CSU Long Beach
CECS 277
Object Oriented Application Development

Syllabus
Schedule
Grading
Project ArrayList
Project Inheritance
Project File I/O
Project Generics
Project Regular Expressions
Project Collections
Project Gui
Project Threads
Project Binary I/O
Code
Lecture Notes
Using Eclipse
Textbook Resources
Java API Documentation
Java Tutorial

Mimi Opkins Home
CECS 323 Home
CECS 493 Home

Project IPI

Inheritance/Polymorphism/Interface Assignment

Part I

- Consider geometric objects. Suppose you want to design the classes to model geometric objects such as circles and rectangles.
- Geometric objects have many common properties and behaviors. They can be drawn in a certain color and be filled or unfilled. Thus a general class GeometricObject can be used to model all geometric objects.
- This class contains the properties color and filled and their appropriate get and set methods. Assume that this class also contains the dateCreated property and the getDateCreated() and toString() methods. Thus it makes sense to define the Circle class that extends the GeometricObject class.
- Likewise, Rectangle can also be defined as a subclass of GeometricObject.
- The Circle class inherits all accessible data fields and methods from the GeometricObject class.
- In addition, it has a new data field, radius, and its associated get and set methods. The Circle class also contains the getArea(), getPerimeter(), and getDiameter() methods for returning the area, perimeter, and diameter of the circle.
- The Rectangle class inherits all accessible data fields and methods from the GeometricObject class.
- In addition, it has the data fields width and height and their associated get and set methods. It also contains the getArea() and getPerimeter() methods for returning the area and perimeter of the rectangle.
- Design three classes
    1. A Circle class as described above with a default radius of 1.0
    2. A Rectangle class as described above with a default width and height of 1.0
    3. A class named Triangle that also extends GeometricObject. The class contains:
        - Three data fields named side1, side2, and side3 with default values 1.0 to denote three sides of the triangle.
        - A no-arg constructor that creates a default triangle.
        - A constructor that creates a triangle with the specified side1, side2, and side3.
        - The accessor methods for all three data fields.
        - A method named getArea() that returns the area of this triangle.
        - A method named getPerimeter() that returns the perimeter of this triangle.
        - A method named toString() that returns a string description for the triangle.
        - The area of a triangle can be calculated with this formula:
          A = sqrt(s(s-a)(s-b)(s-c))
          where a, b, and c are the lengths of the sides and s is half of the perimeter.
- Here is the code for Geometric Object: GeometricObject.java
- Write a test program that creates Circle, Rectangle and Triangle objects. Your test program will be very extensive.
    - Make sure you test all the constructors.
    - Make sure you test all the methods.
    - Create tests that will demonstrate polymorphism. You can do this by creating an ArrayList of Geometric objects then proving that if you invoke the getArea() or getPerimeter() methods, the correct version of the method will be executed.

Part II

- Modify the GeometricObject class to implement the *Comparable* interface, and define a static max method in the GeometricObject class for finding the larger of two GeometricObject objects.
- Write a test program that uses the max() method to find the larger of two circles and the larger of two rectangles. Use the appropriate methods to determine if two GeometricObjects are the same.
- Add all your objects to an ArrayList, sort it, then print out the sorted ArrayList.

Grading Criteria

You will be graded on the following components:

- Does the program do what is required
- Is it properly documented
- Is it fully tested
- Is it properly designed

Latest Update: Friday, 12-Aug-2016 17:50:16 PDT

mimi.opkins@csulb.edu

[Top of Page]