# Zusatztestat der Vorlesung Programmierung

Dieses Testat besteht aus der Schnittstelle **Person** sowie den Klassen **Student**, **Operation**, **LinkedList**, **ListItem** und **Main**. Die Schnittstelle **Person** und die Klasse **ListItem** sollen sie nicht verändern. Die Klasse **Main** enthält eine main-Methode mit der Sie Ihre Implementierung testen können.

**Aufgabe 1 – Klassen, Variablen und Methoden** (40 Punkte)

Erweitern Sie die Klasse **Student** wie folgt:

- Fügen Sie der Klasse zwei private Instanzvariablen hinzu: "name" vom Typ **String** und "id" vom Typ **long**.
- Implementieren Sie die Schnittstelle **Person**, indem Sie den Inhalt der entsprechenden Instanzvariablen in den dafür vorgesehenen Methoden zurückgeben.
- Implementieren Sie einen öffentlichen Konstruktor mit einem Parameter "name" und einem Parameter "id" (in dieser Reihenfolge), der die Instanzvariablen mit dem selben Namen initialisiert.
- Überschreiben Sie die equals-Methode aus der Klasse **java.lang.Object**, sodass Studenten auf Basis ihrer "id" miteinander verglichen werden.

Wenn Sie mit dieser Aufgabe fertig sind, sollte die Klasse **Main** keine Compilerfehler mehr aufweisen.

**Aufgabe 2 – Arrays und Exceptions** (20 Punkte)

Implementieren Sie die Methode **double average(double[] values)** in der Klasse **Operation**, so dass diese den Durchschnitt der im Parameter "values" übergebenen Werte berechnet und zurückliefert. Enthält "values" keine Werte oder ist "values" null, soll Ihre Implementierung der average-Methode eine **java.lang.IllegalArgumentException** auslösen.

**Aufgabe 3 – Verkettete Listen** (40 Punkte)

Die Klasse **LinkedList** soll eine einfach verkettete Liste darstellen. Implementieren Sie mit Hilfe der Definitionen in der Klasse **ListItem** die folgenden Methoden der Klasse **LinkedList**:

- **int size()**: Diese Methode soll die Anzahl der Elemente in der Liste zurückliefern.
- **void prepend(String value)**: Diese Methode soll den übergebenen String "value" an den Anfang der Liste einfügen. Der bisherige Listeninhalt soll dabei natürlich erhalten bleiben und um eins nach hinten verschoben werden.
- **String get(int i)**: Diese Methode soll den Wert des i-ten Elements in der Liste zurückgeben. Falls i kein gültiger Verweis auf ein Element der Liste ist, soll von Ihrer Implementierung der Methode eine **java.lang.IllegalArgumentException** ausgelöst werden.