	männlich	weiblich
Untergewicht	BMI < 20	BMI < 19
Normalgewicht	20-25	19-24
Übergewicht	25-30	24-30
Adipositas	30-40	30-40
Schwere Adipositas	BMI > 40	BMI > 40



Kontrollstrukturen: Fallunterscheidung mit ELSEIF

Bisher haben wir immer nur zwei Fälle unterschieden (erfolgreich oder nicht erfolgreich). Im Falle der Bestimmung des Minimums und Maximums oder der Klassifikation unterscheiden wir X-Fälle (mehrere). Hierfür bietet sich eine neue Kontrollstruktur an, die ELSEIF-Anweisung. Ein Beispiel soll die Verwendung verdeutlichen:

Gewinnkategorie	Min	Max
hoch	20000	> 20000
mittel	10000	19999
niedrig	9999	0

```
public void wahlDerGewinnkategorie(int pGewinn) {
    int mGe = pGewinn;
    if(mGe >= 20000) {
        this.setGewinnKategorie("hoch");
        System.out.println("Gewinnkategorie: hoch");
    } else if(mGe >= 10000 && mGe <= 19999) {
        this.setGewinnKategorie("mittel");
        System.out.println("Gewinnkategorie: mittel");
    } else {
        this.setGewinnKategorie("niedrig");
        System.out.println("Gewinnkategorie: niedrig")
    }
}
```

So könnte die grafische Benutzeroberfläche aussehen:

The screenshot shows a web application for calculating the Body Mass Index (BMI). On the left, there is a cartoon mascot of a white character with a soccer ball. Below the mascot, there is a logo for 'Kaufmännische Schule Wangen' with the tagline 'Wege zeigen, öffnen, gehen'. The main content area is titled 'Eingaben für den Body-Maß-Index:' and contains three input fields: 'Ihr Gewicht (...)' with a text input 'tfGewicht', 'Ihre Größe (m)' with a text input 'tfGroesse', and 'Ihr BMI beträgt' with a text input '<Ergebnis>'. To the right of these inputs are two buttons: 'BMI berechnen' and 'Eingaben löschen'. Below this section, there is another section titled 'Für die Interpretation werden weiter Eingaben benötigt:'. It contains two input fields: 'Ihr Geschlecht' with a dropdown menu showing 'männlich' and 'BMI interpretieren' button, and 'Ihr Alter' with a text input 'tfAlter'. The final output field is 'Ihre Interpretation' with a text input '<Interpretation>'.



Behandlung von Eingaben und Ausgaben der Hauptfenster-Klasse (GUI): Konvertierung in andere Datentypen

Heißt umgangssprachlich „umwandeln“. Da die Werte, die über die Benutzeroberfläche eingegeben werden vom Datentyp String sind müssen Sie umgewandelt werden, um damit Rechnen zu können. Umgekehrt können auf der Benutzeroberfläche nur Strings angezeigt werden.

Man kann einen String (z.B. eine Eingabe über die Benutzeroberfläche) in einen Double umwandeln oder umgekehrt.

Beispiel für die Hauptfensterklasse-Klasse (GUI):

```
//Assoziation (Objekt)
Person person = new Person();

/*Eingabe:
 *Beispiel GUI-Klasse - String in Double: Wandelt den String aus dem
 *Textfeld tfGewicht in den Datentyp double und setzt den Wert
 *in der Klasse Person*/

public void stringInDouble() {
    person.setGewicht(Double.valueOf(tfGewicht.getText()));
}

/*Ausgabe:
 *Beispiel GUI-Klasse - Double in String: Wandelt das Attribut double
 *gewicht in einen String und zeigt diesen im Textfeld
 *tfGewicht auf der Benutzeroberflaeche an*/

public void doubleInString() {
    tfgewicht.setText(String.valueOf(person.getGewicht()));
}
```