Lab – Java - Polymorphism and Abstract

Overview

Write a program that uses polymorphism and interfaces.

Part 1 – Main Class

Create a class called Main that has a main method.

Part 2 – Shape Class

Create an abstract class named Shape. It should contain a member variable for area that is a double. The member variable should be accessible to all derived classes.

Add an abstract method named calculateArea. It should have a void return type and take 0 parameters.

Part 3 – Circle Class

Create a class named Circle that is derived from Shape. It should contain a member variable for radius that is a double. Add a constructor that takes the radius as a parameter and sets the member variable radius to that value.

Override the calculateArea method. It should set the area based on the radius. Use the following formula: 3.14 * radius *radius

Part 4 – Rectangle Class

Create a class named Rectangle that is derived from Shape. It should contain member variables for length and width that are double. Add a constructor that takes the length and width as parameters and sets the corresponding member variables.

Override the calculateArea method. It should set the area based on the length and width. Use the following formula: length * width

Part 5 – Main Method

In main, create an array of Shape that contains 4 elements. Put instances of Circle and Rectangle in the array (two instance of each class should be in the array).

Write a loop that calls calculateArea on each array element.

© 2023 Arthur Hoskey. All rights reserved.

Part 8 – Main calcAndShowArea Method

Add a static method named calcAndShowArea to the Main class. It should take an array of Shape as a parameter. There should be a loop that iterates through all the Shape elements. The loop should calculate the area then display it on screen.

Part 9 – Update Main Method

Add a call to calcAndShowArea. Make sure to pass an array of Shape to the method.

Part 10 – Add Circumference Calculations

Add a member variable for circumference to the Shape class. The member variable should be accessible to all derived classes.

Add an abstract method to the Shape class named calculateCircumference (void return type and no parameters). Update derived classes as necessary to handle the new functionality.

Circle Circumference – 2 * 3.14 * radius

Rectangle Circumference – 2 * length + 2 * width

Part 11 – Main calcAndShowCircumference Method

Add a static method to the Main class to calculate and display the circumference. It should take an array of Shape as a parameter. There should be a loop that iterates through all the Shape elements. The loop should calculate the circumference then display it on screen.

Part 12 – Update Main Method

Add a call to calcAndShowCircumference. Make sure to pass an array of Shape to the method.

© 2023 Arthur Hoskey. All rights reserved.