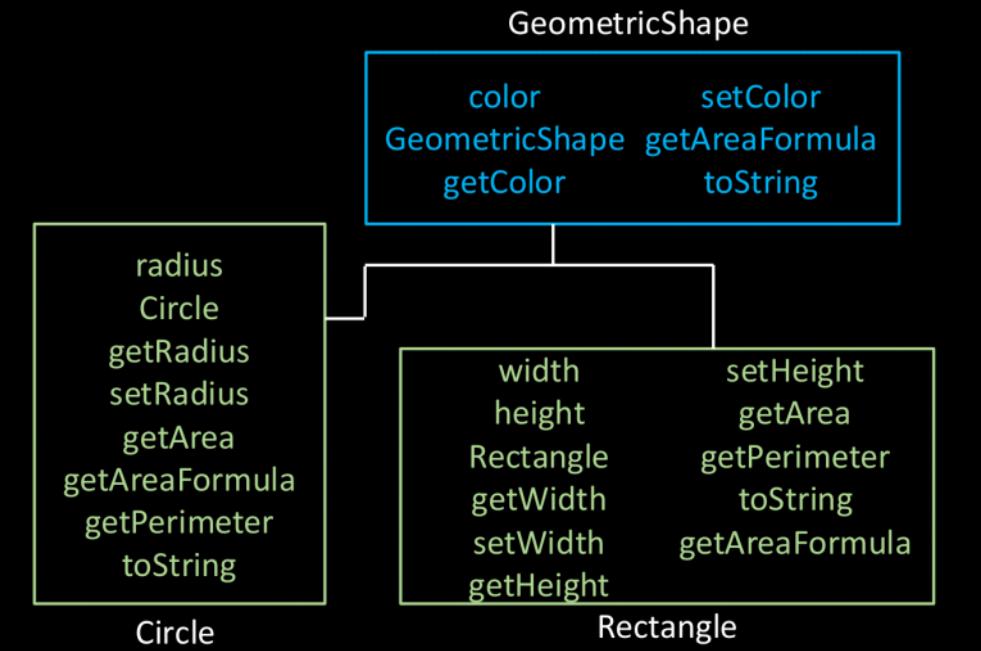


GeometricShape as a Superclass

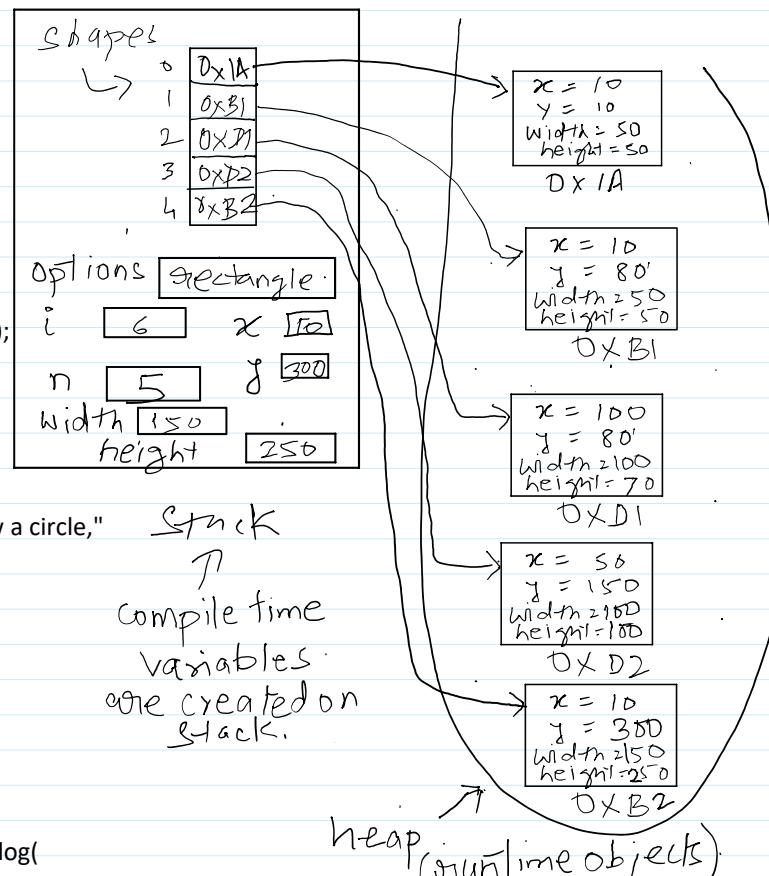


```

package Polymorphism;
import javax.swing.*;
import MyGraphics.*;
/**
 * Write a description of class DrawingCircleAndRectangle here.
 */
/* @author (your name)
 * @version (a version number or a date)
 */
public class DrawingCircleAndRectangle
{
    public static void main(String args[]){
        int n = Integer.parseInt(
            JOptionPane.showInputDialog("Enter number of Shapes:"));
        Shape shapes[] = new Shape[n];
        String options;
        double x, y, width, height;
        for(int i = 0 ; i < n; i++){
            options = JOptionPane.showInputDialog("Input Circle to draw a circle," +
                " and input Rectangle to draw rectangle: ");
            switch(options){
                case "Rectangle":
                case "rectangle":
                    x = Double.parseDouble(JOptionPane.showInputDialog(
                        "Enter your x coordinate:"));
                    y = Double.parseDouble(JOptionPane.showInputDialog(
                        "Enter your y coordinate:"));
                    width = Double.parseDouble(JOptionPane.showInputDialog(
                        "Enter width:"));
                    height = Double.parseDouble(JOptionPane.showInputDialog(
                        "Enter height:"));
                    shapes[i] = new Rectangle(x, y, width, height);
                break;
                default:
                    x = Double.parseDouble(JOptionPane.showInputDialog(
                        "Enter your x coordinate:"));
                    y = Double.parseDouble(JOptionPane.showInputDialog(
                        "Enter your y coordinate:"));
                    shapes[i] = new Circle(x, y);
            }
        }
    }
}

```

This Java code defines a class **DrawingCircleAndRectangle** that prompts the user to enter the number of shapes. It then creates an array of **Shape** objects of size **n**. For each shape, it asks the user if they want to draw a circle or a rectangle. If the answer is "Rectangle" or "rectangle", it asks for the x and y coordinates, width, and height, and creates a **Rectangle** object. Otherwise, it asks for the x and y coordinates and creates a **Circle** object.



```

y = Double.parseDouble(JOptionPane.showInputDialog(
    "Enter your y coordinate:"));
width = Double.parseDouble(JOptionPane.showInputDialog(
    "Enter your width of rectangle:"));
height = Double.parseDouble(JOptionPane.showInputDialog(
    "Enter height of Rectangle: "));
shapes[i] = new Rectangle(x, y, width, height);
break;
case "Circle":
case "circle":
    x = Double.parseDouble(JOptionPane.showInputDialog(
        "Enter your x coordinate:"));
    y = Double.parseDouble(JOptionPane.showInputDialog(
        "Enter your y coordinate:"));
    width = Double.parseDouble(JOptionPane.showInputDialog(
        "Enter your width of Circle:"));
    height = Double.parseDouble(JOptionPane.showInputDialog(
        "Enter height of Circle:"));
    → shapes[i] = new Ellipse(x, y, width, height);
    break;

```

```

default:
JOptionPane.showConfirmDialog(null, "Wrong option typed!!!!");
i--;
} //end of switch case

```

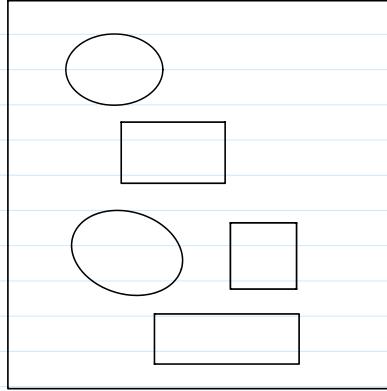
//end of for loop

```

} //end of for loop
{
    Canvas canvas = Canvas.getInstance();
    for (Shape s: shapes){
        canvas.show(s);
    }
}

```

depending upon
shape objects
different shapes
will be drawn on
the canvas.



Rect(10, 10, 100, 70)
Circle(10,120,120,120)
Circle(150,120,150,150)
Rect(10,200, 200, 250)

The parent reference variable ^{of type} (Abstract class or Interface), can keep the reference of its child objects during the execution of code. This is called runtime polymorphism.

heap (runtime objects) → 0xB2