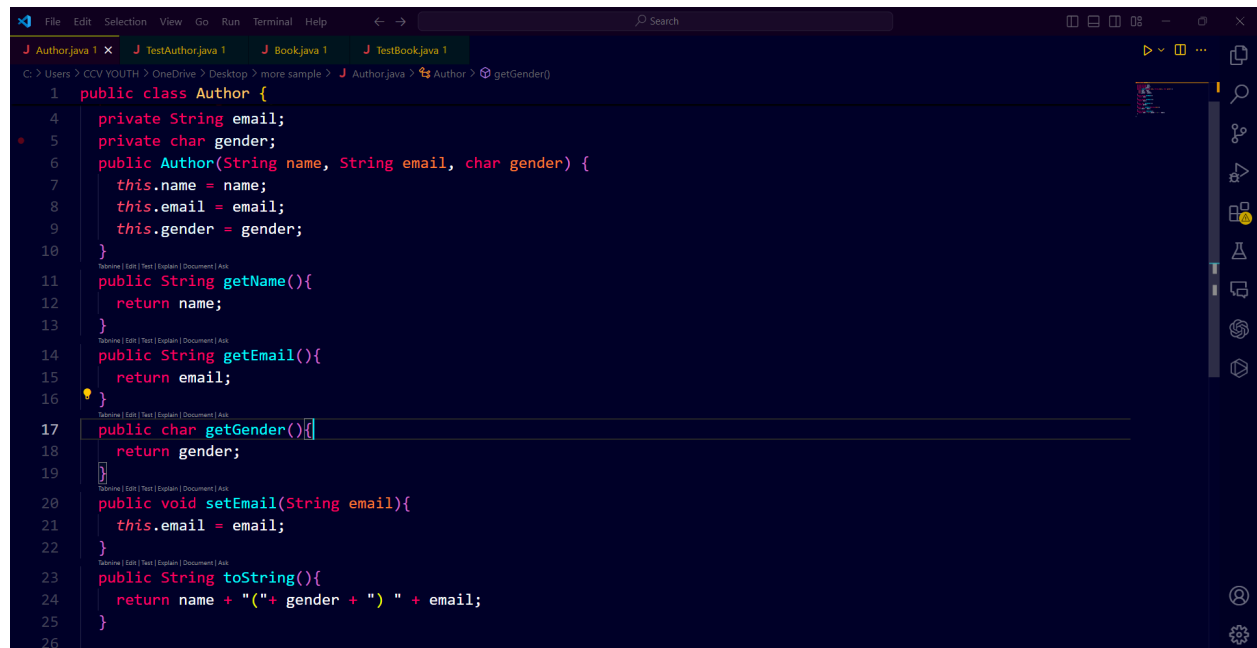


Documentation for Author class

A screenshot of an IDE window showing the source code for the Author class. The code is written in Java and includes private fields for email and gender, a constructor, and several getter and setter methods. The IDE has a dark theme and a sidebar on the right with various icons.

```
1 public class Author {  
4     private String email;  
5     private char gender;  
6     public Author(String name, String email, char gender) {  
7         this.name = name;  
8         this.email = email;  
9         this.gender = gender;  
10    }  
11    public String getName(){  
12        return name;  
13    }  
14    public String getEmail(){  
15        return email;  
16    }  
17    public char getGender(){  
18        return gender;  
19    }  
20    public void setEmail(String email){  
21        this.email = email;  
22    }  
23    public String toString(){  
24        return name + "(" + gender + ") " + email;  
25    }  
26 }
```

Fields:

- `name` (String): The name of the author.
- `email` (String): The email address of the author.
- `gender` (char): The gender of the author ('M' for male, 'F' for female, or other characters).

