Program 1;

This program is used to implement a polyline, and getting a polyline’s length, with different points as the property.

Here is some pseudocode that will help explain the programs code:

1. Start
2. Import java’s array list and list classes
3. Define a class for point, with properties x (int) and y (int)
4. Define methods for the point class:
5. Constructor to set x and y
6. A toString method that will return a string representation of the point
7. Define a class for polyline, with a list property, to hold points
8. Define methods for the polyline class:
   1. Constructor to initialise lists
   2. Methods to append new and existing points onto the polyline
   3. Getter for length (Double) (will calculate and return length of polyline)
   4. A toString that will return the string representation of all the points on the polyline
9. Define a class for the main, which will test the classes:
   1. Create instances of point and polyline
   2. Add points onto the polyline
   3. Print out the polyline and its length
10. End

Code:

A screenshot of a computer program

Description automatically generated

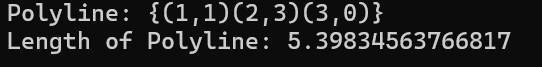
A computer screen shot of white text

Description automatically generated

A computer screen shot of a black screen

Description automatically generated

Answer:



Program 2:

This program is used to implement a discount system for a beauty salon, where discounts depend on membership levels, including getting total expense, discount, and price after discount. You also have the ability to observe (set) and change (get) different properties including membership types, and service/product expenses.

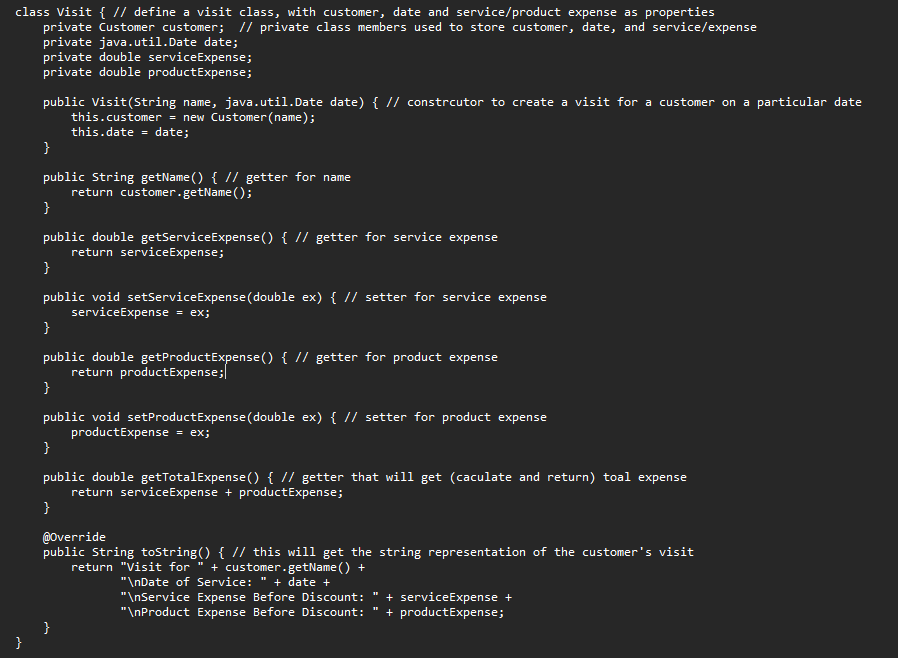
Here is some pseudocode that will help explain the programs code:

1. Start
2. Define a class for customer, with properties name (String), member (Boolean) and memberType (String) – defaults for member and memberType are included
3. Define methods for the customer class:
4. Constructor for name
5. Getter for name, and Getters and Setters for member and memberType
6. A toString method that will return a string describing the customer’s name and membership status.
7. Define a class for visit, with properties customer (String), date (Date), serviceExpense (double) and productExpense (double)
8. Define methods for the visit class:
9. Constructor for a customer’s visit
10. Getter for name, and Getters and Setters for serviceExpense and productExpense
11. Getter for totalExpense (Double) (will calculate and return the total expense of the customer’s visit)
12. A toString method that will return a string that describes the visit’s details
13. Define a discount rate class with static properties for product/service discounts for each membership type – serviceDiscountPremium (double), serviceDiscountGold (double), serviceDiscountSilver (double), productDiscountPremium (double),productDiscountGold (double), productDiscountSilver (double)
14. Define method for discount class:
    1. Getters for product and service discount rates
15. Define a class for the main, which will test the classes
    1. Creates instances of customer and visit
    2. Print out these details, alongside product/service expense before/after discount, alongside total expense after discount
16. End

Code:

A screenshot of a computer program

Description automatically generated



A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

Answer:

A screen shot of a black screen

Description automatically generated

Program 3:

This program is used to perform certain operations on different shapes, including getting a shapes area/volume, and observing (getting) and changing (setting) a shapes properties, including radius, length, colour etc. properties.

