LAB # 10

Task No 01: Write a program for exam department which provide abstract class and method of Exam type which contains general methods related to exams and can be used by different department for conducting exams.

Code:

Main:

```
package lab10;

public class Lab10 {

    public static void main(String[] args) {

        MCQExam mcq = new MCQExam("Math", 60, 100, 20, 4);
        EssayExam essay = new EssayExam("English", 90, 100, "The impact of social
media on youth");

        mcq.displayDetails();
        mcq.conductExam();

        System.out.println();

        essay.displayDetails();
        essay.conductExam();
    }
}
```

ExamType (Abstract Class):

```
package lab10;

public abstract class ExamType {

    String subject;
    int duration;
    int marks;

    public ExamType(String subject, int duration, int marks) {
        this.subject = subject;
        this.duration = duration;
        this.marks = marks;
    }
    abstract void conductExam();
    public void displayDetails() {
        System.out.println("Subject: " + subject);
        System.out.println("Duration: " + duration + " minutes");
        System.out.println("Marks: " + marks);
    }
}
```

MCQExam (Child Class):

```
package lab10;

public class MCQExam extends ExamType {
    int questions;
    int options;

    public MCQExam(String subject, int duration, int marks, int questions, int
options) {
        super(subject, duration, marks);
        this.questions = questions;
        this.options = options;
    }
    public void conductExam() {
        System.out.println("Conducting MCQ exam with " + questions + " questions and
" + options + " options each.");
    }
}
```

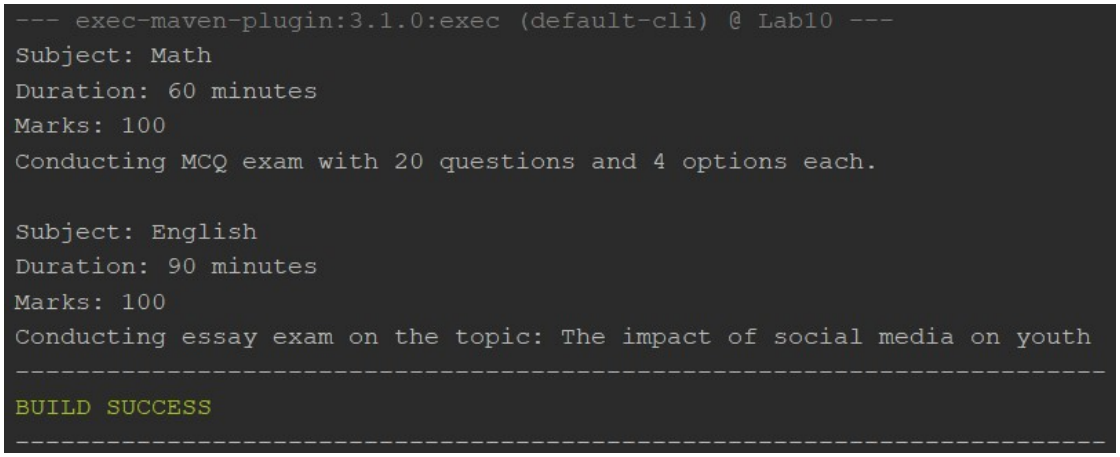
EssayExam (Child Class):

```
package lab10;

public class EssayExam extends ExamType {

    String topic;
    public EssayExam(String subject, int duration, int marks, String topic) {
        super(subject, duration, marks);
        this.topic = topic;
    }
    public void conductExam() {
        System.out.println("Conducting essay exam on the topic: " + topic);
    }
}
```

Output:



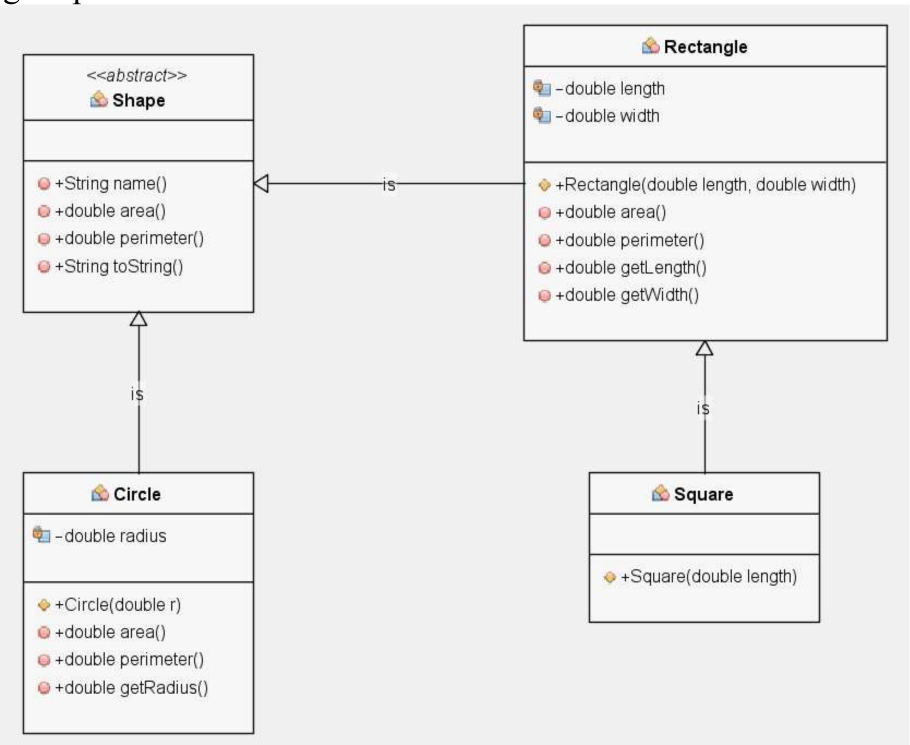
```
--- exec-maven-plugin:3.1.0:exec (default-cli) @ Lab10 ---
Subject: Math
Duration: 60 minutes
Marks: 100
Conducting MCQ exam with 20 questions and 4 options each.

Subject: English
Duration: 90 minutes
Marks: 100
Conducting essay exam on the topic: The impact of social media on youth
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
```

Task No 02: Implement the UML diagram given below. Also Design and implement a subclass “EquilateralTriangle” having a double variable side denoting the three sides of the equilateral triangle [Note that since all the 3 sides are equal, the constructor will have only one parameter]. The area and perimeter of the equilateral triangle are given as follows:

Area = $\frac{1}{4}*\sqrt{3}*(side)^2$
Perimeter = $3*side$

Provide accessor methods for the sides. Test your class using the TestShapes and DownCastingShapes classes.



Code:

Main:

```
package lab10task02;

public class Lab10task02 {

    public static void main(String[] args) {

        Shape[] randomShapes = TestShapes.createShape();

        for (int i = 0; i < randomShapes.length; i++) {
            System.out.println(randomShapes[i]);
            if (randomShapes[i] instanceof Circle) {
                System.out.println("Radius=" + ((Circle)
randomShapes[i]).getRadius());
            } else if (randomShapes[i] instanceof Square) {
                System.out.println("Length=" + ((Square)
randomShapes[i]).getLength());
            } else if (randomShapes[i] instanceof Rectangle) {
                System.out.println("Length=" + ((Rectangle)
randomShapes[i]).getLength()
                        + "\nWidth=  " + ((Rectangle) randomShapes[i]).getWidth());
            } else if (randomShapes[i] instanceof EquilateralTriangle) {
                System.out.println("Side=" + ((EquilateralTriangle)
randomShapes[i]).getside());
            }
        }
    }
}
```

Shape (Parent):

```
package lab10task02;

abstract class Shape {

    public String name() {
        return getClass().getSimpleName();
    }
    public abstract double area();
    public abstract double perimeter();
    public String toString() {
        return "\n" + name() + "\n  Area=" + area() + "\nPerimeter=" + perimeter();
    }
}
```

