Izzy Hyman

Data Structures

Professor Bari

2/17/24

Homework 1: Course Registration Overview

Method overloading: Method overloading is when you define multiple methods that have the same name but different signatures. This could mean two methods having the same name but a different number of arguments or different argument types. This happens in my code a few times, but one example is with the register method in both the Student and Admin classes. In both cases, the method name is “registerStudent”, however in the student class, the parameters are “String courseName, int sectionNum, String fName and String lName” and in the admin class the parameter is “Student student”.

A computer screen shot of text

Description automatically generated

A computer screen with text

Description automatically generated

Method overriding: Method overriding is when the child inherits a method from the parent class but it changes the function of that method. The name and parameters remain unchanged from the parent but the actual execution of what the method is meant to do changes. This occurs several times in this assignment. Two examples are when the admin and student classes inherit from the user class a method called “viewCourses” but then the actual implementation of it in the admin class and student class not only differ from the definition in the user class, but they also differ from each other.

User class:

A black background with white text

Description automatically generated

Admin class:

A computer screen shot of text

Description automatically generated

Student class:

A screen shot of a computer code

Description automatically generated

Abstract class: This is a class that cannot be instantiated. Instead, they are implemented by concrete subclasses. An abstract class contains abstract methods, which are methods that are not implemented. Abstract classes can act as blueprints to the actual class that instantiates it. In this assignment, an example of an abstract class are the interfaces I made for the admin and student classes.

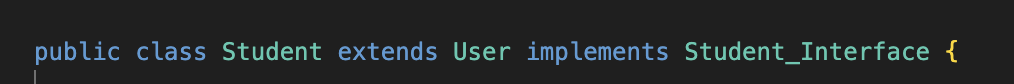
A screen shot of a computer code

Description automatically generated

Inheritance: This is when a child class inherits characteristics (methods and data) that are defined in the a class. Inheritance is denoted by the word “extends”. When a child class inherits from a parent class, it *extends* that parent class. The example in the course registration system program is that the Student and Admin classes both inherit from the user class. This made it so that I didn’t have to specifically define that both a student and admin had a username, password, first name, and last name. Instead, because both of these classes inherited from the user class, they both contained these attributes automatically.

A black background with blue and green text

Description automatically generated



Polymorphism: This is when objects can take on many forms depending on their data types. An example of this in my code is method overloading. See above for the examples of method overloading in my code.

Encapsulation: Encapsulation is one of the four pillars of OOP and it refers to the hiding of information. More specifically, this is the idea of declaring data fields as private, sometimes also methods, and then retrieving them through getters and setters. In this assignment, this was present in the course, student, user, and admin classes and below I have illustrated examples from the course class.

A screen shot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

The concept of ADT: An ADT acts as a high-level description/blueprint that outlines what a specific type of data can do and it describes its characteristics without actually detailing the implementation of those data. In this assignment, an example of ADTs are the interfaces for both student and admin. They are both descriptions of what the admins and students can do, without actually explaining how they are to do them.

A screen shot of a computer code

Description automatically generated