```java
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package lab3.solution;

/**
 *
 * @author vanting
 */
public class Circle {

    double radius = 0;
    static int numOfCircle = 0;

    Circle(int radius) {
        this.radius = radius;
        Circle.numOfCircle++;
    }

    double getArea() {
        return radius * radius * Math.PI;
    }

    public static void main(String[] args) {
        Circle circle1 = new Circle(1);
        Circle circle2 = new Circle(2);
        Circle circle3 = new Circle(3);
        System.out.printf("Total %d circles created.\n", Circle.numOfCircle);
    }
}

```



```java
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package lab3.solution;

/**
 *
 * @author vanting
 */
public class CreateArrayOfCircle {

    public static void main(String[] args) {
        Circle[] circles = new Circle[3];
        for (int i = 0; i < 3; i++) {
            // circles[i] = new Circle(i);
            double area = circles[i].getArea();
            System.out.println(area);
        }
    }
}

```

```java
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package lab3.solution;

import java.util.Formatter;

/**
 *
 * @author vanting
 */
public class Matrix {

    private int[][] elements;

    public Matrix(int[][] e) {
        this.elements = e;
    }

    public int getElement(int row, int col) {
        return elements[row][col];
    }

    public void setElement(int row, int col, int value) {
        elements[row][col] = value;
    }

    public boolean add(Matrix other) {

        int row1 = this.elements.length;
        int col1 = this.elements[0].length;
        int row2 = other.elements.length;
        int col2 = other.elements[0].length;

        if (row1 == row2 && col1 == col2) {
            for (int i = 0; i < row1; i++) {
                for (int j = 0; j < col1; j++) {
                    this.elements[i][j] += other.elements[i][j];
                }
            }
            return true;
        } else {
            return false;
        }
    }

    public Matrix multiply(Matrix other) {

        int row1 = this.elements.length;
        int col1 = this.elements[0].length;
        int row2 = other.elements.length;
        int col2 = other.elements[0].length;

        int sum = 0;
        int multiply[][] = new int[row1][col2];

        if (col1 == row2) {
            for (int i = 0; i < row1; i++) {
```



```java
                for (int j = 0; j < col2; j++) {
                    for (int k = 0; k < col1; k++) {
                        sum = sum + elements[i][k] * other.elements[k][j];
                    }
                    multiply[i][j] = sum;
                    sum = 0;
                }
            }
            return new Matrix(multiply);
        } else
            return null;
    }

    @Override
    public String toString() {

        StringBuilder sb = new StringBuilder();
        Formatter fmt = new Formatter(sb);
        int row = this.elements.length;
        int col = this.elements[0].length;

        for (int i = 0; i < row; i++) {
            for (int j = 0; j < col; j++) {
                fmt.format("[%2d]", this.elements[i][j]);
            }
            fmt.format("%n");
        }

        return sb.toString();
    }

}
```

```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package lab3.solution;

/**
 * Regular Polygon.
 *
 * @author Van
 */
public class RegularPolygon {

    // Declare and initialize default values.
    private int n = 3;
    private double side = 1.0;

    // Create constructor with default values.
    public RegularPolygon() {
    }

    // Create constructor with new n and side values.
    public RegularPolygon(int n, double side) {
        this.setN(n);
        this.setSide(side);
    }

    // get n method
    public int getN() {
        return this.n;
    }

    // set n method.
    public void setN(int n) {
        if (n >= 3)
            this.n = n;
        else
            System.err.println("The number of edges must be greater than three.");
    }

    // get side method.
    public double getSide() {
        return this.side;
    }

    // set side method.
    public void setSide(double side) {
        if (side >= 0)
            this.side = side;
        else
            System.err.println("The side length must be greater than or equal to
zero.");
    }

    // Calculate Perimeter.
    public double getPerimeter() {
        return this.n * this.side;
    }
```



```java

    // Calculate Area.
    public double getArea() {
        return (this.n * Math.pow(this.side, 2)) / (4 * Math.tan(Math.PI / this.n));
    }

    // Representation method of RegularPolygon object.
    @Override
    public String toString() {
        return "The Regular Polygon n: " + this.n + ", side: " + this.side + ", Area:
"
                + String.format("%.2f", this.getArea()) + ", Perimeter: " +
this.getPerimeter();
    }

}

```

```java
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package lab3.solution;

/**
 *
 * @author vanting
 */
public class TestMatrix {

    public static void main(String[] args) {
        // initialize both matrices
        Matrix m1 = new Matrix(new int[][] { { 1, 2 }, { 3, 4 } });
        Matrix m2 = new Matrix(new int[][] { { 5, 6 }, { 0, 0 } }); // anonymous
array
        m2.setElement(1, 0, 7);
        m2.setElement(1, 1, 8);

        System.out.println("Matrix m1:");
        System.out.println(m1);

        System.out.println("Matrix m2:");
        System.out.println(m2);

        System.out.println("Result of m1 + m2 -> m1:");
        if (m1.add(m2))
            System.out.println(m1);
        else
            System.out.println("Invalid matrix size.");

        System.out.println("Result of m1 x m2:");

        Matrix m3 = m1.multiply(m2);
        if (m3 != null)
            System.out.println(m3);
        else
            System.out.println("Invalid matrix size.");
    }
}

```



```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package lab3.solution;

/**
 *
 * @author Van
 */
public class TestRegularPolygon {

    public static void main(String[] args) {

        // Create RegularPolygon object.
        RegularPolygon reg0 = new RegularPolygon();
        RegularPolygon reg1 = new RegularPolygon(6, 4);

        // Print RegularPolygon object values.
        System.out.println(reg0);
        System.out.println(reg1);

        // reg0.setN(2);
        // reg0.setSide(0);
    }

}

```

```java
package lab3.solution;

public class TestSwap {

    public static void main(String[] args) {
        int num1 = 1;
        int num2 = 2;
        Int num3 = new Int(3);
        Int num4 = new Int(4);

        System.out.println("Before invoking the swap method, num1 is " + num1 + " and num2 is " + num2);
        swap(num1, num2);
        System.out.println("After invoking the swap method, num1 is " + num1 + " and num2 is " + num2);

        System.out.println("\n");

        System.out.println("Before invoking the swap method, num3 is " + num3 + " and num4 is " + num4);
        swap(num3, num4);
        System.out.println("After invoking the swap method, num3 is " + num3 + " and num4 is " + num4);
    }

    /**
     * Swap two variables
     */
    public static void swap(int n1, int n2) {
        int temp = n1;
        n1 = n2;
        n2 = temp;
    }

    public static void swap(Int n3, Int n4) {
        int temp = n3.value;
        n3.value = n4.value;
        n4.value = temp;
    }

    // wrapper class
    private static class Int {

        public int value;

        Int(int n) {
            value = n;
        }
        /*
         * @Override public String toString() { return String.valueOf(value); }
         */
    }
}

```