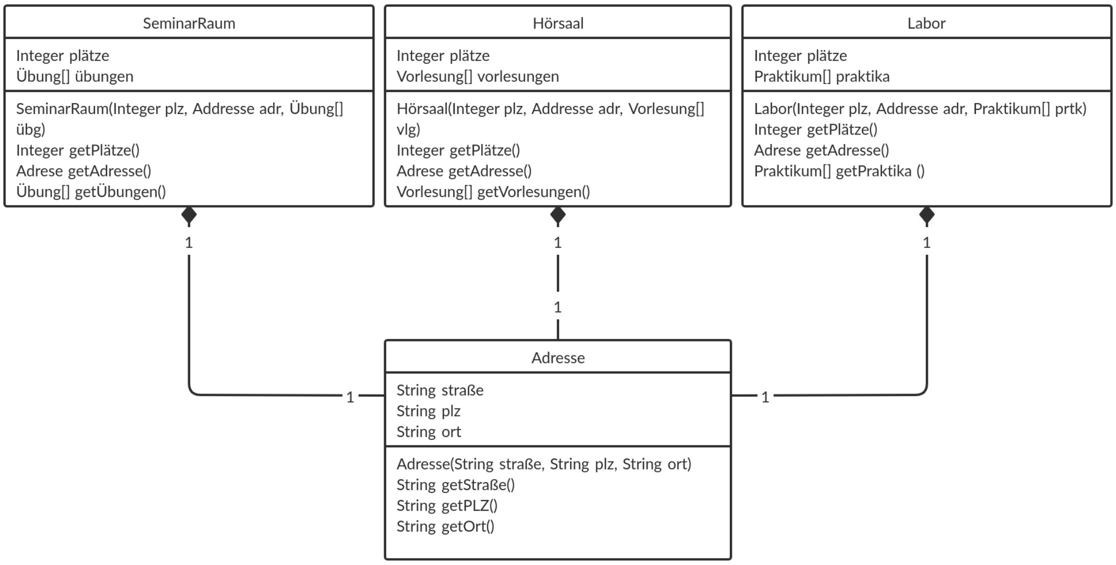
Objektorientierte Programmierung: Aufgabenblatt 3

Florian Ludewig (185722)

3. Juni 2020

Aufgabe 1.a



Aufgabe 1.b

```
// Hörsal.java
public class Hörsaal {
  public int plätze;
  public Vorlesung[] vorlesungen;
  public Adresse adresse;

  public Hörsaal(int plz, Adresse adr, Vorlesung[] vlg) {
    this.plätze = plz;
    this.vorlesungen = vlg;
    this.adresse = adr;
  }

  public int getPlätze() {
    return plätze;
  }

  public int getAdresse() {
    return adresse;
  }

  public int getVorlesungen() {
    return vorlesungen;
  }
}
```

```
// SeminarRaum.java
public class SeminarRaum {
  public int plätze;
  public Übung[] übungen;
  public Adresse adresse;

  public SeminarRaum(int plz, Adresse adr, Übung[] übg) {
    this.plätze = plz;
    this.übungen = übg;
    this.adresse = adr;
  }

  public int getPlätze() {
    return plätze;
  }

  public int getAdresse() {
    return adresse;
  }

  public int getÜbungen() {
    return übungen;
  }
}

// Labor.java
public class Labor {
  public int plätze;
  public Praktikum[] praktika;
  public Adresse adresse;

  public Labor(int plz, Adresse adr, Praktikum[] prtk) {
    this.plätze = plz;
    this.praktika = prtk;
    this.adresse = adr;
  }

  public int getPlätze() {
    return plätze;
  }

  public int getAdresse() {
    return adresse;
  }

  public int getPraktika() {
    return praktika;
  }
}
```

Aufgabe 2

```java
import java.util.ArrayList;

public class Airport {
  private int maxFlights;
  private ArrayList<Flight> flights = new ArrayList<Flight>();

  public Airport(int maxFlights) {
    this.maxFlights = maxFlights;
  }

  void addNewFlight(Flight flight) {
    if (this.flights.size() == this.maxFlights) {
      throw new Error("Max number of flights reached");
    }

    for (Flight f : this.flights) {
      if (f.flightNumber == flight.flightNumber) {
        throw new Error("The flight with number " + flight.flightNumber + " cannot be added twice!");
      }
    }

    this.flights.add(flight);
  }

  void removeFlight(int flightNumber) {
    ArrayList<Integer> flightNumbers = new ArrayList<Integer>();
    for (Flight f : this.flights) {
      flightNumbers.add(f.flightNumber);
    }

    int index = flightNumbers.indexOf(flightNumber);
    if (index >= 0) {
      this.flights.remove(index);
    }
  }

  void listDeparturesOnScreen() {
    for (Flight f : this.flights) {
      if (!f.inOut) {
        System.err.println("Flight number: " + f.flightNumber + " (departure time: " + f.time + ", gate: " + f.gate
            + ", destination: " + f.location + ")");
      }
    }
  }

  void listArrivalsOnScreen() {
    for (Flight f : this.flights) {
      if (f.inOut) {
        System.err.println("Flight number: " + f.flightNumber + " (arrival time: " + f.time + ", gate: " + f.gate
            + ", location: " + f.location + ")");
      }
    }
  }
}
```