



# Programmierung für Mathematik HS10

---

## Übung 6

### 1 Aufgabe: Vererbung

#### 1.1 Lernziele

1. Vererbung verstehen und korrekt implementieren können.
2. Methoden überladen können.

#### 1.2 Aufgabenstellung

Passen Sie Ihre Klassen von der Übung 5, Aufgabe 2 so an bzw. ergänzen Sie neue Klassen, dass folgende Problemstellung gelöst wird. Versuchen sie dabei Codeduplizität zu vermeiden. (Tipp: verwenden Sie Vererbung.)

Eine Universität verwendet Ihr Programm um die Studenten zu speichern. Jetzt will die Universität neuerdings auch das Personal speichern, diese haben natürlich auch einen Namen und ein Geburtsjahr und zusätzlich noch einen Lohn. Die Universität soll die Möglichkeit haben eine Liste mit diesen Angaben des Personals zu drucken. Ausserdem sollen von der Sekretärin wichtige Mitteilungen versendet werden können, welche von dem Personal und den Studenten gleich empfangen werden sollen (wenn die Mitteilung erhalten wurde, soll jeweils von jeder Person der Univerität eine kurze Meldung ausgegeben werden, dass die Nachricht angekommen ist). Um einen besseren Überblick über die Finanzen der Universität zu haben, soll es die Möglichkeit geben, die Summe aller Löhne der Mitarbeiter zu bekommen.

Testen Sie ihr neues Programm mit einem `TestDriver`.

## 2 Aufgabe: Chain of Words

### 2.1 Lernziele

1. Das Wissen über Referenzen vertiefen.
2. Die Verwendung von Loops wiederholen, insbesondere den Unterschied zwischen `while`- und `for`-Schleifen vertiefen.
3. `String`-Operationen wiederholen.

#### a) Aufgabenstellung: Die Elemente

1. Erstellen Sie eine Klasse namens `WordElement`.
2. Definieren Sie einen Konstruktor, welcher einen `String` als Parameter erwartet.
3. Deklarieren Sie nun eine private Instanzvariable des Typs `WordElement` und benennen sie diese mit `nextElement`.
4. Machen Sie dasselbe für eine zweite Variable namens `previousElement`.
5. Implementieren Sie Getter- und Setter-Methoden.
6. Fügen Sie desweiteren zwei Methoden hinzu, welche überprüfen, ob die jeweiligen Instanzvariablen mit einem Wert initialisiert wurden.

#### b) Aufgabenstellung: Die Kette

Im vorherigen Teil haben wir den Grundstein für unsere Kette gelegt: Die einzelnen Elemente verfügen über die Methoden, um sich aneinander zu ketten.

1. Erstellen Sie eine Klasse namens `Chain`.
2. Definieren Sie für diese Klasse eine private Instanzvariable namens `firstElement`.
3. Desweiteren muss eine Methode `addWord(String word)` definiert werden, welche keinen Wert zurückliefert. Diese Methode erstellt bei Aufruf jeweils eine neue Instanz der Klasse `WordElement` und übergibt dieser gleich das erhaltene Wort. Ist `firstElement == null`, so wird diese Variable mit dem neuen Objekt initialisiert. Ist dies nicht der Fall, so muss nun über eine `while`-Schleife das letzte Element der Wortkette ermittelt werden und danach das Objekt angefügt werden.
4. Erweitern Sie die Klasse dahingehend, dass eine Methode namens `getWordCount()` die Anzahl Wörter der Kette zurückgibt.
5. Ebenfalls muss eine Methode namens `getSentence()` erstellt werden, welche die gesamte Kette als `String` (mit Leerzeichen zwischen den Wörtern!) zurückgibt.
6. Erstellen Sie nun eine Methode `getReverse()`, welche bei folgendem Input:

```
1 Chain myChain = new Chain();
2 myChain.addWord("Anna");
3 myChain.addWord("liebt");
4 myChain.addWord("Thomas");
```

diesen Output in Form eines Strings zurückliefert: `Thomas liebt Anna`.

7. Erstellen Sie eine Methode namens `getEverySecond()` welche zuerst eine lokale Variable mit der aktuellen Anzahl Wörter initialisiert und danach mit einer `for`-Schleife jedes zweite Wort zu einem String zusammenhängt. z.B. Die Wörter `Ich lebe und arbeite in der Schweiz` sollten wie folgt ausgegeben werden: `lebe arbeite der`.
8. Testen Sie die Funktionalität in einem `TestDriver`.

### 3 Aufgabe: Lineare Gleichungssysteme

#### 3.1 Lernziele

1. Gauss-Eliminationsverfahren implementieren können.
2. Selbstständig Ideen entwickeln um ein mathematisches Problem mit Java lösen zu können.

#### 3.2 Aufgabenstellung

Schreiben Sie ein Programm welches das folgende lineare Gleichungssystem (in den unbekanntem  $x_i, i \in \{1, \dots, n\}$ ) lösen soll.

$$\begin{aligned}\alpha_{11} \cdot x_1 + \alpha_{12} \cdot x_2 + \dots + \alpha_{1n} \cdot x_n &= \beta_1 \\ \alpha_{21} \cdot x_1 + \alpha_{22} \cdot x_2 + \dots + \alpha_{2n} \cdot x_n &= \beta_2 \\ &\vdots \\ \alpha_{n1} \cdot x_1 + \alpha_{n2} \cdot x_2 + \dots + \alpha_{nn} \cdot x_n &= \beta_n\end{aligned}$$

oder in Matrixschreibweise:

$$\begin{pmatrix} \alpha_{11} & \alpha_{12} & \cdots & \alpha_{1n} \\ \alpha_{21} & \alpha_{22} & \cdots & \alpha_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{n1} & \alpha_{n2} & \cdots & \alpha_{nn} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{pmatrix}$$

Es darf angenommen werden, dass das Gleichungssystem genau eine Lösung besitzt. Verwenden Sie das Gauss-Eliminationsverfahren um den Lösungsvektor zu bekommen.

Falls Sie dazu Fragen haben, können Sie diese im OLAT-Forum stellen.