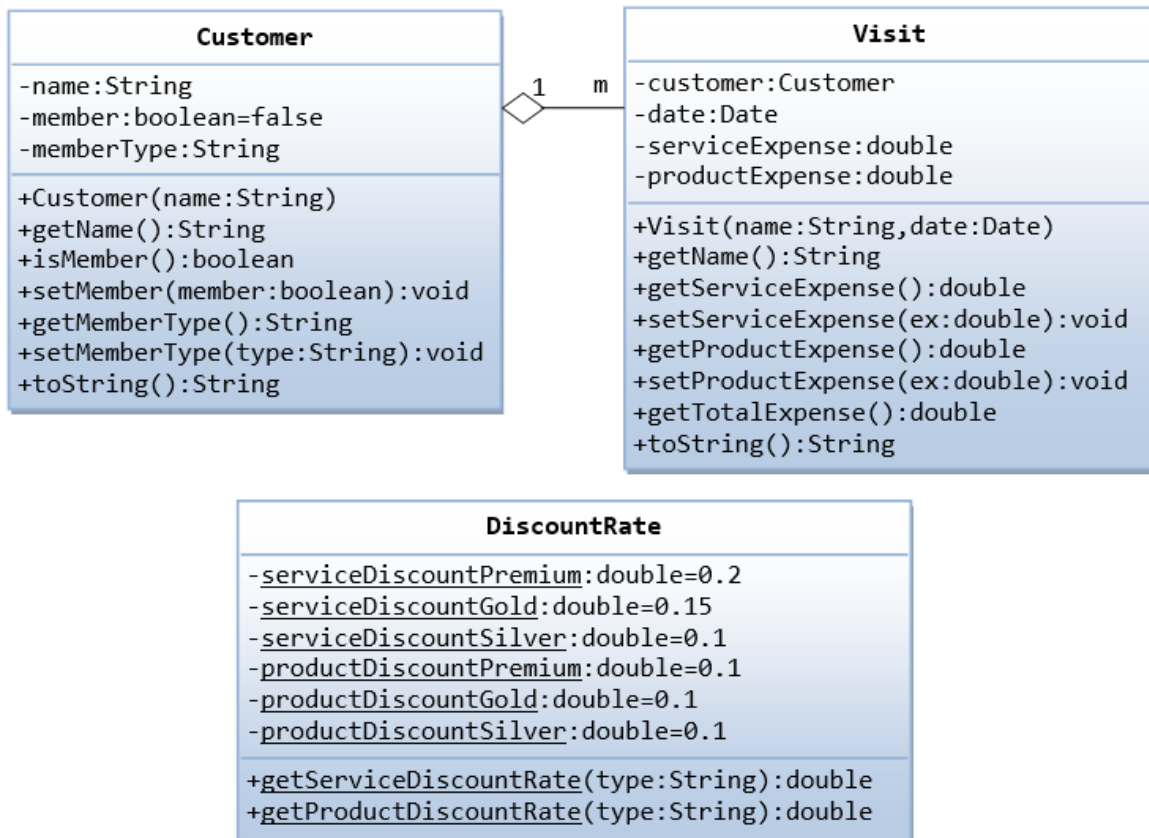


## Lab 6. OOP

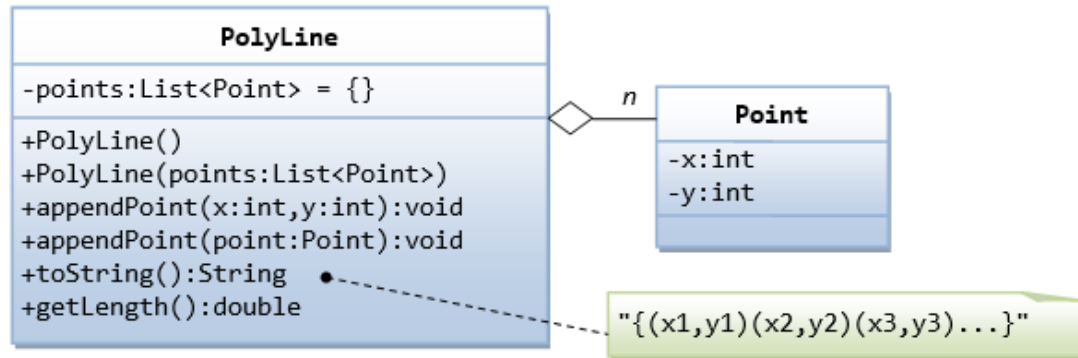
### 1 The Discount System

You are asked to write a discount system for a beauty saloon, which provides services and sells beauty products. It offers 3 types of memberships: Premium, Gold and Silver. Premium, gold and silver members receive a discount of 20%, 15%, and 10%, respectively, for all services provided. Customers without membership receive no discount. All members receives a flat 10% discount on products purchased (this might change in future). Your system shall consist of three classes: Customer, Discount and Visit, as shown in the class diagram. It shall compute the total bill if a customer purchases  $x$  of products and  $y$  of services, for a visit. Also write a test program to exercise all the classes.



The class DiscountRate contains only static variables and methods (underlined in the class diagram).

## 2 Polyline of Points with ArrayList



A polyline is a line with segments formed by points. Let's use the ArrayList (dynamically allocated array) to keep the points, but upcast to List in the instance variable. (Take note that array is of fixed-length, and you need to set the initial length).



```

1  public class Point {
    private int x;
3   private int y;
    public Point(int x, int y) { ..... }
5   public int getX() { ..... }
    public int getY() { ..... }
7   public void setX(int x) { ..... }
    public void setY(int y) { ..... }
9   public int[] getXY() { ..... }
    public void setXY(int x, int y) { ..... }
11  public String toString() { ..... }
    public double distance(Point another) { ..... }
13 }

```



```

1  import java.util.*;
    public class PolyLine {
3   private List<Point> points;    // List of Point instances

5   // Constructors
    public PolyLine() { // default constructor
7       points = new ArrayList<Point>(); // implement with ArrayList
    }
9   public PolyLine(List<Point> points) {
        this.points = points;
11  }

```



```

13 // Append a point (x, y) to the end of this polyline
    public void appendPoint(int x, int y) {
15         Point newPoint = new Point(x, y);
            points.add(newPoint);
17     }

19 // Append a point instance to the end of this polyline
    public void appendPoint(Point point) {
21         points.add(point);
    }
23

25 // Return {(x1,y1)(x2,y2)(x3,y3)....}
    public String toString() {
        // Use a StringBuilder to efficiently build the return String
27         StringBuilder sb = new StringBuilder("{}");
            for (Point aPoint : points) {
29                 sb.append(aPoint.toString());
            }
31         sb.append("}");
        return sb.toString();
33     }

35 // Return the total length of this polyline
    public double getLength() { ..... }
37 }

```



```

1  /*
    * A Test Driver for the PolyLine class.
    */
3  import java.util.*;
5  public class TestPolyLine {
        public static void main(String[] args) {
7            // Test default constructor and toString()
                PolyLine l1 = new PolyLine();
9                System.out.println(l1); // {}

11           // Test appendPoint()
                l1.appendPoint(new Point(1, 2));
13                l1.appendPoint(3, 4);
                l1.appendPoint(5, 6);
15                System.out.println(l1); // {(1,2)(3,4)(5,6)}

17           // Test constructor 2
                List<Point> points = new ArrayList<Point>();
19                points.add(new Point(11, 12));
                points.add(new Point(13, 14));
21                PolyLine l2 = new PolyLine(points);

```



```
23      System.out.println(12);    // {(11,12)(13,14)}  
      }
```