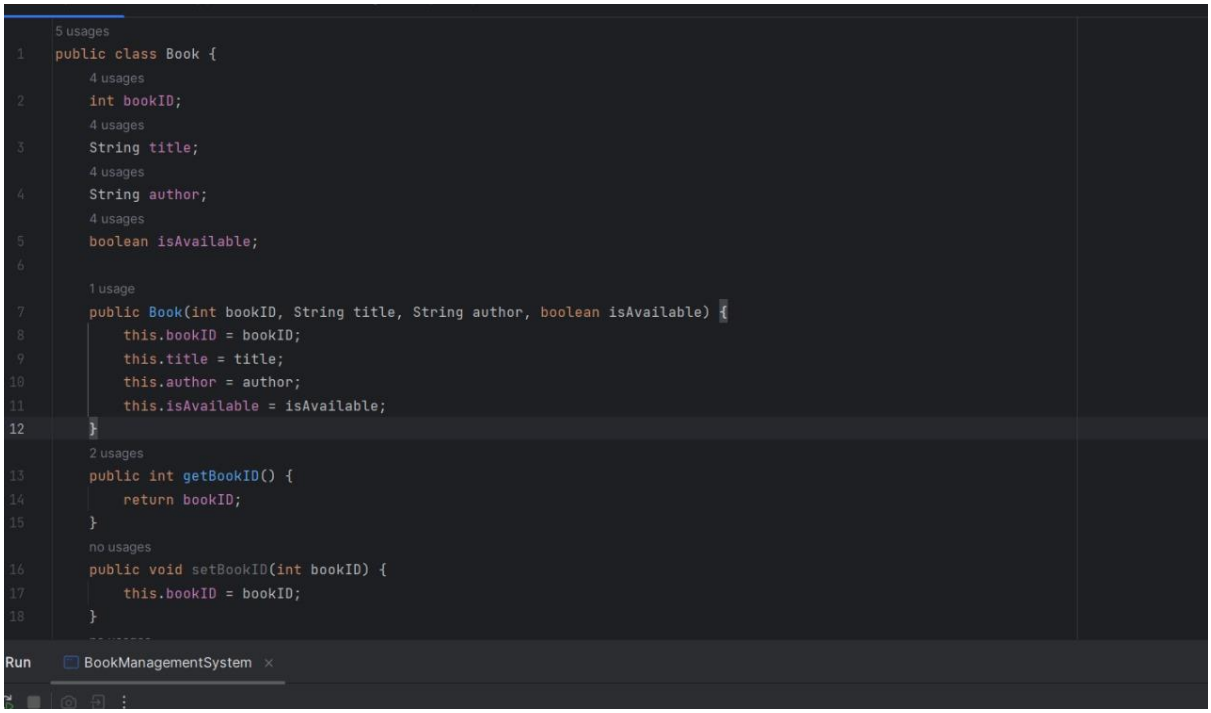
Task 3 Solution

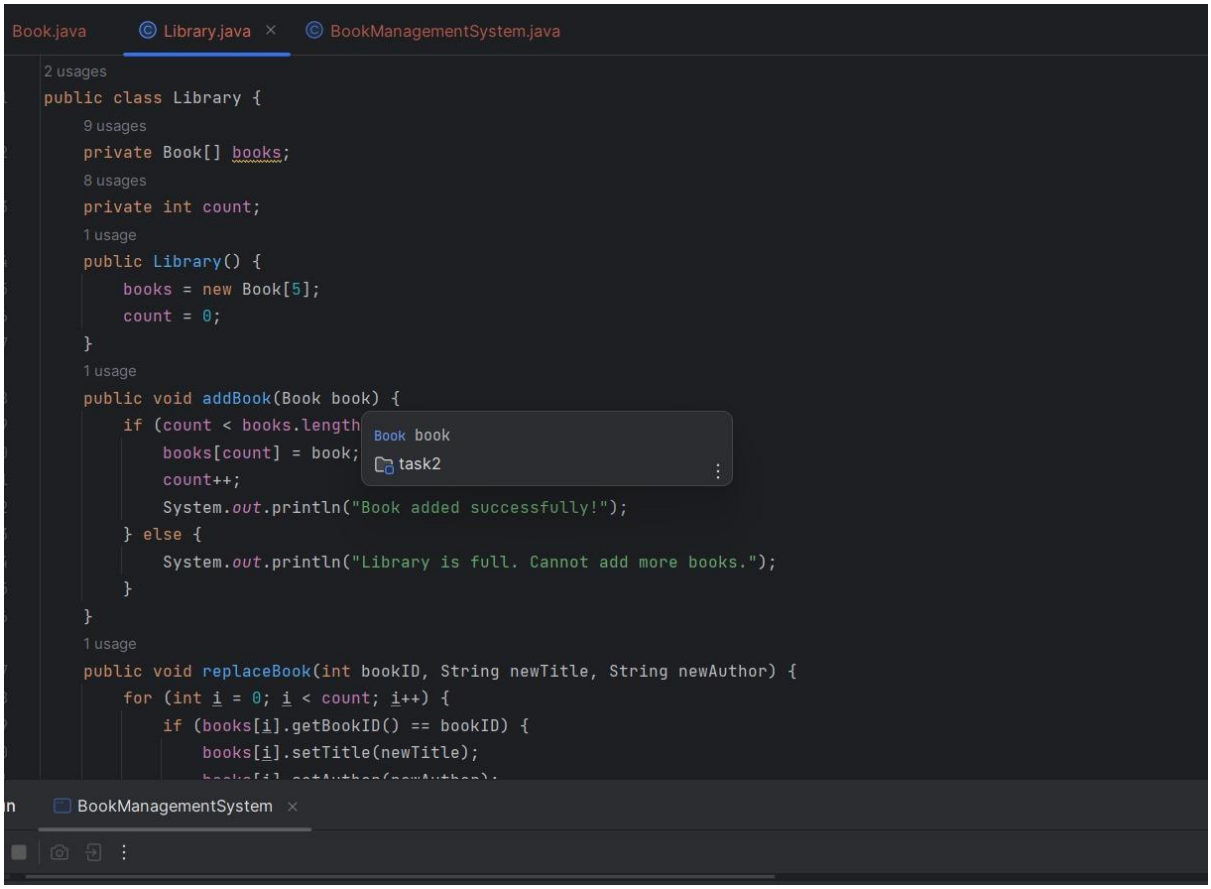
1.Design a Java program that uses OOP principles to model the Book. Create two classes: Book and Library. The Book class should have attributes such as bookID, title, author, and isAvailable. The Library class should include an array to store book objects.

2.Provide methods to add books, remove book search books (using id)and display books

Write a Java program that demonstrates the use of these classes and methods by allowing the user to interact with the library system.



```
Book.java × Library.java BookManagementSystem.java
19      public String getTitle() {
20          return title;
21      }
        1 usage
22      public void setTitle(String title) {
23          this.title = title;
24      }
        no usages
25      public String getAuthor() {
26          return author;
27      }
        1 usage
28      public void setAuthor(String author) {
29          this.author = author;
30      }
        no usages
31      public boolean isAvailable() {
32          return isAvailable;
33      }
        no usages
34      public void setAvailable(boolean isAvailable) {
35          this.isAvailable = isAvailable;
36      }
        2 usages
37      public void displayInfo() {
38          System.out.println("ID: " + bookID + ", Title: " + title + ", Author: " + author + ", Available: " + isAvailable);
Run  BookManagementSystem ×
```



```
Book.java  Library.java ×  BookManagementSystem.java
2 usages
public class Library {
    9 usages
    private Book[] books;
    8 usages
    private int count;
    1 usage
    public Library() {
        books = new Book[5];
        count = 0;
    }
    1 usage
    public void addBook(Book book) {
        if (count < books.length
            books[count] = book;
            count++;
            System.out.println("Book added successfully!");
        } else {
            System.out.println("Library is full. Cannot add more books.");
        }
    }
    1 usage
    public void replaceBook(int bookID, String newTitle, String newAuthor) {
        for (int i = 0; i < count; i++) {
            if (books[i].getBookID() == bookID) {
                books[i].setTitle(newTitle);
Book book
task2
n  BookManagementSystem ×
```

