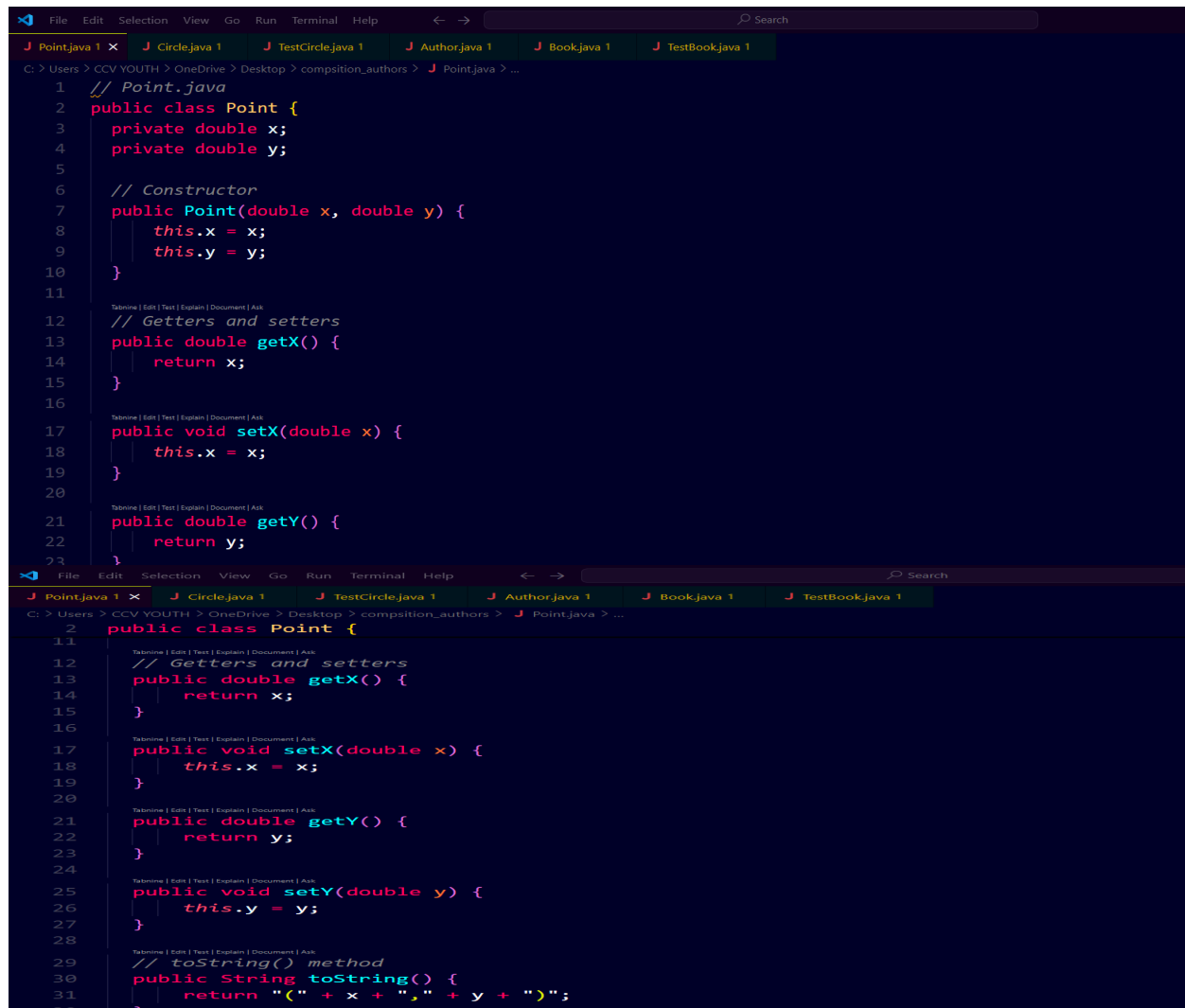


1. Point Class (Point.java)

Purpose:

The **Point** class represents a point in a 2D coordinate system with **x** and **y** coordinates. It provides getter and setter methods for these coordinates and a **toString()** method to represent the point in a readable format.



```
1 // Point.java
2 public class Point {
3     private double x;
4     private double y;
5
6     // Constructor
7     public Point(double x, double y) {
8         this.x = x;
9         this.y = y;
10    }
11
12    // Getters and setters
13    public double getX() {
14        return x;
15    }
16
17    public void setX(double x) {
18        this.x = x;
19    }
20
21    public double getY() {
22        return y;
23    }
24
25    public void setY(double y) {
26        this.y = y;
27    }
28
29    // toString() method
30    public String toString() {
31        return "(" + x + "," + y + ")";
32    }
33 }
```

Explanation:

- **Constructor:** Initializes the **x** and **y** coordinates of the point.
- **Getter and Setter Methods:** Allow access and modification of the **x** and **y** coordinates.
- **toString() Method:** Provides a string representation of the point in the form **(x,y)**.

2. Circle Class (Circle.java)

Purpose:

The **Circle** class represents a circle with a specified **radius** and a **center**, which is a **Point**. It includes methods for modifying and retrieving the radius, center, calculating the distance between two circles' centers, and calculating the area and circumference of the circle.

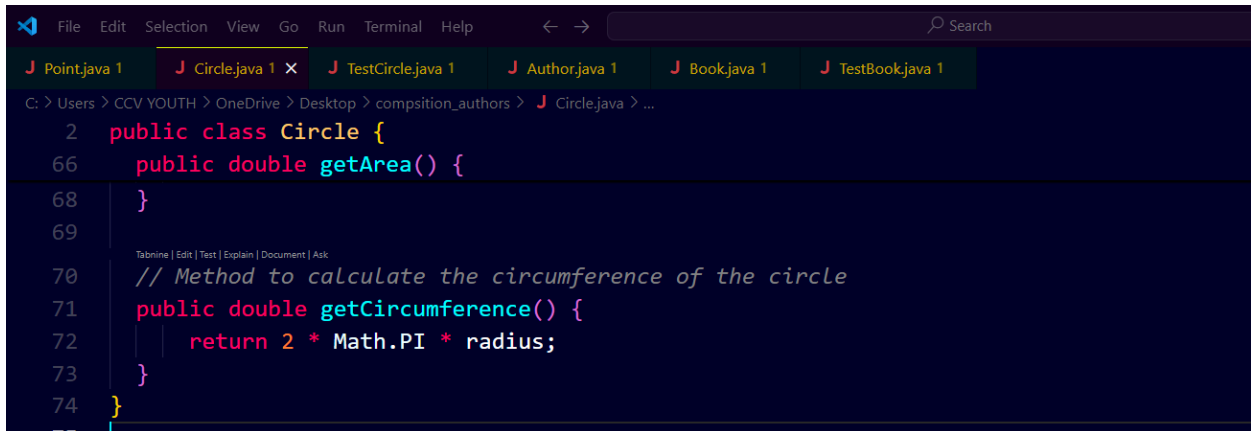
Code:



```
1  // Circle.java
2  public class Circle {
3      private double radius;
4      private Point center;
5
6      // Constructor
7      public Circle(double radius, Point center) {
8          this.radius = radius;
9          this.center = center;
10     }
11
12     // Getter and setter for radius
13     public double getRadius() {
14         return radius;
15     }
16
17     // Setter for radius
18     public void setRadius(double radius) {
19         this.radius = radius;
20     }
21
22     // Getter and setter for center
23     public Point getCenter() {
24         return center;
25     }
26 }
```

```
2 public class Circle {
25
26     public void setCenter(Point center) {
27         this.center = center;
28     }
29
30     // Getters and setters for x and y of center
31     public double getCenterX() {
32         return center.getX();
33     }
34
35     public void setCenterX(double x) {
36         center.setX(x);
37     }
38
39     public double getCenterY() {
40         return center.getY();
41     }
42
43     public void setCenterY(double y) {
44         center.setY(y);
45     }
46 }
```

```
46
47     // Set center using x and y
48     public void setCenterXY(double x, double y) {
49         center.setX(x);
50         center.setY(y);
51     }
52
53     // toString method
54     public String toString() {
55         return "Circle[center=" + center.toString() + ",radius=" + radius + "];";
56     }
57
58     // Method to calculate distance between centers of two circles
59     public double distance(Circle another) {
60         double dx = this.center.getX() - another.center.getX();
61         double dy = this.center.getY() - another.center.getY();
62         return Math.sqrt(dx * dx + dy * dy);
63     }
64
65     // Method to calculate the area of the circle
66     public double getArea() {
67         return Math.PI * radius * radius;
68     }
69 }
```



```
File Edit Selection View Go Run Terminal Help
J Point.java 1 J Circle.java 1 X J TestCircle.java 1 J Author.java 1 J Book.java 1 J TestBook.java 1
C:\Users\CCV YOUTH\OneDrive\Desktop\compition_authors\Circle.java > ...
2 public class Circle {
66 public double getArea() {
68 }
69
70 // Method to calculate the circumference of the circle
71 public double getCircumference() {
72     return 2 * Math.PI * radius;
73 }
74 }
75
```

