ASSIGNMENT -4

Package and Interface

a) Create a user-defined package named 'geoshapes'. In this package, define the following classes with necessary methods such as area and perimeter.

i) Circle

ii) Rectangle

iii) Triangle

iv) Sphere

import the package and demonstrate the use of objects of each defined class

CODE :-

```
package geoshapes;
public class Circle {
   private double radius;
   public Circle(double radius) {
      this.radius = radius;
   }
   public double area() {
      return Math.PI * radius * radius;
   }
   public double perimeter() {
      return 2 * Math.PI * radius;
   }
   @Override
   public String toString() {
      return "Circle: \nRadius : " + radius;
   }
```

```
}

package geoshapes;
public class Rectangle {
   private double width;
   private double height;
   public Rectangle(double width, double height) {
      this.width = width;
      this.height = height;
   }
   public double area() {
      return width * height;
   }
   public double perimeter() {
      return 2 * (width + height);
   }
   @Override
   public String toString() {
      return "Rectangle :\nWidth : " + width + " , Height : " + height;
   }
}

package geoshapes;
public class Triangle {
   private double a;
   private double b;
   private double c;
   public Triangle(double a, double b, double c) {
      this.a = a;
```

```
        this.b = b;
        this.c = c;
    }
    public double perimeter() {
        return a + b + c;
    }
    public double area() {
        double s = perimeter() / 2.0;
        return Math.sqrt(s * (s - a) * (s - b) * (s - c));
    }
    @Override
    public String toString() {
        return "Triangle : \nSides : " + a + " , " + b +  " , " + c;
    }
}

package geoshapes;
public class Sphere {
    private double radius;
    public Sphere(double radius) {
        this.radius = radius;
    }
    public double surfaceArea() {
        return 4 * Math.PI * radius * radius;
    }
    public double volume() {
        return (4.0 / 3.0) * Math.PI * radius * radius * radius;
    }
    @Override
    public String toString() {
        return "Sphere : \nRadius : " + radius;
    }
}

import geoshapes.Circle;
import geoshapes.Rectangle;
import geoshapes.Sphere;
import geoshapes.Triangle;
import java.util.Scanner;
public class GeoDemo {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int choice;
        System.out.println("Geoshapes Menu:");
        System.out.println("1. Circle (area & perimeter)");
        System.out.println("2. Rectangle (area & perimeter)");
        System.out.println("3. Triangle (area & perimeter)");
        System.out.println("4. Sphere (surface area & volume)");
        System.out.println("5. Exit");
        do {
            System.out.print("Enter choice: ");
            choice = sc.nextInt();
            switch(choice) {
                case 1:
                    System.out.print("Enter radius: ");
                    double r = sc.nextDouble();
                    Circle c = new Circle(r);
                    System.out.println(c);
                    System.out.println("Area: " + c.area());
                    System.out.println("Perimeter: " + c.perimeter());
                    break;
                case 2:
                    System.out.print("Enter width: ");
                    double w = sc.nextDouble();
                    System.out.print("Enter height: ");
                    double h = sc.nextDouble();
                    Rectangle rect = new Rectangle(w, h);
                    System.out.println(rect);
                    System.out.println("Area: " + rect.area());
                    System.out.println("Perimeter: " + rect.perimeter());
                    break;
                case 3:
                    System.out.print("Enter side a: ");
                    double a = sc.nextDouble();
                    System.out.print("Enter side b: ");
                    double b = sc.nextDouble();
                    System.out.print("Enter side c: ");
                    double c3 = sc.nextDouble();
                    Triangle t = new Triangle(a, b, c3);
                    System.out.println(t);
                    System.out.println("Perimeter: " + t.perimeter());
                    System.out.println("Area: " + t.area());
                    break;
                case 4:
                    System.out.print("Enter radius: ");
                    double rs = sc.nextDouble();
                    Sphere s = new Sphere(rs);
```

```
                System.out.println(s);
                System.out.println("Surface area: " + s.surfaceArea());
                System.out.println("Volume: " + s.volume());
                break;
            case 5:
                System.out.println("Exiting...");
                break;
            default:
                System.out.println("Invalid choice");
          }
        } while(choice != 5);
        sc.close();
    }
}
```

---

b) Create a user-defined package named as 'coordinates'. Define two Classes namely Cartesian and Polar in coordinate package with all necessary methods. Define methods toPolar() and toCartesian() in Cartesian and Polar class respectively for converting in to respective types of coordinate. Demonstrate the conversion in both way by importing the package and using method of these objects