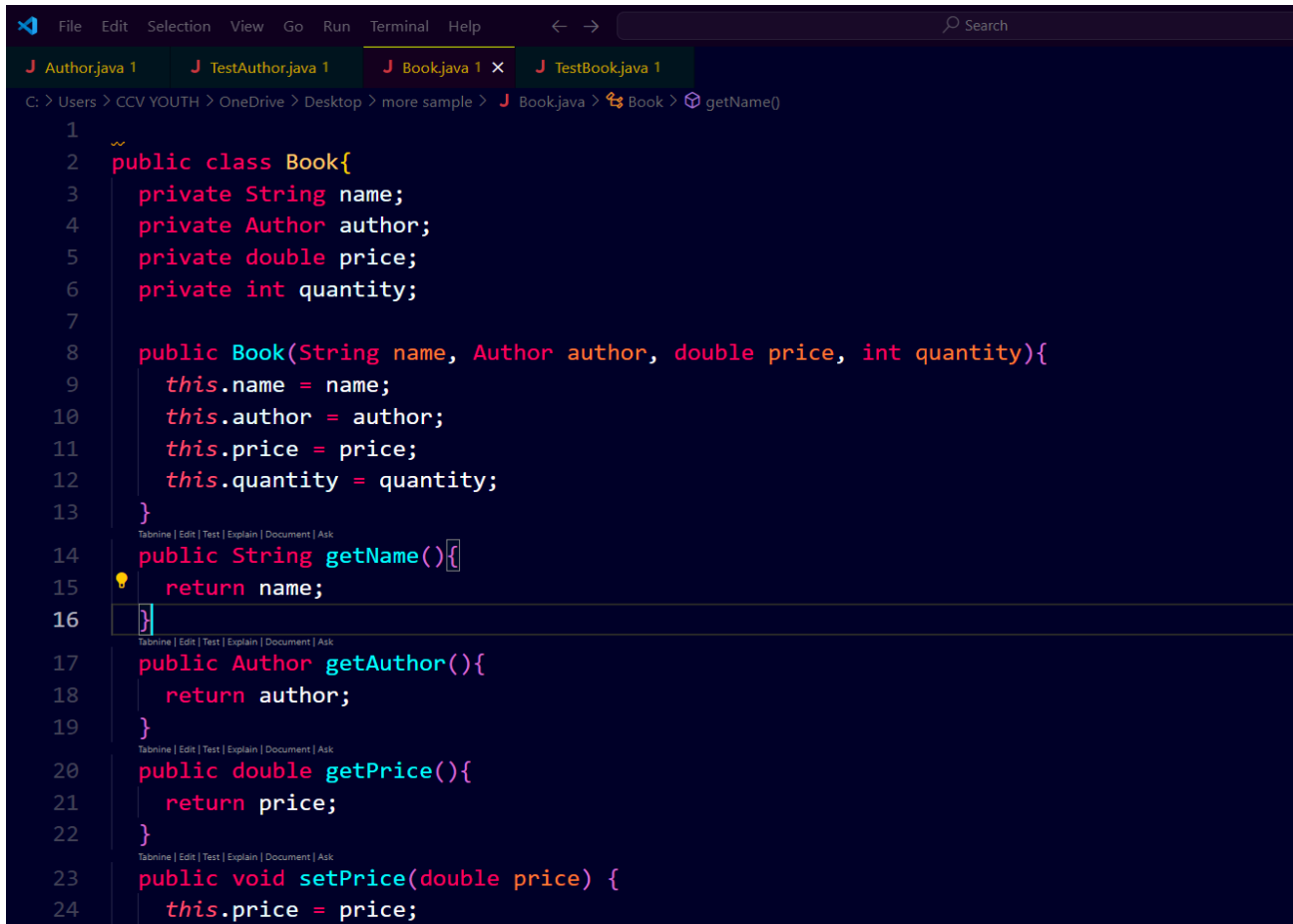
Constructor:

- `Author(String name, String email, char gender)`: Initializes an `Author` object with a name, email, and gender.