Here is some pseudocode that will help explain the programs code:

1. Start
2. Define a class for circle, with properties colour (String) and radius (Double) – defaults for these properties will be included
3. Define methods for the circle class:
   1. Constructor without parameters (it will use the defaults)
   2. Constructor with parameters (it will set colour and radius)
   3. Getters and Setters for colour and radius
   4. Getter for area (Double) (will calculate and return the area of the circle)
   5. A toString method that will return a string describing the circle
4. Define a class for cylinder, that will extend (inherit) the previous circle class, but with the additional height property (Double) – defaults for height will be included
5. Define methods for the cylinder class:
   1. Constructor without parameters (it will use the defaults)
   2. Constructor with parameters (one with inherited radius, one with that will set height and also inherits radius, and finally one that’s sets radius, and inherits height and colour)
   3. Getter and Setter for hieght
   4. Getter for Volume (Double) (will calculate and return the volume of the cylinder)
   5. A toString that will return a string describing the cylinder.
6. Define a class for the main, which will test the classes
   1. Creates instances of circle, cylinder
   2. Print out these details, alongside area and volume calculations
7. End

Code:

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A computer code with white text

Description automatically generated

Answer:

A black screen with white text

Description automatically generated

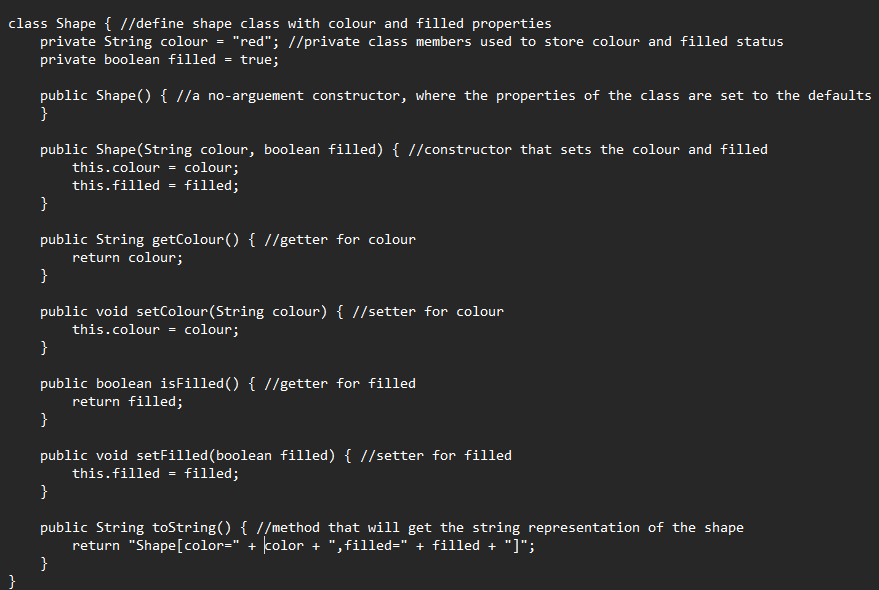
Program 4

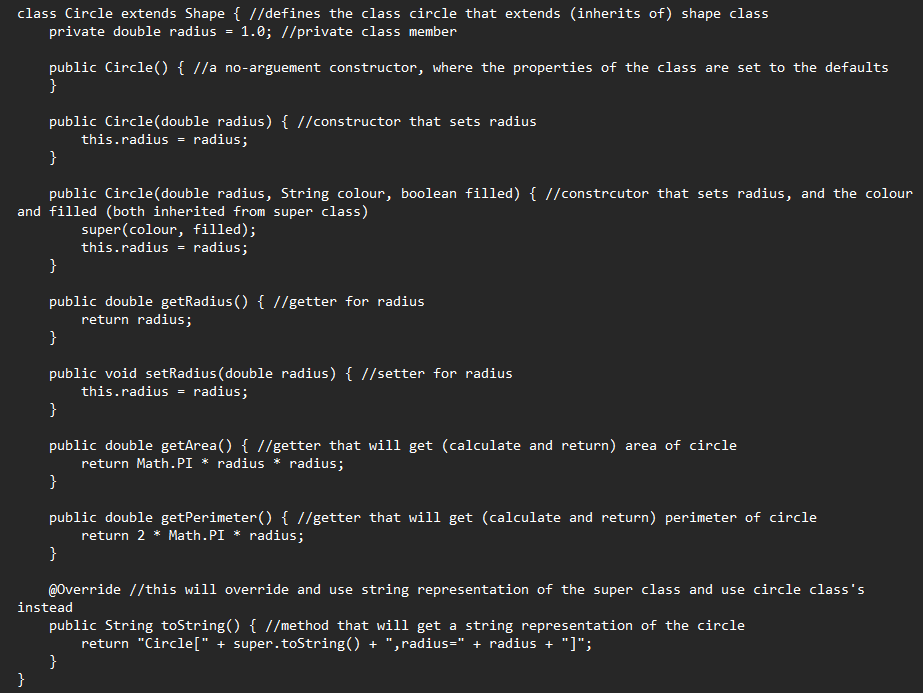
This program is used to perform certain operations on different shapes, including getting a shapes area/perimeter, and observing (getting) and changing (setting) a shapes properties, including radius, length, colour etc. properties.

