**GIC Geometry Puzzle App**

This command line application was written in Java and uses Gradle as the build tool.

The following versions of Java and Gradle should be the minimum versions to run the application, although other versions should most likely work too:

Java 17.0.10+

Gradle 8.6+

README.txt overview and project structure:

A screenshot of a computer program

Description automatically generatedA screenshot of a computer

Description automatically generated

**A screenshot of a computer program

Description automatically generated**

**Output #1 ([1] Create a custom shape)**

A screenshot of a computer program

Description automatically generated

**Output #2 ([2] Generate a random shape)**

A screenshot of a computer program

Description automatically generated

**Test Cases**

The following screens showcase the different classes and public methods that have test coverage. These tests are meant to ensure correct output and return values of methods etc.

A screenshot of a test results

Description automatically generated

A screenshot of a test results

Description automatically generated

package puzzle;  
  
import org.junit.jupiter.api.Test;  
  
import static org.junit.jupiter.api.Assertions.*assertEquals*;  
  
public class PointTest {  
  
 @Test  
 public void testPointValues() {  
 Point p = new Point(1, 2);  
 *assertEquals*(1, p.x());  
 *assertEquals*(2, p.y());  
 }  
  
}

A screenshot of a test results

Description automatically generated

package puzzle;  
  
import org.junit.jupiter.api.Test;  
  
import java.io.ByteArrayOutputStream;  
import java.io.PrintStream;  
  
import static org.junit.jupiter.api.Assertions.*assertEquals*;  
  
public class PromptTest {  
  
 @Test  
 public void testStart() {  
 ByteArrayOutputStream outContent = new ByteArrayOutputStream();  
 System.*setOut*(new PrintStream(outContent));  
  
 Prompt.*start*();  
  
 StringBuilder stringBuilder = new StringBuilder();  
 stringBuilder.append("Welcome to the GIC geometry puzzle app\n");  
 stringBuilder.append("[1] Create a custom shape\n");  
 stringBuilder.append("[2] Generate a random shape\n");  
 stringBuilder.append("Thank you for playing the GIC geometry puzzle app\n");  
 stringBuilder.append("Have a nice day!\n");  
 String mainOutput = stringBuilder.toString();  
  
 *assertEquals*(mainOutput, outContent.toString());  
  
 System.*setOut*(System.*out*);  
 }  
  
}

A screenshot of a test results

Description automatically generated

package puzzle;  
  
import org.junit.jupiter.api.Test;  
  
import java.io.ByteArrayOutputStream;  
import java.io.PrintStream;  
  
import static org.junit.jupiter.api.Assertions.*assertEquals*;  
  
public class PuzzleAppTest {  
  
 @Test  
 public void testMainMethod() {  
 ByteArrayOutputStream outContent = new ByteArrayOutputStream();  
 System.*setOut*(new PrintStream(outContent));  
  
 PuzzleApp.*main*(new String[]{});  
  
 StringBuilder stringBuilder = new StringBuilder();  
 stringBuilder.append("Welcome to the GIC geometry puzzle app\n");  
 stringBuilder.append("[1] Create a custom shape\n");  
 stringBuilder.append("[2] Generate a random shape\n");  
 stringBuilder.append("Thank you for playing the GIC geometry puzzle app\n");  
 stringBuilder.append("Have a nice day!\n");  
 String mainOutput = stringBuilder.toString();  
  
 *assertEquals*(mainOutput, outContent.toString());  
  
 System.*setOut*(System.*out*);  
 }  
  
}

A screenshot of a test results

Description automatically generated

package puzzle;  
  
import org.junit.jupiter.api.Test;  
  
import java.io.ByteArrayOutputStream;  
import java.io.PrintStream;  
  
import static org.junit.jupiter.api.Assertions.\*;  
  
public class ShapeTest {  
  
 @Test  
 public void testClear() {  
 Shape shape = new Shape();  
 shape.addPoint(new Point(1, 1));  
 shape.addPoint(new Point(5, 1));  
 shape.addPoint(new Point(5, 5));  
 *assertEquals*(shape.clear(), 0);  
 }  
  
 @Test  
 public void testGenerateRandom() throws Exception {  
 Shape shape = new Shape();  
 shape.generateRandom();  
 *assertTrue*(shape.isValidConvex());  
 }  
  
 @Test  
 public void testAddPoint() {  
 Shape shape = new Shape();  
 *assertTrue*(shape.addPoint(new Point(1, 1)));  
 *assertTrue*(shape.addPoint(new Point(5, 1)));  
 *assertTrue*(shape.addPoint(new Point(5, 5)));  
 *assertFalse*(shape.addPoint(new Point(4, 0)));  
 *assertFalse*(shape.addPoint(new Point(1, 1)));  
 }  
  
 @Test  
 public void testPrint() {  
 ByteArrayOutputStream outContent = new ByteArrayOutputStream();  
 System.*setOut*(new PrintStream(outContent));  
  
 Shape shape = new Shape();  
  
 shape.print();  
  
 *assertEquals*("", outContent.toString());  
  
 *assertTrue*(shape.addPoint(new Point(1, 1)));  
 *assertTrue*(shape.addPoint(new Point(5, 1)));  
 *assertTrue*(shape.addPoint(new Point(5, 5)));  
  
 shape.print();  
  
 *assertEquals*("1:(1,1)\n2:(5,1)\n3:(5,5)\n", outContent.toString());  
  
 System.*setOut*(System.*out*);  
 }  
  
 @Test  
 public void testIsValidConvex() {  
 Shape shape = new Shape();  
 shape.addPoint(new Point(1, 1));  
 shape.addPoint(new Point(5, 1));  
  
 *assertFalse*(shape.isValidConvex());  
  
 shape.addPoint(new Point(5, 5));  
  
 *assertTrue*(shape.isValidConvex());  
 }  
  
 @Test  
 public void testIsWithinShape() {  
 Shape shape = new Shape();  
 shape.addPoint(new Point(0, 0));  
 shape.addPoint(new Point(10, 0));  
 shape.addPoint(new Point(5, 10));  
 *assertTrue*(shape.isValidConvex());  
 *assertTrue*(shape.isWithinShape(new Point(0, 0)));  
 *assertTrue*(shape.isWithinShape(new Point(5, 5)));  
 *assertTrue*(shape.isWithinShape(new Point(3, 8)));  
 *assertFalse*(shape.isWithinShape(new Point(11, 0)));  
 *assertFalse*(shape.isWithinShape(new Point(5, 11)));  
 }  
  
}