Methods:

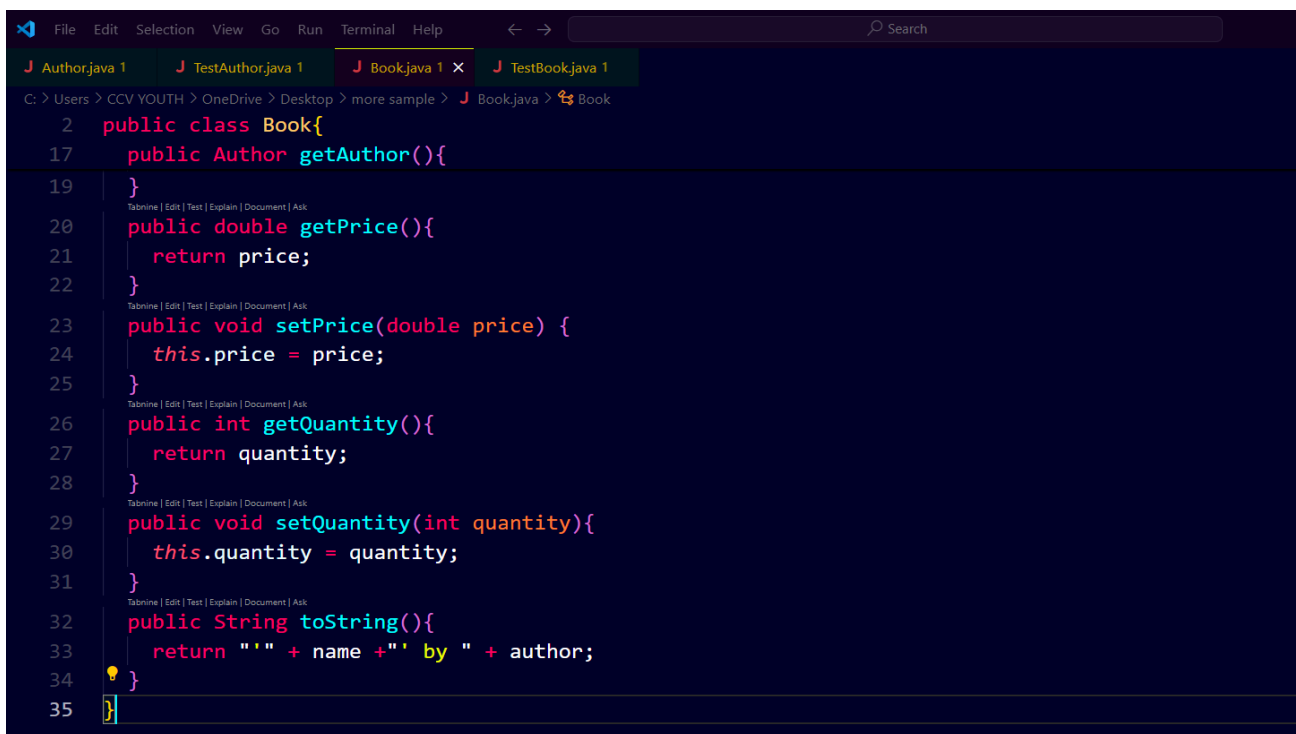
- `getName()`: Returns the name of the author.
- `getEmail()`: Returns the email address of the author.
- `getGender()`: Returns the gender of the author.
- `setEmail(String email)`: Allows you to update the author's email address.
- `toString()`: Returns a string representation of the `Author` object, showing the author's name, gender, and email.

Documentation for **Book** class:



The screenshot shows an IDE window with the file `Book.java` open. The code defines the `Book` class with the following attributes and methods:

```
1  ~
2  public class Book{
3      private String name;
4      private Author author;
5      private double price;
6      private int quantity;
7
8      public Book(String name, Author author, double price, int quantity){
9          this.name = name;
10         this.author = author;
11         this.price = price;
12         this.quantity = quantity;
13     }
14     public String getName(){
15         return name;
16     }
17     public Author getAuthor(){
18         return author;
19     }
20     public double getPrice(){
21         return price;
22     }
23     public void setPrice(double price) {
24         this.price = price;
```



The screenshot continues the `Book.java` file from the previous one, showing the following methods:

```
2     public class Book{
17     public Author getAuthor(){
19     }
20     public double getPrice(){
21         return price;
22     }
23     public void setPrice(double price) {
24         this.price = price;
25     }
26     public int getQuantity(){
27         return quantity;
28     }
29     public void setQuantity(int quantity){
30         this.quantity = quantity;
31     }
32     public String toString(){
33         return "" + name + " by " + author;
34     }
35 }
```

Fields:

- **name** (String): The title of the book.
- **author** (Author): The **Author** object representing the book's author.
- **price** (double): The price of the book.
- **quantity** (int): The number of copies of the book available.