Explanation:

- **Constructor:** Initializes the `radius` and `center` of the circle.
- **Getter and Setter Methods:** Provide access to the circle's radius and center. Separate methods are provided to access and set the x and y coordinates of the center.
- **toString() Method:** Returns a string that describes the circle, including its center's coordinates and radius.
- **distance() Method:** Computes the Euclidean distance between the centers of two circles.
- **getArea() and getCircumference() Methods:** Calculate and return the area and circumference of the circle, respectively.
-

3. Test Driver for Circle Class (TestCircle.java)

Purpose:

This test driver program demonstrates the usage of the `Circle` class by creating instances of circles, testing getter and setter methods, and calculating distances, areas, and circumferences.

```
File Edit Selection View Go Run Terminal Help
J Point.java 1 J Circle.java 1 J TestCircle.java 1 X J Author.java 1 J Book.java 1 J TestBook.java 1
C:\Users\CCV YOUTH\OneDrive\Desktop> compition_authors > J TestCircle.java > ...

1 // TestCircle.java
2 public class TestCircle {
3     public static void main(String[] args) {
4         // Create some points
5         Point p1 = new Point(x:0, y:0);
6         Point p2 = new Point(x:1, y:2);
7         Point p3 = new Point(x:4, y:5);
8
9         // Create some circles with points and radius
10        Circle c1 = new Circle(radius:1.0, p1);
11        Circle c2 = new Circle(radius:3.3, p2);
12        Circle c3 = new Circle(radius:6.6, p3);
13
14        // Output the circle information
15        System.out.println("c1: " + c1);
16        System.out.println("c2: " + c2);
17        System.out.println("c3: " + c3);
18
19        // Test Setters and Getters of c1
20        c1.setCenterXY(x:11, y:12); // Set center to (11, 12)
21        c1.setRadius(radius:13.3); // Set radius to 13.3
22        System.out.println("\nc1: " + c1); // Output updated circle
23        System.out.println("c1 is: " + c1.getCenter()); // Output center
24        System.out.println("Radius is: " + c1.getRadius()); // Output radius
}
```

```
J Point.java 1 J Circle.java 1 J TestCircle.java 1 X J Author.java 1 J Book.java 1 J TestBook.java 1
C:\Users\CCV YOUTH\OneDrive\Desktop> compition_authors > J TestCircle.java > ...

2 public class TestCircle {
3     public static void main(String[] args) {
4
5         // Set Center X and Y individually
6         c1.setCenterX(x:21);
7         c1.setCenterY(y:22);
8         System.out.println("\nc1: " + c1); // Output updated circle
9         System.out.println("c1's x is: " + c1.getCenterX()); // Output center X
10        System.out.println("c1's y is: " + c1.getCenterY()); // Output center Y
11
12        // Set Center using setCenterXY method
13        c1.setCenterXY(x:31, y:32);
14        System.out.println("\nc1: " + c1); // Output updated circle
15        System.out.println("c1's x is: " + c1.getCenterX()); // Output center X
16        System.out.println("c1's y is: " + c1.getCenterY()); // Output center Y
17
18        // Test getArea() and getCircumference()
19        System.out.println("\nArea of c1: " + c1.getArea());
20        System.out.println("Circumference of c1: " + c1.getCircumference());
21
22        // Test distance between c1 and c2
23        double dist = c1.distance(c2);
24        System.out.println("\nDistance between c1 and c2: " + dist);
25    }
26 }
```

Explanation:

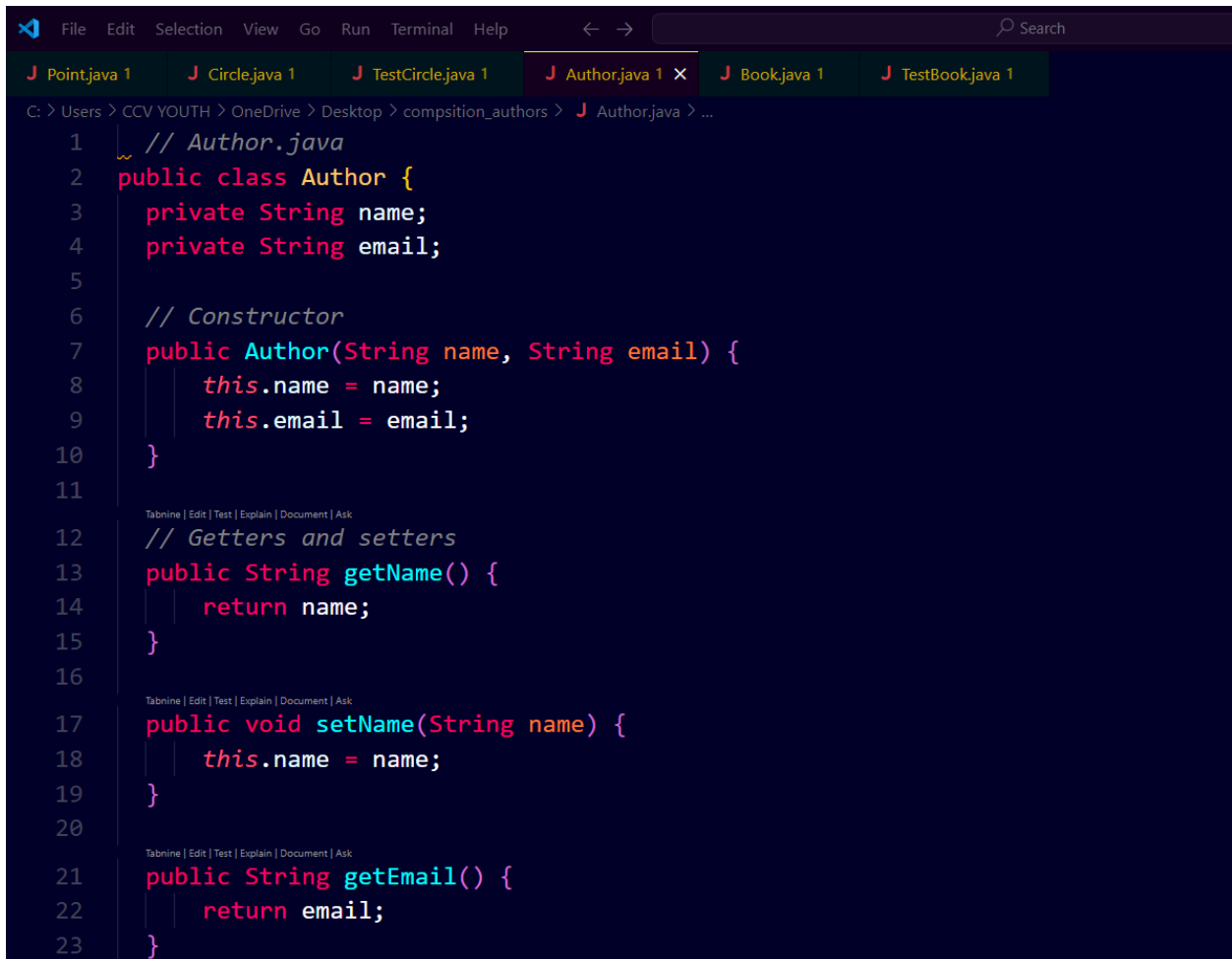
- This program tests the `Circle` class by:
 - Creating `Point` and `Circle` objects.
 - Setting and getting values for the circle's center and radius.
 - Displaying the results of area, circumference, and distance calculations.

4. Author Class (Author.java)

Purpose:

The `Author` class represents an author with a `name` and `email` address. It provides getter and setter methods for these attributes and a `toString()` method for displaying the author's information.

Code:



```
1 // Author.java
2 public class Author {
3     private String name;
4     private String email;
5
6     // Constructor
7     public Author(String name, String email) {
8         this.name = name;
9         this.email = email;
10    }
11
12    // Getters and setters
13    public String getName() {
14        return name;
15    }
16
17    public void setName(String name) {
18        this.name = name;
19    }
20
21    public String getEmail() {
22        return email;
23    }
```

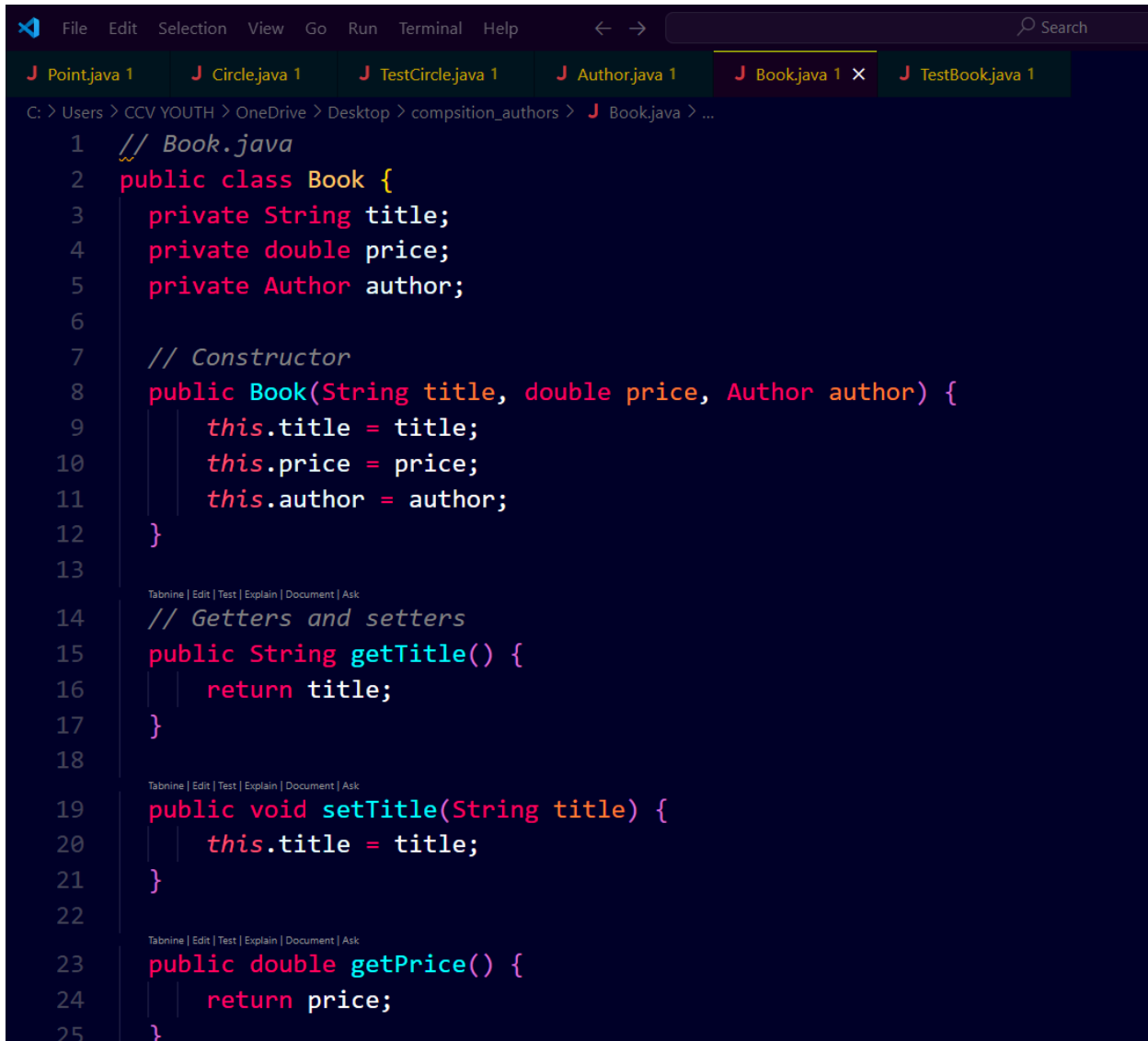
Explanation:

- **Constructor:** Initializes the `name` and `email` of the author.
- **Getter and Setter Methods:** Allow access to and modification of the `name` and `email`.
- **`toString()` Method:** Returns a string describing the author.

