Object-Oriented Programming

Welcome back!

Object-Oriented Programming

Sharing…

1. What did you study?
2. What exercises did you do?
3. What [new] thing/s did you learn?
4. Were you able to improve your OOP understanding?
5. What are your realizations?

Object-Oriented
Programming

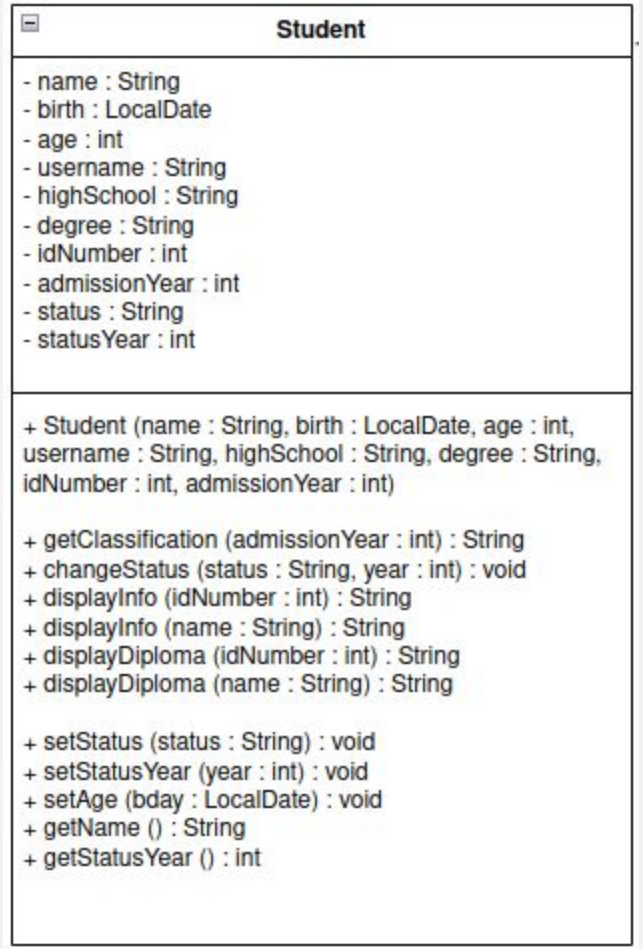
Practice GUI & MVC

Next meeting…

- Graded Exercise 6 (UML)
- Graded Exercise 6.5 (M-**View**-C Coding)

Practice Exercise

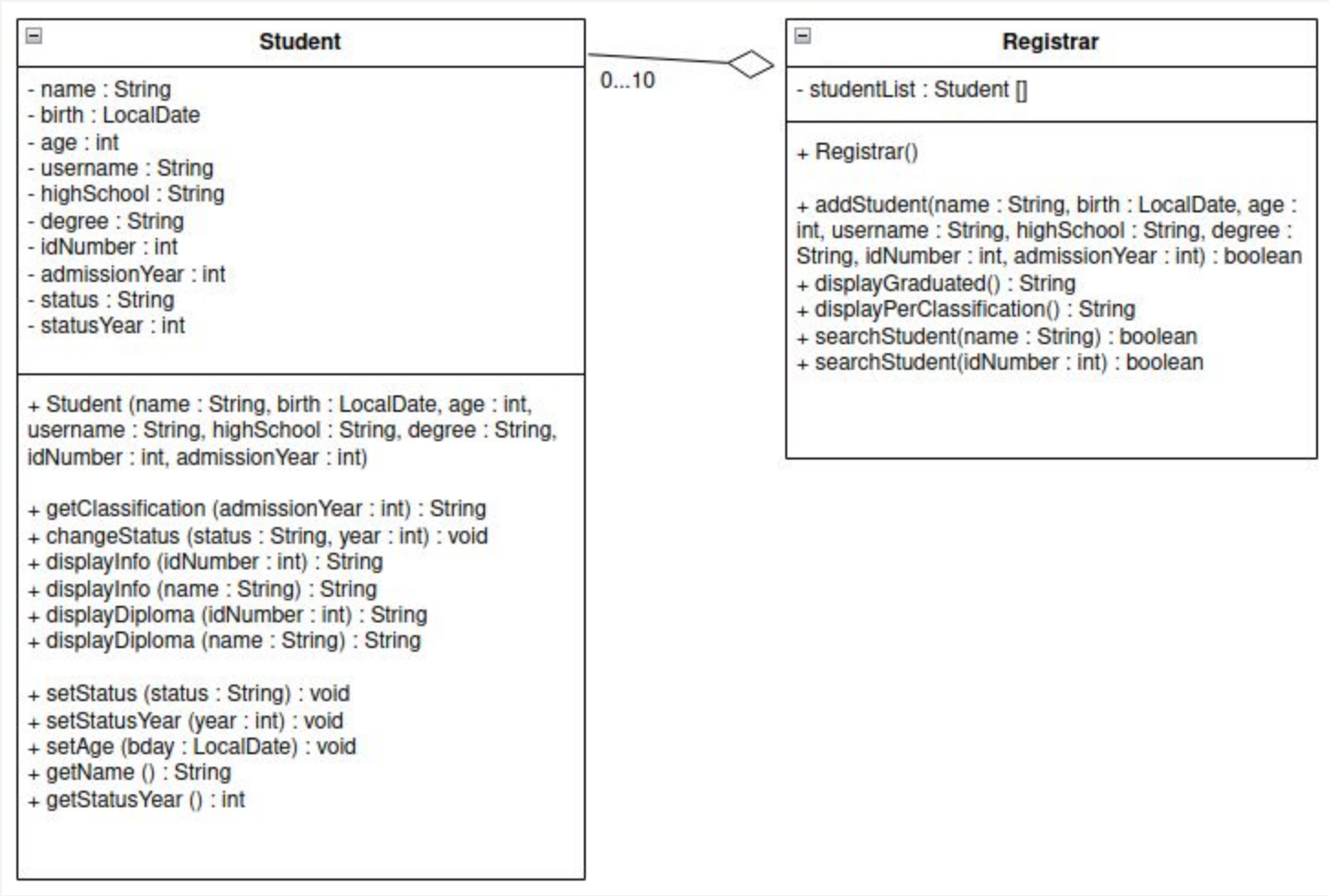
- Upon admission, D-University students are required to fill out a bio information sheet. In this sheet, their name, date of birth, age, preferred username, high school, Degree Code, student ID number and year of admission are captured. By default, the status of a student is “admitted”. Your task is to design a system that can accommodate the following features:
  - Determine the classification of a student [Freshman, Sophomore, Junior, Senior or Terminal (year of admission >= 5)]
  - Change Student Status
    - Expelled student
      - flips the status of student to “expelled”
      - Ask for year of expulsion
    - Graduate student
      - flips the status of student to “graduated”
      - Ask for year of graduation
  - Generate Info Sheet of a Student (Show full student info)
  - Generate Diploma Fields
    - Display the following: Name, Year Admitted, Degree Code, Year Graduated
    - Your system should print an error message if the student has not graduated
- Convert your Student class diagram to Code.
- Create a Main Class (that acts as your View Class) triggering the defined features.


