MODULE

Week 10 Workshop Walk-throughs, Tutorial Discussions and Lab Tasks

Stephen Huxford, Marc Cheong
Updated 30 September 2021

Week 10 Tutorial Discussions

Week 10 Lab Tasks

In this week, you will continue your work on the week 9 University Application.

L1 (0.2 marks)

Add a new class called **Faculty** with the following attributes:

INSTANCE VARIABLE	DESCRIPTION
facultyName	at least 3 characters

- add a one-parameter constructor
- add a getter and a setter. The setter must return true if the input value is valid; false otherwise.
- add a toString method

L2 (0.2 marks)

Add a new class called **Student** with the following attributes:

INSTANCE VARIABLE	DESCRIPTION
studentName	at least 2 characters
age	a numeric value between 15 and 95 (inclusive)

- add a two-parameter constructor
- add getters and setters for both instance variables. The setters must return true if the input value is valid; false otherwise.
- add a toString method

L3 (0.2 marks)

Add a new class called **Enrollment** with the following attributes:

INSTANCE VARIABLE	DESCRIPTION
student	An instance of class Student

INSTANCE VARIABLE	DESCRIPTION
faculty	An instance of class Faculty

- add a two-parameter constructor
- add getters for both instance variables.
- add a toString method

L4 (0.2 marks)

Update (refactor) class **University** with the followings:

INSTANCE VARIABLE	DESCRIPTION
universityName	at least 3 characters
worldRank	a positive non-zero integer
faculties	an ArrayList of Class Faculty
students	an ArrayList f Class Student
enrollments	an ArrayList of Class Enrollment

- add a two-parameter constructor for the attributes unversityName and worldRank.
  - use the constructor to initialize the three array lists.
- add getters for both instance variables faculties and students.

L5 (0.6 marks)

- Add/Update **AddStudent** method such that:
  - accepts two input values represent the student's **name** and **age**
  - if the **name** and/or **age** are not valid, return false
  - if the input name already exists in the array list **students**, return false.
    - Hint:'Use the appropriate loop structure'
  - If the student **name** and **age** are valid and the name does not exist in the ArrayList **students**, then:
    - Create a new instance of class **Student**
    - pass the input parameters **name** and **age** to the constructor of class **Student**
    - add the new instance to the ArrayList **students**
- Add/Update **AddFaculty** method such that:
  - accepts one input value represents the name of the faculties
  - if the **name** is not valid, return false
  - if the input name already exists in the array list **faculties**, return false.
    - Hint:'Use the appropriate loop structure'
  - If the faculty **name** is valid and the name does not exist in the ArrayList **faculties**, then:
    - Create a new instance of class **Faculty**
    - pass the input parameter **name** to the constructor of class **Faculty**
    - add the new instance to the ArrayList **faculties**

L6 (0.6 marks)

- Add/Update **AddEnrollment** method such that:
  - It accepts two input parameters: the **studentName** and **facultyName**.
  - if the array list **students** does not contain a student with the name **studentName**, print an error message and return false.
    - if the **studentName** can be found, set a reference to its object
      - `Student selectedStudent = students.get(i);`
  - if the array list **faculties** does not contain a faculty with the name **facultyName**, print an error message and return false.
    - if the **facultyName** can be found, set a reference to its object
      - `Faculty selectedFaculty = faculties.get(i);<br>`
  - Create a new instance of the **Enrollment** class, add it to the **enrollments** arraylist, and return true

```
1. Enrollment newEnrol=new Enrollment(selectedStudent, selectedFaculty);
2. enrollments.add(newEnrol);
```

- Apply all the required updates to the method countStudentsPerFaculty such that it uses the new structure to return the number of students per the given faculty.

Driver Class

for the given driver class:

```
public class DriverClass {
    public static void main(String[] args) {
        University monash=new University("Monash",20);
        monash.addFaculty("FIT");
        monash.addFaculty("FIT");
        monash.addFaculty("Law");
        monash.addFaculty("Business");
        monash.addFaculty("La");//rejected
        System.out.println("Faculties="+monash.getFaculties()); //test faculties OK
        monash.addStudent("John",23);
        monash.addStudent("John",98);
        monash.addStudent("Tim",54);
        monash.addStudent("Emma",19);
        monash.addStudent("E",67); //rejected
        System.out.println("Students="+monash.getStudents());
        monash.addEnrollment("Michael","FIT"); //rejected, no such student
        monash.addEnrollment("Tim","Art"); //rejected, no such faculty
        monash.addEnrollment("Emma","FIT");
        monash.addEnrollment("Tim","Law");

        int studentsInFIT=monash.countStudentsPerFaculty("FIT");
        int studentsInLaw=monash.countStudentsPerFaculty("Law");
        int studentsInBusiness=monash.countStudentsPerFaculty("Business");
        System.out.println("Number of students in FIT ="+studentsInFIT);
        System.out.println("Number of students in Law ="+studentsInLaw);
        System.out.println("Number of students in Business ="+studentsInBusiness);
    }
}
```

the expected output can be:

```
Faculties=[Faculty{facultyName='FIT'}, Faculty{facultyName='Law'}, Faculty{facultyName='Business'}]
Students=[Student{studentName='John', age=23}, Student{studentName='Tim', age=54}, Student{studentName='Emma', age=19}]
Error:Student or Faculty cannot be found
Error:Student or Faculty cannot be found
Number of students in FIT =1
Number of students in Law =1
Number of students in Business =0
```