<u>CODE</u> :-

```
package coordinates;
public class Cartesian {
    private double x;
    private double y;
    public Cartesian(double x, double y) {
        this.x = x;
        this.y = y;
    }
    public double getX() { return x; }
    public double getY() { return y; }
    public Polar toPolar() {
        double r = Math.sqrt(x * x + y * y);
        double theta = Math.atan2(y, x);
        return new Polar(r, theta);
    }
    @Override
    public String toString() {
        return "Cartesian " + x + " , " + y;
    }
}

package coordinates;
public class Polar {
    private double r;
    private double theta;
    public Polar(double r, double theta) {
        this.r = r;
        this.theta = theta;
    }
    public double getR() { return r; }
    public double getTheta() { return theta; }
    public Cartesian toCartesian() {
        double x = r * Math.cos(theta);
        double y = r * Math.sin(theta);
        return new Cartesian(x, y);
    }
    @Override
    public String toString() {
        return "Polar : "  + r + " , "  + theta;
    }
}

import coordinates.Cartesian;
import coordinates.Polar;
import java.util.Scanner;
public class CoordinateDemo {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int choice;
        System.out.println("Coordinate Conversion Menu:");
        System.out.println("1. Cartesian -> Polar");
        System.out.println("2. Polar -> Cartesian");
        System.out.println("3. Exit");
        do {
            System.out.print("Enter choice: ");
            choice = sc.nextInt();
            switch (choice) {
                case 1:
                    System.out.print("Enter x: ");
                    double x = sc.nextDouble();
                    System.out.print("Enter y: ");
                    double y = sc.nextDouble();
                    Cartesian c = new Cartesian(x, y);
                    Polar p = c.toPolar();
                    System.out.println(c);
                    System.out.println(p);
                    break;
                case 2:
                    System.out.print("Enter r: ");
                    double r = sc.nextDouble();
                    System.out.print("Enter theta (in radians): ");
                    double theta = sc.nextDouble();
                    Polar p2 = new Polar(r, theta);
                    Cartesian c2 = p2.toCartesian();
                    System.out.println(p2);
                    System.out.println(c2);
                    break;
                case 3:
                    System.out.println("Exiting...");
                    break;
                default:
                    System.out.println("Invalid choice");
            }
        } while (choice != 3);
        sc.close();
    }
}
```

---

c) Create an interface with name 'LinearDS' with the following member and methods:

i) MAXSIZE – indicate the maximum capacity of the data structure

ii) add(element) - the method to add an element to the datastructure (as per its rule)

iii) remove() - the method to remove an element from the data structure (as per its rule)

iv) displayElement() - the method to display all the element of the data structure.

Implement your interface to define two classes MyStack and MyQueue to implement all operations of a stack and queue using LinearDS respectivley.

CODE :-

```
import java.util.Scanner;
interface LinearDs{
   int MAXSIZE = 100;
   void add(int element);
   int remove();
   void displayElement();
}

class MyStack implements LinearDs{
   int top = -1;
   int arr[] = new int[MAXSIZE];
   public void add(int element){
      if(top == MAXSIZE - 1){
         System.out.println("Stack Overflow");
         return;
      }
      top++;
      arr[top] = element;
   }

   public int remove(){
      if(top == -1){
            System.out.println("Stack Underflow");
            return -1;
        }
        int element = arr[top];
        top--;
        return element;
    }

    public void displayElement(){
        int start = top;
        System.out.println(".Elements Of Stack are :- ");
        while(start != -1){
            System.out.print(arr[start] + " ");
            start--;
        }
    }
}

class MyQueue implements LinearDs{
    int rear = -1;
    int front = -1;
    int size = 0;
    int arr[] = new int[MAXSIZE];
    public void add(int element){
        if(size == MAXSIZE){
            System.out.println("Queue is Full");
            return;
        }
        else if(size == 0){
```

```
        front = rear = 0;
        arr[front] = element;
    }
    else{
        rear = (rear + 1) % MAXSIZE;
        arr[rear] = element;
    }
    size++;
}

public int remove(){
    if(size == 0){
        System.out.println("Queue is Empty");
        return -1;
    }
    else{
        int element = arr[front];
        front = (front + 1)  % MAXSIZE;
        size--;
        return element;
    }
}

public void displayElement(){
    System.out.println("Element Of Queue are :- ");
    if(rear > front){
        for(int i = front ; i <= rear ; i++){
            System.out.print(arr[i] + " ");
        }
        }
        else{
            for(int i = front ; i < MAXSIZE ; i++){
                System.out.print(arr[i] + " ");
            }
            for(int i = 0; i < rear ; i++){
                System.out.print(arr[i] + " ");
            }
        }
    }
}
public class LinearDataStructure{
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        MyStack stack = new MyStack();
        MyQueue queue = new MyQueue();
        int choice;
        System.out.println("Menu:");
        System.out.println("1. Push to Stack");
        System.out.println("2. Pop from Stack");
        System.out.println("3. Display Stack");
        System.out.println("4. Enqueue to Queue");
        System.out.println("5. Dequeue from Queue");
        System.out.println("6. Display Queue");
        System.out.println("7. Exit");
        do {
            System.out.print("Enter your choice: ");
            choice = sc.nextInt();
            switch(choice) {
```

```
case 1:
    System.out.print("Enter element to push: ");
    int sElem = sc.nextInt();
    stack.add(sElem);
    break;
case 2:
    int element = stack.remove();
    System.out.println("Element removed is : " + element);
    break;
case 3:
    stack.displayElement();
    System.out.println();
    break;
case 4:
    System.out.print("Enter element to enqueue: ");
    int qElem = sc.nextInt();
    queue.add(qElem);
    break;
case 5:
    int ele = queue.remove();
    System.out.println("Element removed is : " + ele);
    break;
case 6:
    queue.displayElement();
    System.out.println();
    break;
case 7:
    System.out.println("Exiting...");
    break;
```

```
                default:
                    System.out.println("Invalid choice");
            }
        } while(choice != 7);
        sc.close();
    }
}
```