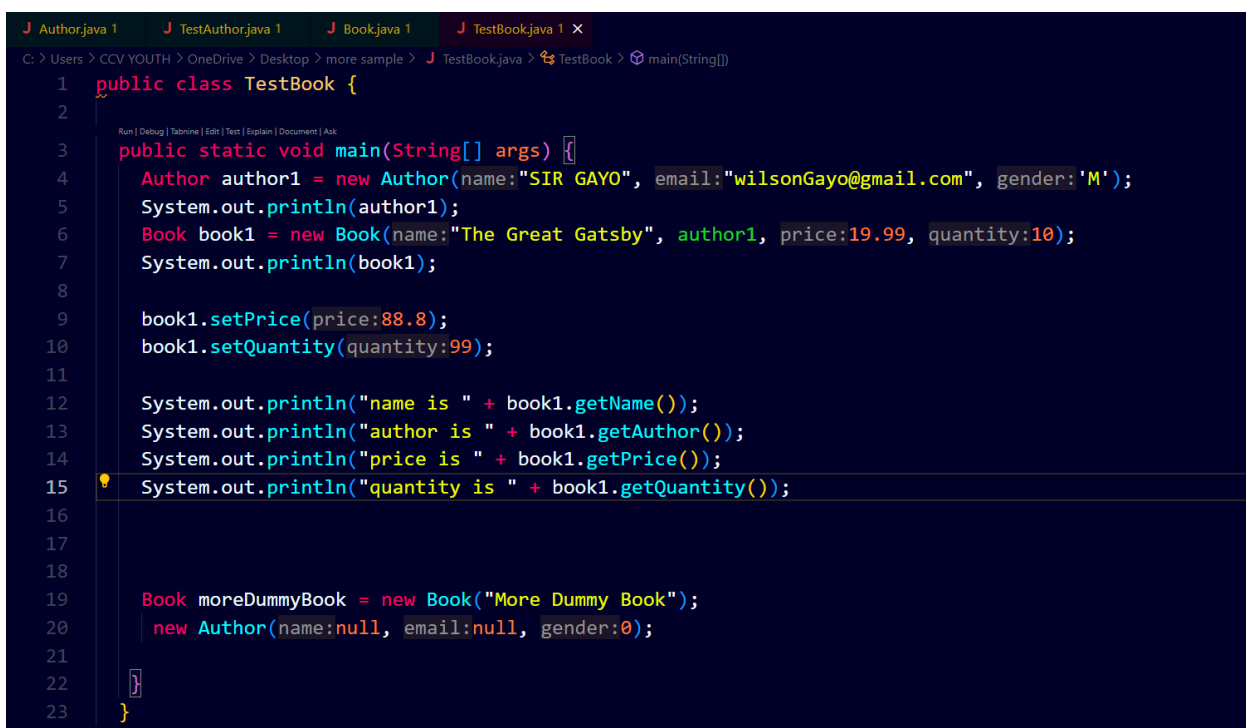
Constructors:

- **Book(String name, Author author, double price, int quantity)**: Initializes a **Book** object with the specified title, author, price, and quantity.
- **Book(String name)**: Initializes a **Book** object with only the book name. Default values are used for the author (**Unknown**), price (0.0), and quantity (0).

Methods:

- **getName()**: Returns the title of the book.
- **getAuthor()**: Returns the **Author** object representing the book's author.
- **getPrice()**: Returns the price of the book.
- **setPrice(double price)**: Allows you to update the price of the book.
- **getQuantity()**: Returns the quantity of the book available.
- **setQuantity(int quantity)**: Allows you to update the quantity of the book.
- **toString()**: Returns a string representation of the **Book** object in the format: 'book name' by author.

