# VL08-Aufgabe 1 (Übung) (☆)

Beantworten Sie folgende Fragen:

- Was ist ein Objekt?
- Was ist eine Klasse?
- Worin besteht der Unterschied zwischen einer Klasse und einem Objekt?
- Wie greift man auf ein Attribut bzw. eine Methode eines Objektes zu?
- Was ist die „garbage collection“ und wozu dient diese?

# VL08-Aufgabe 2 (Übung) (☆☆)

Gegeben sind die Klassen `Person` und `PersonTest`.

```
class Person
{
    String vorname, nachname;
    int steuerklasse = 1;

    Person(String vorname, String nachname)
    {
        this.vorname = vorname;
        this.nachname = nachname;
    }

    void setVorname(String vorname)
    {
        this.vorname = vorname;
    }

    void setNachname(String nachname)
    {
        this.nachname = nachname;
    }

    void setSteuerklasse(int steuerklasse)
    {
        this.steuerklasse = steuerklasse;
    }

    String getVorname()
    {
        return this.vorname;
    }

    String getNachname()
    {
        return this.nachname;
    }

    int getSteuerklasse()
    {
        return this.steuerklasse;
    }
}
```

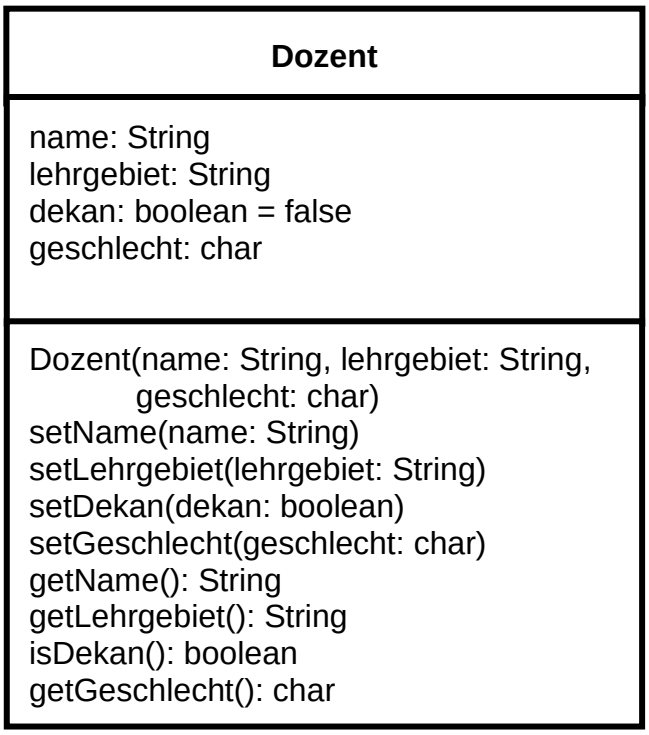
```
class PersonTest
{

    public static void main(String[] args)
    {

1       Person erstePerson = new Person("Dagobert","Reich");
2       Person zweitePerson = new Person("Gustav", "Lebenskünstler");
3       erstePerson.setNachname("Reicher");
4       zweitePerson = new Person("Donald","Arm");
5       zweitePerson.setSteuerklasse(3);
6       System.out.println(erstePerson.getVorname());
7       System.out.println(erstePerson.getNachname());
8       System.out.println(erstePerson.getSteuerklasse());
9       System.out.println(zweitePerson.getVorname());
10      System.out.println(zweitePerson.getNachname());
11      System.out.println(zweitePerson.getSteuerklasse());
    }
}
```

a) Zeichnen Sie je ein Bild des Hauptspeichers (entsprechend VL08) nach der Anweisung 2, während der Ausführung von Anweisung 3 vor Verlassen der Methode `setNachname` und nach Anweisung 5.

b) Welche Bildschirmausgaben erzeugen die Anweisungen 6-11?

# VL08-Aufgabe 3 (Praktikum) (☆☆)

1. Erstellen Sie in Eclipse ein neues Projekt mit dem Namen `EidP-VL08Aufgabe3`.
2. Fügen Sie dem Projekt eine Klasse `Dozent` hinzu und übertragen Sie die UML-Notation der Klasse `Dozent` mit Attributen, Konstruktor und Methoden in eine Java-Klasse.




3. Fügen Sie dem Projekt eine zweite Klasse `DozentTest` mit einem Hauptprogramm (main-Methode) mit folgenden Eigenschaften hinzu:

   - Die beiden Referenzvariablen `dieDozentin` und `derDekan` sollen deklariert werden.
   - Die beiden Objekte sollen erzeugt werden.
   - Alle Attribute des Dekans sollen durch Methodenaufruf gelesen und auf der Konsole ausgegeben werden.
   - Das Lehrgebiet der Dozentin soll durch Methodenaufruf geändert werden.