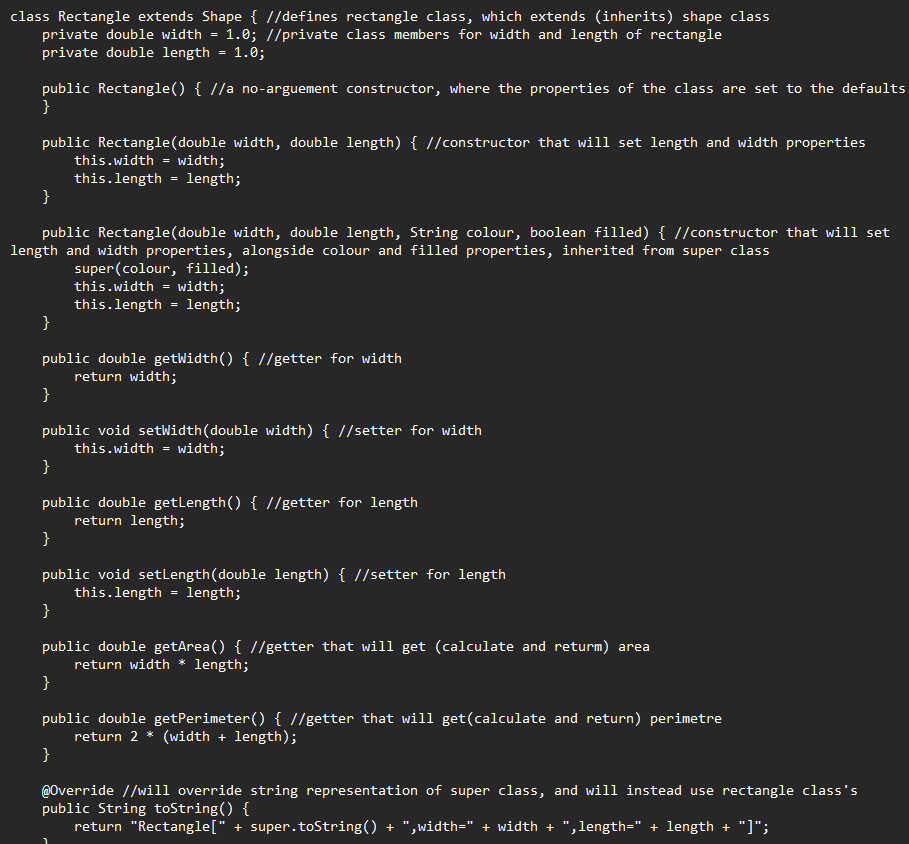
Here is some pseudocode that will help explain the programs code:

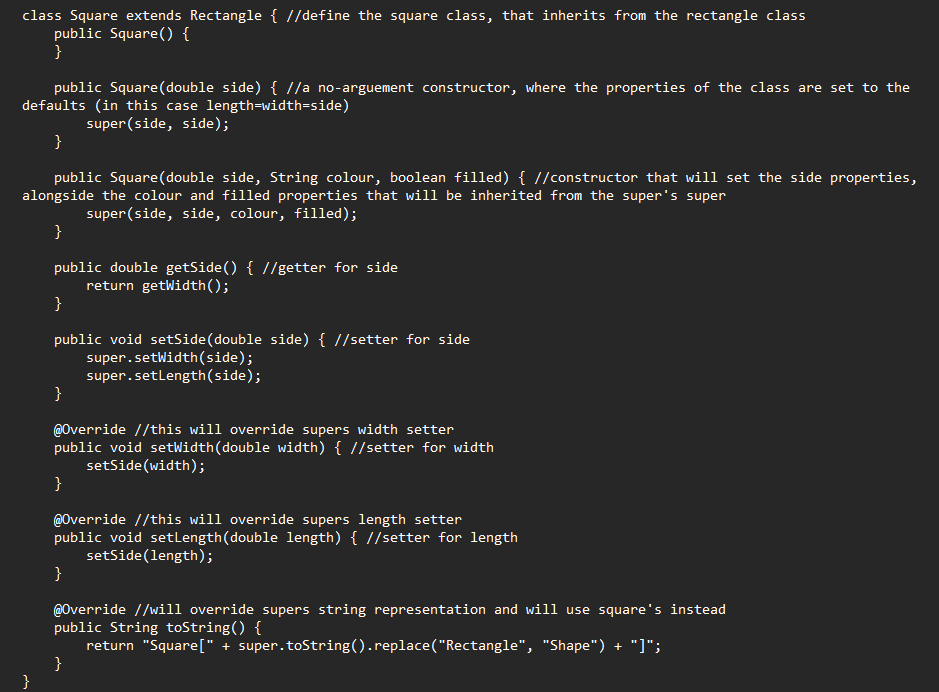
1. Start
2. Define a class for shape, with properties colour (a string colour) and filled (Boolean) – defaults for these properties will be included
3. Define methods for the shape class:
   1. Constructor without parameters (it will use the defaults)
   2. Constructor with parameters (it will set colour and filled)
   3. Getters and Setters for colour and filled
   4. A toString method that will return a string describing the shape
4. Define a class for circle, that will extend (inherit) the previous shape class, but with the additional radius property (Double) – defaults for radius will be included
5. Define methods for the circle class:
   1. Constructor without parameters (it will use the defaults)
   2. Constructor with parameters (one with set radius, and another set radius, and inherited parameters)
   3. Getter and Setter for radius
   4. Getter for area (Double) and parameter (Double) (will calculate and return the area and perimeter of the circle)
   5. A toString that will return a string describing the circle.
6. Define a class for rectangle, that will extend (inherit) the previous shape class, but with the additional length (Double) and width (Double) properties – defaults for length and width will be included
7. Define methods for the rectangle class:
   1. Constructor without parameters (it will use the defaults)
   2. Constructor with parameters (one with set width and length, and another width and length, and inherited parameters)
   3. Getters and Setters for length and width
   4. Getter for area (Double) and parameter (Double) (will calculate and return the area and perimeter of the rectangle)
   5. A toString that will return a string describing the rectangle.
8. Define a class for square, that will extend (inherit) the previous rectangle class, but with the length and width being equal (known as sides (Double)) – defaults for side will be included
9. Define methods for the square class:
   1. Constructor without parameters (it will use the defaults)
   2. Constructor with parameter side and inherited parameters
   3. Getter and Setter for side (this will override length and width)
   4. A toString that will return a string describing the square.
10. Define a class for the main, which will test the classes
    1. Creates instances of shape, circle, square, rectangle
    2. Print out these details, alongside area and perimeter calculations

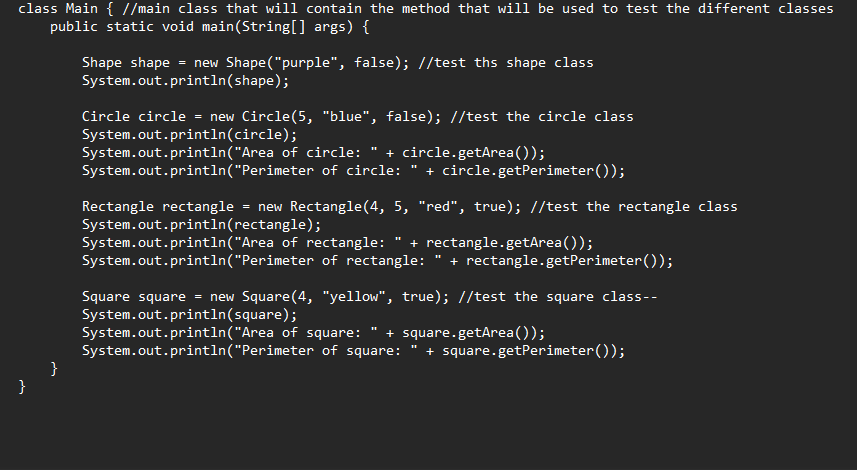
Code:











Answer:

