1. Author.java

- Represents an **Author** with two attributes: `name` and `email`.
- Includes:
    - **Constructor**: Initializes `name` and `email`.
    - **Getters and Setters**: Allow controlled access to `name` and `email`.
    - **toString Method**: Provides a string representation of the author.

```java
// Author.java
public class Author {
  private String name;
  private String email;

  // Constructor
  public Author(String name, String email) {
      this.name = name;
      this.email = email;
  }

  Tabnine | Edit | Test | Explain | Document | Ask
  // Getters and setters
  public String getName() {
      return name;
  }

  Tabnine | Edit | Test | Explain | Document | Ask
  public void setName(String name) {
      this.name = name;
  }

  Tabnine | Edit | Test | Explain | Document | Ask
  public String getEmail() {
      return email;
```

---

2. Book.java

- Represents a **Book**, which has a title, price, and an associated `Author` object (composition).
- Includes:
    - **Constructor**: Initializes `title`, `price`, and an `Author` object.
    - **Getters and Setters**: Allow access to and modification of the book's attributes.

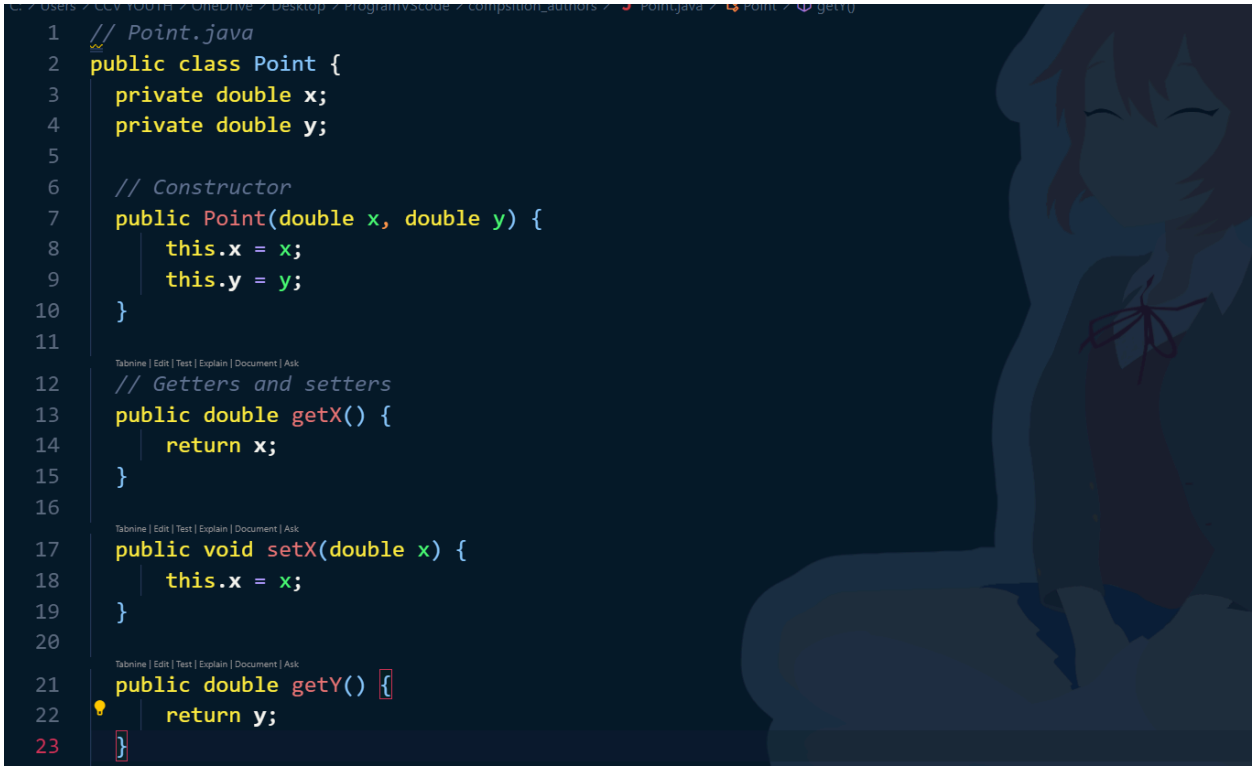
- **toString Method**: Combines book details with the associated author's information.

```java
// Book.java
public class Book {
  private String title;
  private double price;
  private Author author;

  // Constructor
  public Book(String title, double price, Author author) {
      this.title = title;
      this.price = price;
      this.author = author;
  }

  // Getters and setters
  public String getTitle() {
      return title;
  }

  public void setTitle(String title) {
      this.title = title;
  }
```

---

3. Point.java

- Represents a **Point** in a 2D plane with `x` and `y` coordinates.
- Includes:
  - **Constructor**: Initializes `x` and `y`.
  - **Getters and Setters**: Allow access to and modification of `x` and `y`.
  - **toString Method**: Provides a string representation of the point in the format `(x, y)`.



---

4. Circle.java

- Represents a **Circle** with a `radius` and a `center` (a `Point` object, showcasing composition).
- Includes:
  - **Constructor**: Initializes `radius` and `center`.
  - **Getters and Setters**: Provide access to and modification of `radius` and `center`.
  - **Additional Features**:
    - `getCenterX`, `getCenterY`: Access `x` and `y` of the center point.
    - `setCenterXY`: Updates both `x` and `y` at once.
    - `getArea`: Calculates the circle's area using π · radius2\pi \cdot \text{radius}^2π · radius2.
    - `getCircumference`: Calculates the circle's circumference using 2 · π · radius2 \cdot \pi \cdot \text{radius}2 · π · radius.
    - `distance`: Calculates the Euclidean distance between the centers of two circles.
  - **toString Method**: Combines the radius and the center's information.

```java
// Circle.java
public class Circle {
  private double radius;
  private Point center;

  // Constructor
  public Circle(double radius, Point center) {
      this.radius = radius;
      this.center = center;
  }

  Tabnine | Edit | Test | Explain | Document | Ask
  // Getter and setter for radius
  public double getRadius() {
      return radius;
  }

  Tabnine | Edit | Test | Explain | Document | Ask
  public void setRadius(double radius) {
      this.radius = radius;
  }

  Tabnine | Edit | Test | Explain | Document | Ask
  // Getter and setter for center
  public Point getCenter() {
      return center;
```

---

5. TestBook.java

- A **driver class** to test the `Book` and `Author` classes.
- **Main Method**:
  - Creates an `Author` object with `name` and `email`.
  - Creates a `Book` object, associating it with the `Author`.
  - Prints the `Book` object, triggering the `toString` methods in both `Book` and `Author`.

```java
C: > Users > CCV YOUTH > OneDrive > Desktop > ProgramVScode > compsition_authors > J TestBook.java > TestBook
// TestBook.java
public class TestBook {
    Run | Debug | Tabnine | Edit | Test | Explain | Document | Ask
    public static void main(String[] args) {
        // Create an author
        Author author = new Author(name:"J.K. Rowling", email:"jk@rowling.com");

        // Create a book with the author and price
        Book book = new Book(title:"Harry Potter", price:29.99, author);

        // Output the book information
        System.out.println(book);
    }
}
```

---

6. TestCircle.java

- A **driver class** to test the `Circle` and `Point` classes.
- **Main Method**:
    - Creates `Point` objects for circle centers.
    - Creates `Circle` objects using these points and specific radii.
    - Demonstrates:
        - `toString` method to display circle information.
        - Setters to update `radius` and center coordinates (`setCenterX`, `setCenterY`, `setCenterXY`).
        - Calculation of area, circumference, and distance between circles.
        - Outputs various results to the console.

```java
// TestCircle.java
public class TestCircle {
    public static void main(String[] args) {
        // Create some points
        Point p1 = new Point(x:0, y:0);
        Point p2 = new Point(x:1, y:2);
        Point p3 = new Point(x:4, y:5);

        // Create some circles with points and radius
        Circle c1 = new Circle(radius:1.0, p1);
        Circle c2 = new Circle(radius:3.3, p2);
        Circle c3 = new Circle(radius:6.6, p3);

        // Output the circle information
        System.out.println("c1: " + c1);
        System.out.println("c2: " + c2);
        System.out.println("c3: " + c3);

        // Test Setters and Getters of c1
        c1.setCenterXY(x:11, y:12); // Set center to (11, 12)
        c1.setRadius(radius:13.3); // Set radius to 13.3
        System.out.println("\nc1: " + c1); // Output updated circle
        System.out.println("c1 is: " + c1.getCenter()); // Output center
        System.out.println("Radius is: " + c1.getRadius()); // Output radius

```