5. Book Class (Book.java)

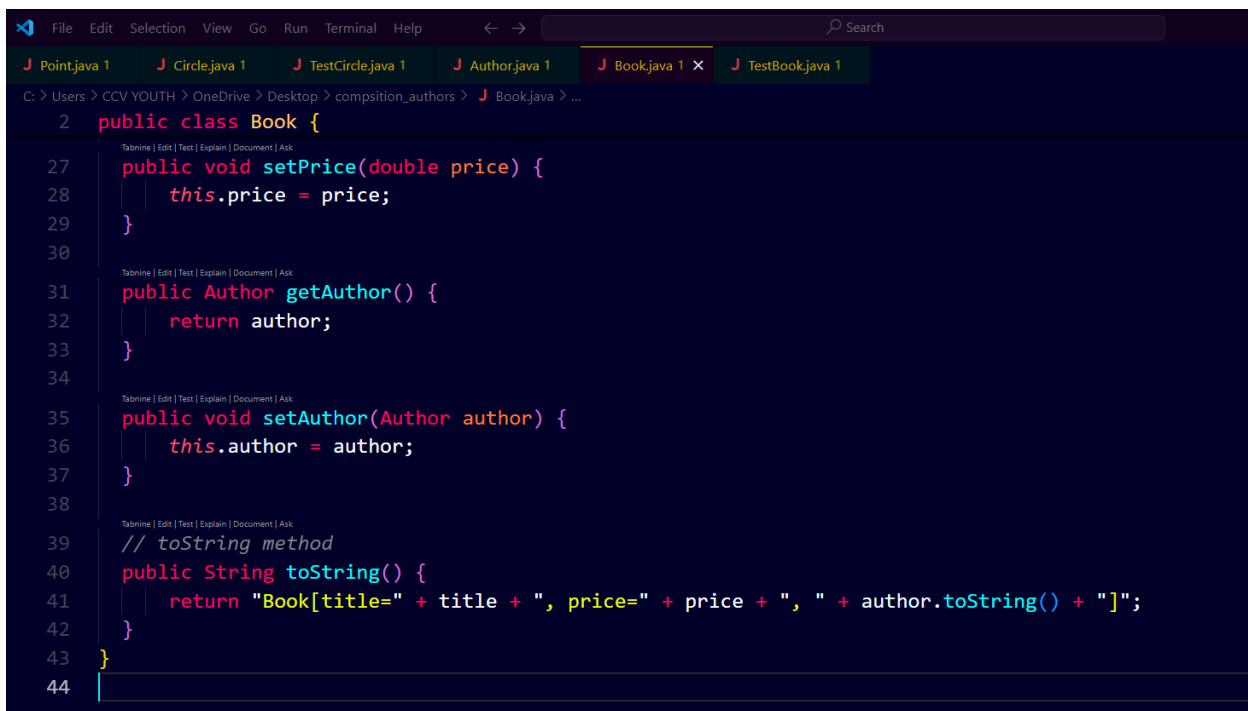
Purpose:

The `Book` class represents a book with a `title`, `price`, and an `Author` object. It includes getter and setter methods for these attributes and a `toString()` method for displaying the book's details.



This screenshot shows the first part of a Java program in an IDE. The file explorer at the top shows a project named 'compsition_authors' with several Java files. The current file is 'Book.java'. The code defines a 'Book' class with private attributes 'title', 'price', and 'author'. It includes a constructor that initializes these attributes and two methods, 'getTitle()' and 'setTitle()', for handling the title attribute. The code is as follows:

```
1  // Book.java
2  public class Book {
3      private String title;
4      private double price;
5      private Author author;
6
7      // Constructor
8      public Book(String title, double price, Author author) {
9          this.title = title;
10         this.price = price;
11         this.author = author;
12     }
13
14     // Getters and setters
15     public String getTitle() {
16         return title;
17     }
18
19     public void setTitle(String title) {
20         this.title = title;
21     }
22
23     public double getPrice() {
24         return price;
25     }
```



This screenshot shows the second part of the Java program. It continues the 'Book' class with methods for handling the 'price' and 'author' attributes, and a 'toString()' method for string representation. The code is as follows:

```
27     public void setPrice(double price) {
28         this.price = price;
29     }
30
31     public Author getAuthor() {
32         return author;
33     }
34
35     public void setAuthor(Author author) {
36         this.author = author;
37     }
38
39     // toString method
40     public String toString() {
41         return "Book[title=" + title + ", price=" + price + ", " + author.toString() + "]";
42     }
43 }
44
```


Explanation:

- **Constructor:** Initializes the book with a `title`, `price`, and an `Author` object.
- **Getter and Setter Methods:** Provide access to and modification of the book's `title`, `price`, and `author`.
- **`toString()` Method:** Returns a string describing the book's title, price, and author.

