VENTSPILS AUGSTSKOLA in collaboration with accenture

Seminar 2

Classes, Objects

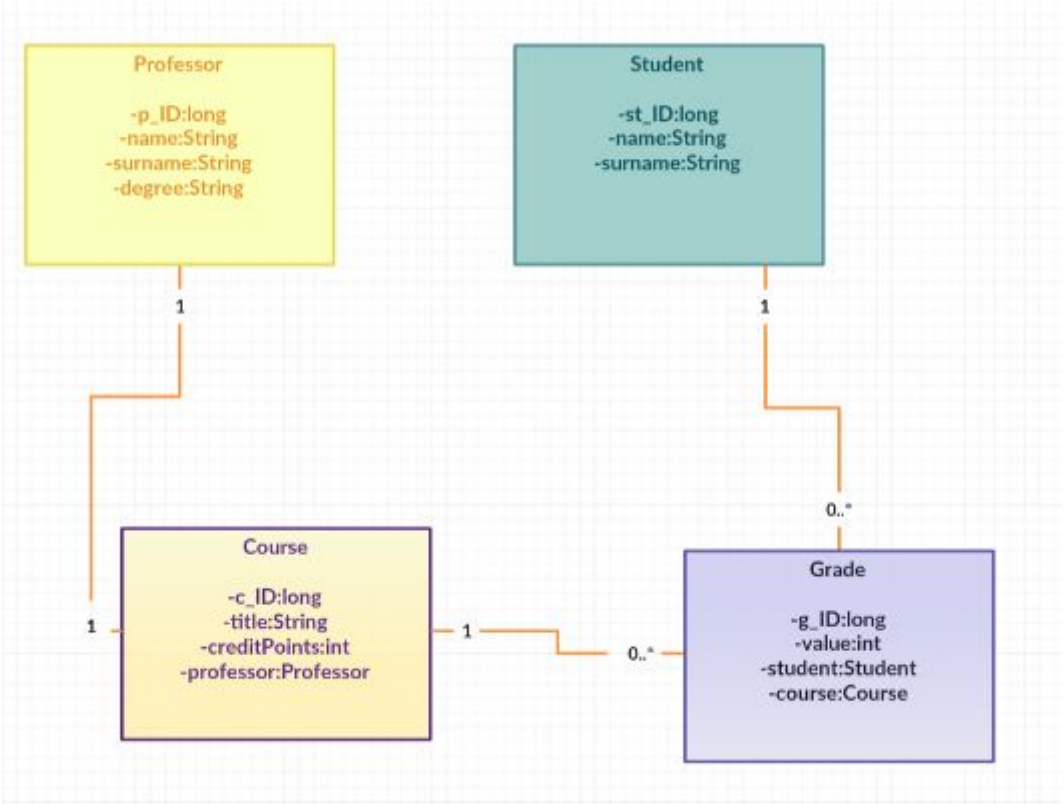
- **Open Eclipse(or NetBeans, or ...), create new empty Java Application Project "Seminar_2"**

Exercise 1- create model: class definition, constructor, set and get methods, toString methods

- **Create new package named model**

Exercise 1.1 - classes and constructors

- **Create new classes (no main() method in it): Professor, Student, Course and Grade based on Entity Relationship Diagram (see below)**



- **Create two constructors for each class:**
  - no-argument constructor, which initializes the default values
  - argument constructor, which initializes variables based on input parameters

Exercise 1.2 - getters and setters

- **Create get() and set() methods for all non-static variables of classes Professor, Student, Course and Grade**

Hint:

Easier way: Right click on the source code->Source->Generate Getters and Setters…

Exercise 1.3 - model verification

- **Add verification for each set method. For example, please verify if the grade is between 1 and 10 or if the name consists from only letters.**

Exercise 1.4 -Static variables: calculate id using autoincrement

- **Add a static variable long idCounter for each class and initialize it to**
  - 0 in Professor class;
  - 10000 in Student class;
  - 100000 in Course class;
  - 200000 in Grade class.
- **Add an increment of idCounter in each class constructors;**

```
idCounter++;
```

Exercise 1.5 - adding ID

- **Add a variable long p_ID to the class Professor (if it has not been done before) and initialize it to be equal with the variable idCounter in the constructors (during the creation of the object);**
- **Add a get() method for this variable (getProfID()) but no set() method**
- **Repeat the same actions in classes Student, Course and Grade**

Exercise 1.6 - toString() methods

- **Create toString() methods for each class**

Hint:

Easier way: Right click on the source code->Source->Generate toString()…

Exercise 2-model verification in main method

- **Create new package named main**
- **Create a new class VeA_IS containing:**
  - **Main method:**

```
public static void main(String[] args)
```

In main() you need to create new objects of classes Professor, Student, Course and Grade. Call methods toString().

Exercise 3 - storing

- **Create new arrays or ArrayLists for objects of each class:**

```
public static ArrayList<Professor> profLists = new
ArrayList<>();
```

- **Save objects in the appropriate array or ArrayList**

Exercise 4 - calculation

- **Create a new method which calculates average grade for each student;**
- **Create a new method which calculates average grade based on course credit points (weighted average grade) for each student;**
- **Create a new method which calculates average grade for each course;**
- **Create a new method which calculates how many courses does professor teach;**
- **Create a new method which sort all students by their average grade;**

Exercise 5 - Inheritance*

Let us practice inheritance.

- **Create a new class Person with variables *name* and *surname* (the same as in classes Student and Professor);**
- **Change definition of classes Student and Professor -> add inheritance**

```
public class Student extends Person
```

- **Change constructors and toString() methods based on inheritance**
- **Remove unnecessary set() and get() methods in each class.**

Exercise 6 - Polymorphism*

- **Create a mixed array or ArrayList of both types of Person**
- **Call toString() method to each object from for-loop**