TestShapes:

```
package lab10task02;
import java.util.*;
public class TestShapes {

    public static Shape[] createShape() {
        final int SIZE = 5;
        final double DIMENSION = 100;
        final int NUMBEROFSHAPES = 4;
        //final int assigner = 3;


        Random generator = new Random();

        Shape[] randomShapes = new Shape[generator.nextInt(SIZE) + 1];

        for (int i = 0; i < randomShapes.length; i++) {

            switch (assigner) {
                case 0:
                    randomShapes[i] = new Rectangle(generator.nextDouble() *
DIMENSION, generator.nextDouble() * DIMENSION);
                    break;
                case 1:
                    randomShapes[i] = new Circle(generator.nextDouble() * DIMENSION);
                    break;
                case 2:
                    randomShapes[i] = new Square(generator.nextDouble() * DIMENSION);
                    break;
                case 3:
                    randomShapes[i] = new EquilateralTriangle(generator.nextDouble()
* DIMENSION);
                    break;
            }
        }
        return randomShapes;
    }
    public static void main(String[] args) {
        Shape[] randomShapes = TestShapes.createShape();
        for (int i = 0; i < randomShapes.length; i++) {
            System.out.println(randomShapes[i].toString());
        }
    }
}
```

Rectangle (Child):

```
package lab10task02;

public class Rectangle extends Shape {

    private double length;
    private double width;

    public Rectangle(double length, double width) {
        this.length = length;
        this.width = width;
    }
    public double area() {
        return length * width;
    }
    public double perimeter() {
        return 2 * (length + width);
    }
    public double getLength() {
        return length;
    }
    public double getWidth() {
        return width;
    }
}
```

Square (Child):

```
package lab10task02;

public class Square extends Rectangle {

    public Square(double length) {
        super(length, length);
    }
}
```

Circle (Child):

```
package lab10task02;

public class Circle extends Shape {

    private double radius;

    public Circle(double r) {
        radius = r;
    }
    public double area() {
        return Math.PI * (radius * radius);
    }
    public double perimeter() {
        return 2.0 * Math.PI * radius;
    }
    public double getRadius() {
        return radius;
    }
}
```

EquilateralTriangle (Child):

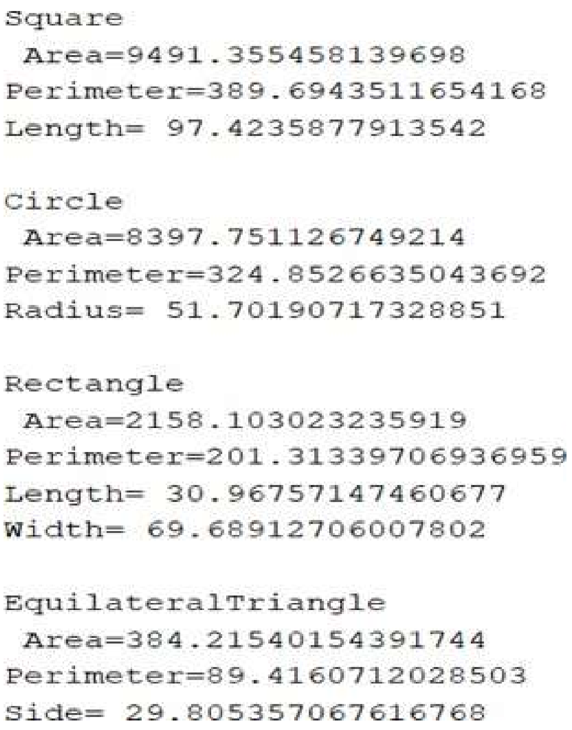
```
package lab10task02;

public class EquilateralTriangle extends Shape {

    private double side;

    public EquilateralTriangle(double side) {
        this.side = side;
    }
    public double area() {
        return 0.25 * 1.73 * side * side;
    }
    public double perimeter() {
        return 3 * side;
    }
    public double getside() {
        return side;
    }
}
```

Output:



```
Square
 Area=9491.355458139698
Perimeter=389.6943511654168
Length= 97.4235877913542

Circle
 Area=8397.751126749214
Perimeter=324.8526635043692
Radius= 51.70190717328851

Rectangle
 Area=2158.103023235919
Perimeter=201.31339706936959
Length= 30.96757147460677
Width= 69.68912706007802

EquilateralTriangle
 Area=384.21540154391744
Perimeter=89.4160712028503
Side= 29.805357067616768
```