UTI 526 Object Oriented Programming

5^{th} Homework
Assignment Date: 15.04.2024
Due Date: 22.04.2024

From: Java How to Program, Early Objects (10th Edition), Paul J. Deitel and Harvey Deitel, Pearson, 2015.

9.5) Draw an inheritance hierarchy for students at a university similar to the hierarchy shown in Fig. 9.2. Use `Student` as the base class of the hierarchy, then extend `Student` with classes `UndergraduateStudent` and `GraduateStudent`. Continue to extend the hierarchy as deeply (i.e., as many levels) as possible. For example, `Freshman`, `Sophomore`, `Junior` and `Senior` might extend `UndergraduateStudent`, and `DoctoralStudent` and `MastersStudent` might be subclasses of `GraduateStudent`. After drawing the hierarchy, discuss the relationships that exist between the classes. [Note: You do not need to write any code for this exercise.] **(20 points)**

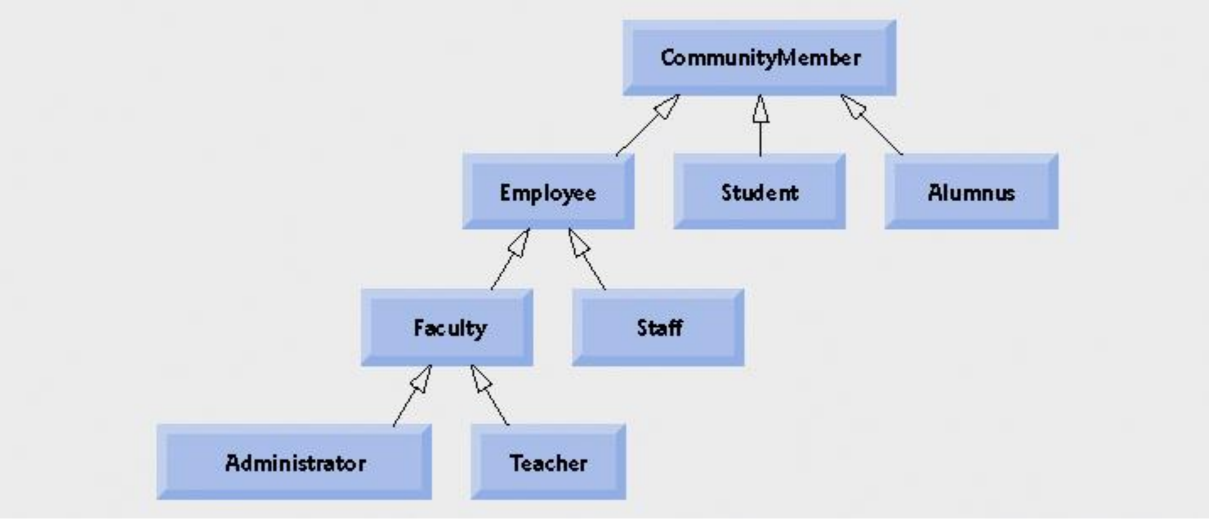

Fig. 9.2. Inheritance hierarchy UML class diagram for university `CommunityMembers`

10.9) ***(Shape Hierarchy)*** Implement the Shape hierarchy shown in Fig. 9.3. Each `TwoDimensionalShape` should contain method `getArea` to calculate the area of the two-dimensional shape. Each `ThreeDimensionalShape` should have methods `getArea` and `getVolume` to calculate the surface area and volume, respectively, of the three-dimensional shape. Create a program that uses an array of `Shape` references to objects of each concrete class in the hierarchy. The program should print a text description of the object to which each array element refers. Also, in the loop that processes all the shapes in the array, determine whether each shape is a `TwoDimensionalShape` or a `ThreeDimensionalShape`. If it's a `TwoDimensionalShape`, display its area. If it's a `ThreeDimensionalShape`, display its area and volume. **(80 points)**

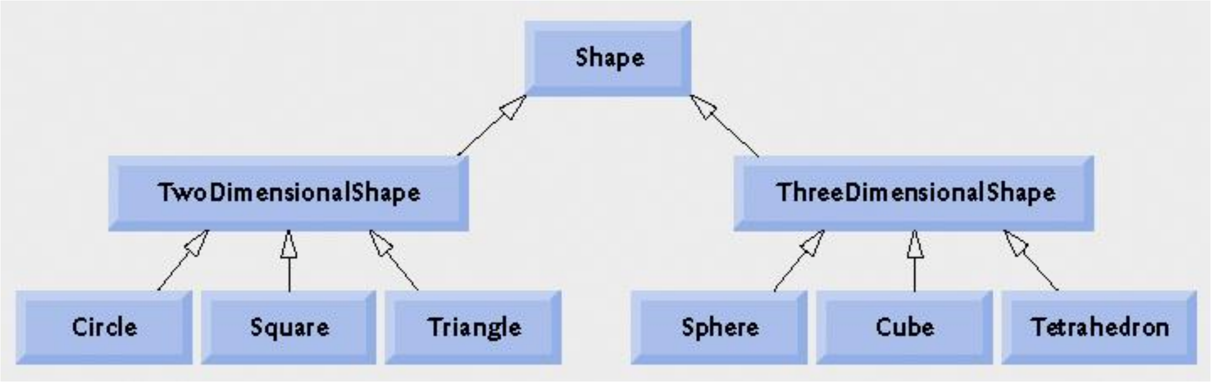

Fig. 9.3. Inheritance hierarchy UML class diagram for Shapes

Important Notes:
1. All source code and related homework reports should be submitted via Ege Ders platform: 2023 - 2024 Bahar Dönemi → Enstitüler → Fen Bilimleri Enstitüsü → Uluslararası Bilgisayar → Bilgi Teknolojileri ve İnternet Güvenliği → İÖ - Nesne Yönelimli Programlama - 540443 - 2324B → Hafta 9: Polymorphism and Interfaces → Homework 5.
2. Do not forget to include appropriate comments in the source code. Hence the grader can easily understand the program during his/her assessment.
3. Write the programs in a simple and straightforward manner by considering object-oriented analysis and design principles.
4. Each report should include the printout of the related source code, two or more screenshots (depending on the illustration requirements) which exemplify execution of the programs and proper UML diagrams.
5. Homework reports are MANDATORY! Sending only source code without reports including the above mentioned content is subject to getting lower points.
6. IMPORTANT NOTICE: There will be significant point deductions for late, copied or shared submissions.