6. Test Driver for Book Class (TestBook.java)

Purpose:

This program tests the `Book` class by creating an `Author` and a `Book` and printing the details.

Code:



```
1 // TestBook.java
2 public class TestBook {
3     public static void main(String[] args) {
4         // Create an author
5         Author author = new Author(name:"J.K. Rowling", email:"jk@rowling.com");
6
7         // Create a book with the author and price
8         Book book = new Book(title:"Harry Potter", price:29.99, author);
9
10        // Output the book information
11        System.out.println(book);
12    }
13 }
14
```

Explanation:

- This program creates an `Author` and a `Book` object and prints the details of the book using the `toString()` method.

THE OUTPUT

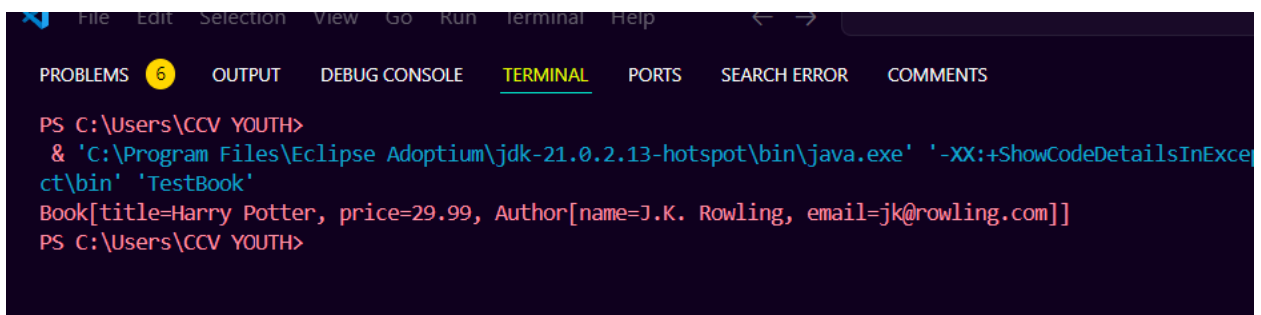
```
PS C:\Users\CCV YOUTH>
  & 'C:\Program Files\Eclipse Adoptium\jdk-21.0.2.13-hotspot\bin\java.exe' '-XX:+S
ct\bin' 'TestCircle'
c1: Circle[center=(0.0,0.0),radius=1.0]
c2: Circle[center=(1.0,2.0),radius=3.3]
c3: Circle[center=(4.0,5.0),radius=6.6]

c1: Circle[center=(11.0,12.0),radius=13.3]
c1 is: (11.0,12.0)
Radius is: 13.3

c1: Circle[center=(21.0,22.0),radius=13.3]
c1's x is: 21.0
C1's y is: 22.0

c1: Circle[center=(31.0,32.0),radius=13.3]
c1's x is: 31.0
c1's y is: 32.0
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
    Math cannot be resolved to a variable

    at Circle.getArea(Circle.java:67)
    at TestCircle.main(TestCircle.java:40)
PS C:\Users\CCV YOUTH>
```



The screenshot shows the Eclipse IDE interface with the 'Terminal' tab selected. The terminal output displays the execution of the TestBook.java program, which creates a Book object with the title 'Harry Potter', price 29.99, and author J.K. Rowling. The output is as follows:

```
PS C:\Users\CCV YOUTH>
  & 'C:\Program Files\Eclipse Adoptium\jdk-21.0.2.13-hotspot\bin\java.exe' '-XX:+ShowCodeDetailsInExce
ct\bin' 'TestBook'
Book[title=Harry Potter, price=29.99, Author[name=J.K. Rowling, email=jk@rowling.com]]
PS C:\Users\CCV YOUTH>
```

THE OUTPUT