TOBB ETÜ, BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

BİL 113/012 Computer Programming I

HOMEWORK 4 (50 Points)

To be demonstrated by November 23th

1 [30 Points] Undergraduate and Graduate Applications

In this assignment, you are going to simulate both the undergraduate and graduate school application process using the following 4 classes: **Student**, **OSYM**, **Result**, and **School**. The **Student** class represents both high school and university students, who have the following personal information: name, surname, skills (depends on whether the student is a high school or university student), school name, GPA, etc. **School** class represents schools that have a name, location, minimum scores for programs and decides applicants should be accepted or rejected to the program which they applied to. University students must enter ALES and YDS before applying to graduate programs, while high school students must enter YKS before applying to undergraduate programs. **OSYM** class conducts all of these exams, evaluates students, and returns a Result object for the exam taken. **Result** class stores the raw scores of each section for YDS, YKS, and ALES.
The (incomplete) designs of these classes are described below.

1. **Student Class**:

   Student class should have the additional instance variables such as, *scienceSkill, socialScienceSkill, yksResult, and schoolType.* There are more instance variables needed to complete this task, but you should infer them from the Demo class we share, the example outputs, and the signatures of the other methods.

   Student class should have the following methods (and other methods that you need to infer):

   - The constructor: *Student(String name, String surname, int age, int mathSkill, int scienceSkill, int turkishSkill, int socialScienceSkill, String schoolName, double gpa)*

- Another constructor for university students.
- The method with signature *public boolean enterExam(int examType)*. This method must call the *makeExam* method of the **OSYM** class and save the result returned by that method to correct instance variable.
- The method with signature *public void applyToSchool(School school, int program)*. This method must call the *evaluateApplication* method of **School** class and save the result to applicationStatus instance variable.

2. **OSYM Class**: OSYM class should have (at least) the 4 methods whose signatures are given below:

    - *Result makeExam(Student student, int type)*: This method must call one of the methods described below according to the type parameter.
    - *Result makeALES(Student student)*: This method should create and return a Result object which stores scores to mathScore, turkishScore.
    - *Result makeYDS(Student student)*: This method should create and return a Result object which stores scores to the englishScore.
    - *Result makeYKS(Student student)*: This method should create and return a Result object which stores scores to mathScore, scienceScore, turkishScore, socialScienceScore
    - Score for each section is selected uniformly at random from the range $[0.95 * skill, 1.05 * skill]$ using the appropriate skill.

    Scores cannot be higher than 100 and fractional scores should be rounded to the next integer.
    High school students can only enter YKS.
    University students can enter both YDS and ALES.

3. **Result Class:**
    Result class should have (at least) the following instance variables: *type, mathScore, scienceScore, turkishScore, socialScienceScore, englishScore.*

    This class stores raw scores for each section of an exam, e.g., for ALES, it should store mathScore and turkishScore, and its type should indicate that it is an ALES result.

    Result class should have a separate constructor for each type of exam (**and should not have type as a parameter**).

4. **School Class:**
    School class should have the following instance variables (and some other variables):

    a) *minimumUndergraduateComputerEngineeringScore*

    b) *minimumUndergraduateElectronicEngineeringScore*

    c) *minimumUndergraduateMechanicalEngineeringScore*

    d) *minimumUndergraduateIndustrialEngineeringScore*

and the *boolean evaluateApplication(Student student, int program)* method. This method decides if the student should be accepted to the program or not.
For undergraduate programs, a *final score* is calculated and then compared with the minimum score of the target program for the acceptance or rejection decision. The final score is calculated from the YKS result and GPA by the formula below:

$$examScore = 0.35*mathScore+0.25*scienceScore+0.25*turkishScore+0.15*socialScienceScore$$

$$finalScore\ = 0.8*examScore+0.2*student's\ GPA$$

If the final score is greater than or equal to the minimiumScore of the undergraduate program, *evaluateApplication* method should return true, and return false otherwise.

For graduate programs method should check the following conditions:

- The quantitative score of the ALES results must be greater than or equal to 85.
- The English score must be greater than or equal to 75.
- The GPA must be greater than or equal to 2.5.

The quantitative score is calculated from an ALES result as follows:

$$QuantitativeScore = 0.75*mathScore+0.25*turkishScore$$

If all conditions are satisfied the *evaluateApplication* method should return true, and return false otherwise.
Also, high school students cannot apply to graduate programs, and university students cannot apply to undergraduate programs.

5. **Student** and **School** classes should not have setter methods.

6. **There are some methods and instance variables that are not explicitly described above that you need to infer from the signatures of the other methods and Demo classes.**

7. Your codes **must** be compatible with the Demo classes we shared. You **cannot change** the Demo classes.

```
Example output of Demo1
name=Utku Umur, surname=ACIKALIN, age=23, schoolName=TOBB ETU
name=TOBB ETU, location=Ankara
name=Bilkent, location=Ankara
Math Score:95    Turkish Score:90
YDS Score:91
You are not allowed to enter this exam!..
You cannot apply to undergraduate programs.
Rejected!..
Accepted!..
```

```
Example output of Demo2
name=Utku Umur, surname=ACIKALIN, age=18, schoolName=YFL
name=TOBB ETU, location=Ankara
name=Bilkent, location=Ankara
You are not allowed to enter this exam!..
You are not allowed to enter this exam!..
Math score:86
Science Score:93
Turkish Score:86
SocialScience Score:81
Accepted!..
Your YKS score must be higher!..
Rejected!..
Accepted!..
You cannot apply to graduate programs.
Rejected!..
```

2 [10 POINTS] CALCULATING PI USING MONTE CARLO SIMULATIONS

In this task, we are going to try to calculate the value of the PI empirically using Monte Carlo Simulations. Consider the ratio of the area of the circle to the area of the square in the figure below.

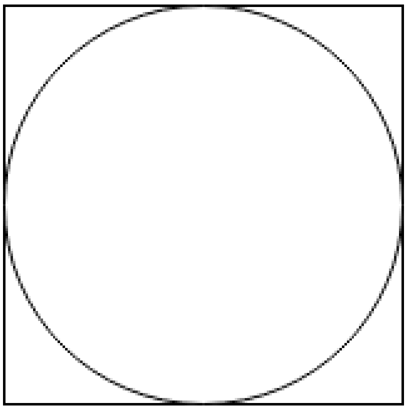

Figure 2.1: Cirle and Square

If the radius of the circle equals to 1, then sides of the square have length 2. The area of the circle is π and the area of the square is 4. $\frac{Area(circle)}{Area(square)} = \frac{\pi}{4}$. We can calculate the PI as follows:

- Select n points inside the square uniformly at random.
- Count how many of these points are inside the circle. Let m denote the number of points inside the circle.
- If n is large, ratio of m to n will be close to the ratio of the are of the circle to area of the square which is equal to $\frac{\pi}{4}$.
- So, $\frac{4m}{n} \approx \pi$

Write a program which calculates PI using this method and print the values of PI for $n \in \{100, 10000, 1000000, 100000000, 1000000000\}$.

```
Example Output:
100:3,080000
10000:3,134000
1000000:3,143340
100000000:3,141625
1000000000:3,141566
```

3 [10 POINTS] FINDING THE KTH SMALLEST NUMBER OF A GIVEN ARRAY

In this task, you are going to write a method that takes an integer array A and an integer k as parameters, and returns k^{th} smallest integer in A.

For instance, if $A = [3, 1, 7, 5, 3]$ and $k = 2$, your method should return 3.