Student
- name : String
- birth : LocalDate
- age : int
- username : String
- highSchool : String
- degree : String
- idNumber : int
- admissionYear : int
- status : String
- statusYear : int
+ Student (name : String, birth : LocalDate, age : int, username : String, highSchool : String, degree : String, idNumber : int, admissionYear : int)
+ getClassification (admissionYear : int) : String
+ changeStatus (status : String, year : int) : void
+ displayInfo (idNumber : int) : String
+ displayInfo (name : String) : String
+ displayDiploma (idNumber : int) : String
+ displayDiploma (name : String) : String
+ setStatus (status : String) : void
+ setStatusYear (year : int) : void
+ setAge (bday : LocalDate) : void
+ getName () : String
+ getStatusYear () : int

Interaction of Classes: Expanding from Practice Exercise

- Create a **Registrar class** that has an array of student objects as its attribute. You may assume that the system can only accommodate 10 student profiles.
  - Observe that Registrar Class is still a model class.
- Implement the following features:
  - Create a new student profile
  - Display the number of students graduated
  - Display the number of students per classification
  - Search* for a student based on the substring name *(case insensitive)* and display student info *(e.g., name: "Joseph" vs. search keyword: "Jose")*
  - Search* for a student based on their ID number *(full match)* and display student information

** For search functions, system should display "no match found" if there are no matches in the search. If there are multiple matches, display only first result.*


Student
- name : String
- birth : LocalDate
- age : int
- username : String
- highSchool : String
- degree : String
- idNumber : int
- admissionYear : int
- status : String
- statusYear : int
+ Student (name : String, birth : LocalDate, age : int, username : String, highSchool : String, degree : String, idNumber : int, admissionYear : int)
+ getClassification (admissionYear : int) : String
+ changeStatus (status : String, year : int) : void
+ displayInfo (idNumber : int) : String
+ displayInfo (name : String) : String
+ displayDiploma (idNumber : int) : String
+ displayDiploma (name : String) : String
+ setStatus (status : String) : void
+ setStatusYear (year : int) : void
+ setAge (bday : LocalDate) : void
+ getName () : String
+ getStatusYear () : int
0...10
Registrar
- studentList : Student []
+ Registrar()
+ addStudent(name : String, birth : LocalDate, age : int, username : String, highSchool : String, degree : String, idNumber : int, admissionYear : int) : boolean
+ displayGraduated() : String
+ displayPerClassification() : String
+ searchStudent(name : String) : boolean
+ searchStudent(idNumber : int) : boolean

Keep learning…