Documentation for **TestBook** class:



```
1 public class TestBook {
2
3     public static void main(String[] args) {
4         Author author1 = new Author(name:"SIR GAYO", email:"wilsonGayo@gmail.com", gender:'M');
5         System.out.println(author1);
6         Book book1 = new Book(name:"The Great Gatsby", author1, price:19.99, quantity:10);
7         System.out.println(book1);
8
9         book1.setPrice(price:88.8);
10        book1.setQuantity(quantity:99);
11
12        System.out.println("name is " + book1.getName());
13        System.out.println("author is " + book1.getAuthor());
14        System.out.println("price is " + book1.getPrice());
15        System.out.println("quantity is " + book1.getQuantity());
16
17
18
19        Book moreDummyBook = new Book("More Dummy Book");
20        new Author(name:null, email:null, gender:0);
21
22    }
23 }
```

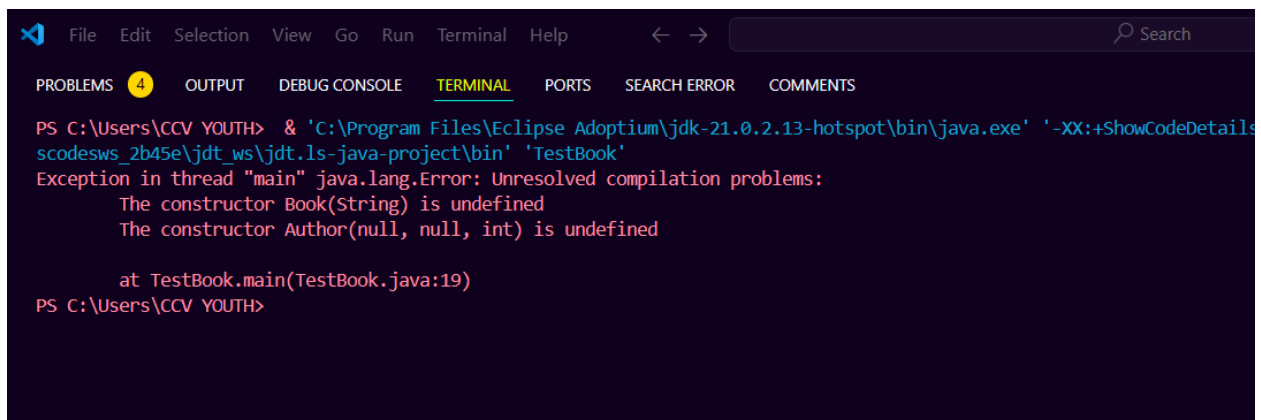
Purpose:

- The **TestBook** class is a test class used to create and test the functionality of **Book** and **Author** objects.

main Method:

- Creates an **Author** object (**author1**) with name "SIR GAYO", email "wilsonGayo@gmail.com", and gender 'M'.
- Creates a **Book** object (**book1**) with title "The Great Gatsby", associated with **author1**, price 19.99, and quantity 10.
- Modifies the **Book** object (**book1**) by changing the price to 88.8 and quantity to 99.
- Prints the updated details of the book, including the name, author, price, and quantity.
- Creates another **Book** object (**moreDummyBook**) using the overloaded constructor with only the book name, which uses default values for other fields.
- Creates an **Author** object with **null** values for name and email, and 'U' for gender, and prints it (this is not ideal but syntactically valid).

The Outputs

A screenshot of an IDE terminal window. The terminal shows a command prompt where a Java program is being executed. The command is: `PS C:\Users\CCV YOUTH> & 'C:\Program Files\Eclipse Adoptium\jdk-21.0.2.13-hotspot\bin\java.exe' '-XX:+ShowCodeDetails' 'scodesws_2b45e\jdt_ws\jdt.ls-java-project\bin' 'TestBook'`. The output shows an exception: `Exception in thread "main" java.lang.Error: Unresolved compilation problems:`, followed by two lines of error messages: `The constructor Book(String) is undefined` and `The constructor Author(null, null, int) is undefined`. The stack trace shows the error occurred at `TestBook.main(TestBook.java:19)`. The terminal window has a dark theme and a menu bar at the top with options like File, Edit, Selection, View, Go, Run, Terminal, and Help.

