

```
1 /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5 package lab3.solution;
6
7 /**
8  *
9  * @author vanting
10 */
11 public class Circle {
12
13     double radius = 0;
14     static int numOfCircle = 0;
15
16     Circle(int radius) {
17         this.radius = radius;
18         Circle.numOfCircle++;
19     }
20
21     double getArea() {
22         return radius * radius * Math.PI;
23     }
24
25     public static void main(String[] args) {
26         Circle circle1 = new Circle(1);
27         Circle circle2 = new Circle(2);
28         Circle circle3 = new Circle(3);
29         System.out.printf("Total %d circles created.\n", Circle.numOfCircle);
30     }
31 }
32
```

```
1 /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5 package lab3.solution;
6
7 /**
8  *
9  * @author vanting
10 */
11
12 public class CreateArrayOfCircle {
13
14     public static void main(String[] args) {
15         Circle[] circles = new Circle[3];
16         for (int i = 0; i < 3; i++) {
17             // circles[i] = new Circle(i);
18             double area = circles[i].getArea();
19             System.out.println(area);
20         }
21     }
22 }
23
```

```
1  /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5  package lab3.solution;
6
7  import java.util.Formatter;
8
9  /**
10 *
11 * @author wanting
12 */
13 public class Matrix {
14
15     private int[][] elements;
16
17     public Matrix(int[][] e) {
18         this.elements = e;
19     }
20
21     public int getElement(int row, int col) {
22         return elements[row][col];
23     }
24
25     public void setElement(int row, int col, int value) {
26         elements[row][col] = value;
27     }
28
29     public boolean add(Matrix other) {
30
31         int row1 = this.elements.length;
32         int col1 = this.elements[0].length;
33         int row2 = other.elements.length;
34         int col2 = other.elements[0].length;
35
36         if (row1 == row2 && col1 == col2) {
37             for (int i = 0; i < row1; i++) {
38                 for (int j = 0; j < col1; j++) {
39                     this.elements[i][j] += other.elements[i][j];
40                 }
41             }
42             return true;
43         } else {
44             return false;
45         }
46     }
47
48     public Matrix multiply(Matrix other) {
49
50         int row1 = this.elements.length;
51         int col1 = this.elements[0].length;
52         int row2 = other.elements.length;
53         int col2 = other.elements[0].length;
54
55         int sum = 0;
56         int multiply[][] = new int[row1][col2];
57
58         if (col1 == row2) {
59             for (int i = 0; i < row1; i++) {
```

```
60     for (int j = 0; j < col2; j++) {
61         for (int k = 0; k < col1; k++) {
62             sum = sum + elements[i][k] * other.elements[k][j];
63         }
64         multiply[i][j] = sum;
65         sum = 0;
66     }
67 }
68 return new Matrix(multiply);
69 } else
70     return null;
71 }
72
73 @Override
74 public String toString() {
75
76     StringBuilder sb = new StringBuilder();
77     Formatter fmt = new Formatter(sb);
78     int row = this.elements.length;
79     int col = this.elements[0].length;
80
81     for (int i = 0; i < row; i++) {
82         for (int j = 0; j < col; j++) {
83             fmt.format("[%2d]", this.elements[i][j]);
84         }
85         fmt.format("\n");
86     }
87
88     return sb.toString();
89 }
90
91 }
92 }
```

2021/12/10 上午2:57	RegularPolygon.java	2021/12/10 上午2:57	RegularPolygon.java
1 /*	2 * To change this license header, choose License Headers in Project Properties.	59	// Calculate Area.
3 * To change this template file, choose Tools Templates	4 * and open the template in the editor.	60	public double getArea() {
5 */	6 package lab3.solution;	61	return (this.n * Math.pow(this.side, 2)) / (4 * Math.tan(Math.PI / this.n));
7	8 /**	62	}
9 * Regular Polygon.	10 * @author Van	63	
11 * @author Van	12 */	64	// Representation method of RegularPolygon object.