# VL08-Aufgabe 4 (Praktikum) (☆☆☆-☆☆☆☆)

1. Erstellen Sie in Eclipse ein neues Projekt mit dem Namen EidP-UEB08-Aufgabe4.
2. Legen Sie eine Klasse Punkt2D an. Instanzen dieser Klasse repräsentieren einen Punkt in einem zweidimensionalen Koordinatensystem. Die Objektattribute sind `double x` und `double y`. Schreiben Sie einen Konstruktor, der beide Objektattribute setzt, sowie Getter- und Setter-Methoden für x und y.
3. Legen Sie eine Klasse Rechteck an. Instanzen dieser Klasse repräsentieren ein Rechteck in einem zweidimensionalen Koordinatensystem. Die Attribute sind die linke untere Ecke als Punkt2D sowie die Seitenlängen in x- und y-Richtung. Implementieren Sie einen Konstruktor

   `Rechteck(`**`double`** `x,` **`double`** `y,` **`double`** `seitenlaengeX,` **`double`** `seitenlaengeY)`

   sowie Getter- und Setter-Methoden für die Seitenlängen.
4. Ergänzen Sie Methoden, die die vier Ecken des Rechtecks zurückgeben:

   ```
   Punkt2D getEckeLinksUnten()
   Punkt2D getEckeLinksOben()
   Punkt2D getEckeRechtsUnten()
   Punkt2D getEckeRechtsOben()
   ```

5. Legen Sie eine weitere Klasse RechteckTest an, die die folgende main-Methode enthält:

   ```
   public static void main(String[] args) {
      Rechteck r1 = new Rechteck(0,0,7,5);
      System.out.println(r1);
      Rechteck r2 = new Rechteck(6,4,2,2);
      System.out.println(r2);
      Rechteck r3 = new Rechteck(-1,2,9,2);
      System.out.println(r3);
      Rechteck r4 = new Rechteck(-1,-1,2,2);
      System.out.println(r4);
      Rechteck r5 = new Rechteck(5,-1,7,3);
      System.out.println(r5);
   }
   ```

6. Legen Sie in der Klasse Rechteck eine Methode `public String toString()` an (warum das Schlüsselwort `public` hier nötig ist, lernen wir in der nächsten Vorlesung), so dass die Ausführung der Main-Methode von RechteckTest die folgende Ausgabe erzeugt:

   ```
   Rechteck mit Seitenlängen 7.0 und 5.0 an der Stelle (0.0,0.0)
   Rechteck mit Seitenlängen 2.0 und 2.0 an der Stelle (6.0,4.0)
   Rechteck mit Seitenlängen 9.0 und 2.0 an der Stelle (-1.0,2.0)
   Rechteck mit Seitenlängen 2.0 und 2.0 an der Stelle (-1.0,-1.0)
   Rechteck mit Seitenlängen 7.0 und 3.0 an der Stelle (5.0,-1.0)
   ```

7. Ergänzen Sie die Klasse Rechteck um eine Methode (☆☆☆☆)
   mit dem Methodenkopf

   ```
   public boolean schneidet (Rechteck R)
   ```

   die genau dann den Wert true zurückliefert, wenn das „aufrufende Rechteck“ eine nichtleere Schnittmenge mit dem übergebenen Rechteck R hat.

   Um zu ermitteln, ob zwei Rechtecke einen nichtleeren Schnitt haben, gibt es verschiedene Möglichkeiten.

   Sie können beispielsweise überlegen, welche Bedingung für eine der vier Kanten des Rechtecks R gegeben sein muss, damit dieses einen nichtleeren Schnitt mit dem „aufrufenden Rechteck“ hat.

   Die folgenden Fragen helfen Ihnen vielleicht auch, zu verstehen, wie Sie ermitteln können, ob zwei Rechtecke einen nichtleeren Schnitt haben:

   1. Wenn sie nicht leer ist, ist die Schnittmenge zweier Rechtecke wieder ein Rechteck. Wie können Sie die Koordinaten der linken unteren Ecke dieses Rechtecks bestimmen?
   2. Wie können Sie die Koordinaten der rechten oberen Ecke einer Schnittmenge bestimmen?
   3. Wie können Sie prüfen, ob diese Koordinaten tatsächlich ein Rechteck definieren?

8. Um die Funktion der Methode schneidet() zu testen, ergänzen Sie die main-Methode der Klasse RechteckTest um die Zeilen

   ```
   System.out.print(r1.schneidet(r2) + " ");
   System.out.print(r1.schneidet(r3) + " ");
   System.out.print(r1.schneidet(r4) + " ");
   System.out.print(r1.schneidet(r5) + " ");
   System.out.println(r4.schneidet(r2));
   ```

   und prüfen Sie, ob die Ausführung dieser Zeilen die folgende Ausgabe bewirkt:

   ```
   true true true true false
   ```