13 public class RegularPolygon {	14	65	@Override
15	16 // Declare and initialize default values.	66	public String toString() {
17 private int n = 3;	18 private double side = 1.0;	67	return "The Regular Polygon n: " + this.n + ", side: " + this.side + ", Area:
19	20 // Create constructor with default values.	68	"
21 public RegularPolygon() {	22 }	69	+ String.format("%.2f", this.getArea()) + ", Perimeter: " +
23	24 // Create constructor with new n and side values.	70	this.getPerimeter();
25 public RegularPolygon(int n, double side) {	26 this.setN(n);	71	}
27 this.setSide(side);	28 }	72	}
29	30 // get n method	73	
31 public int getN() {	32 return this.n;		
33 }	34		
35 // set n method.	36 public void setN(int n) {		
37 if (n >= 3)	38 this.n = n;		
39 else	40 System.err.println("The number of edges must be greater than three.");		
41 }	42		
43 // get side method.	44 public double getSide() {		
45 return this.side;	46 }		
47	48 // set side method.		
49 public void setSide(double side) {	50 if (side >= 0)		
51 this.side = side;	52 else		
53	54 System.err.println("The side length must be greater than or equal to		
55 zero.");	56 }		
57	58		
59	60 // Calculate Perimeter.		
61	62 public double getPerimeter() {		
63	64 return this.n * this.side;		
65	66 }		

```
1 /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5
6 package lab3.solution;
7
8 /**
9  *
10  * @author vanting
11  */
12 public class TestMatrix {
13
14     public static void main(String[] args) {
15         // initialize both matrices
16         Matrix m1 = new Matrix(new int[][] { { 1, 2 }, { 3, 4 } });
17         Matrix m2 = new Matrix(new int[][] { { 5, 6 }, { 0, 0 } }); // anonymous
18         array
19
20         m2.setElement(1, 0, 7);
21         m2.setElement(1, 1, 8);
22
23         System.out.println("Matrix m1:");
24         System.out.println(m1);
25
26         System.out.println("Matrix m2:");
27         System.out.println(m2);
28
29         System.out.println("Result of m1 + m2 -> m1:");
30         if (m1.add(m2))
31             System.out.println(m1);
32         else
33             System.out.println("Invalid matrix size.");
34
35         System.out.println("Result of m1 x m2:");
36
37         Matrix m3 = m1.multiply(m2);
38         if (m3 != null)
39             System.out.println(m3);
40         else
41             System.out.println("Invalid matrix size.");
42     }
43 }
```

```
1 /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6 package lab3.solution;
7
8 /**
9  *
10  * @author Van
11  */
12 public class TestRegularPolygon {
13
14     public static void main(String[] args) {
15
16         // Create RegularPolygon object.
17         RegularPolygon reg0 = new RegularPolygon();
18         RegularPolygon reg1 = new RegularPolygon(6, 4);
19
20         // Print RegularPolygon object values.
21         System.out.println(reg0);
22         System.out.println(reg1);
23
24         // reg0.setN(2);
25         // reg0.setSide(0);
26     }
27
28 }
29
```

```
1 package lab3.solution;
2
3 public class TestSwap {
4
5     public static void main(String[] args) {
6         int num1 = 1;
7         int num2 = 2;
8         Int num3 = new Int(3);
9         Int num4 = new Int(4);
10
11         System.out.println("Before invoking the swap method, num1 is " + num1 + " and
num2 is " + num2);
12         swap(num1, num2);
13         System.out.println("After invoking the swap method, num1 is " + num1 + " and
num2 is " + num2);
14
15         System.out.println("\n");
16
17         System.out.println("Before invoking the swap method, num3 is " + num3 + " and
num4 is " + num4);
18         swap(num3, num4);
19         System.out.println("After invoking the swap method, num3 is " + num3 + " and
num4 is " + num4);
20     }
21
22     /**
23      * Swap two variables
24      */
25     public static void swap(int n1, int n2) {
26         int temp = n1;
27         n1 = n2;
28         n2 = temp;
29     }
30
31     public static void swap(Int n3, Int n4) {
32         int temp = n3.value;
33         n3.value = n4.value;
34         n4.value = temp;
35     }
36
37     // wrapper class
38     private static class Int {
39
40         public int value;
41
42         Int(int n) {
43             value = n;
44         }
45         /*
46          * @Override public String toString() { return String.valueOf(value); }
47          */
48     }
49 }
50
```