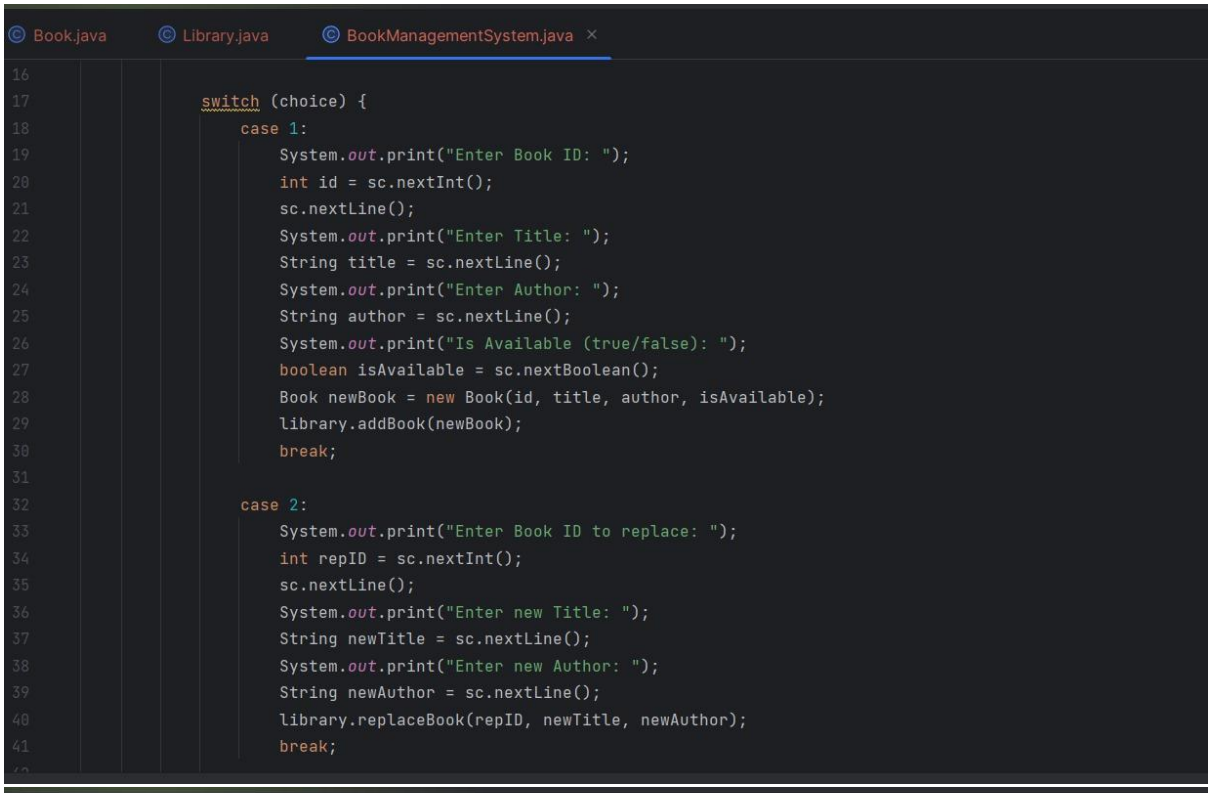
```
                System.out.println("Book updated successfully!");
                return;
            }
        }
        System.out.println("Book not found with ID: " + bookID);
    }
    1 usage
    public void displayBooks() {
        if (count == 0) {
            System.out.println("No books in the library.");
        } else {
            for (int i = 0; i < count; i++) {
                books[i].displayInfo();
            }
        }
    }
    1 usage
    public void searchBookByID(int bookID) {
        for (int i = 0; i < count; i++) {
            if (books[i].getBookID() == bookID) {
                books[i].displayInfo();
                return;
            }
        }
        System.out.println("Book not found with ID: " + bookID);
    }
```

un BookManagementSystem ×

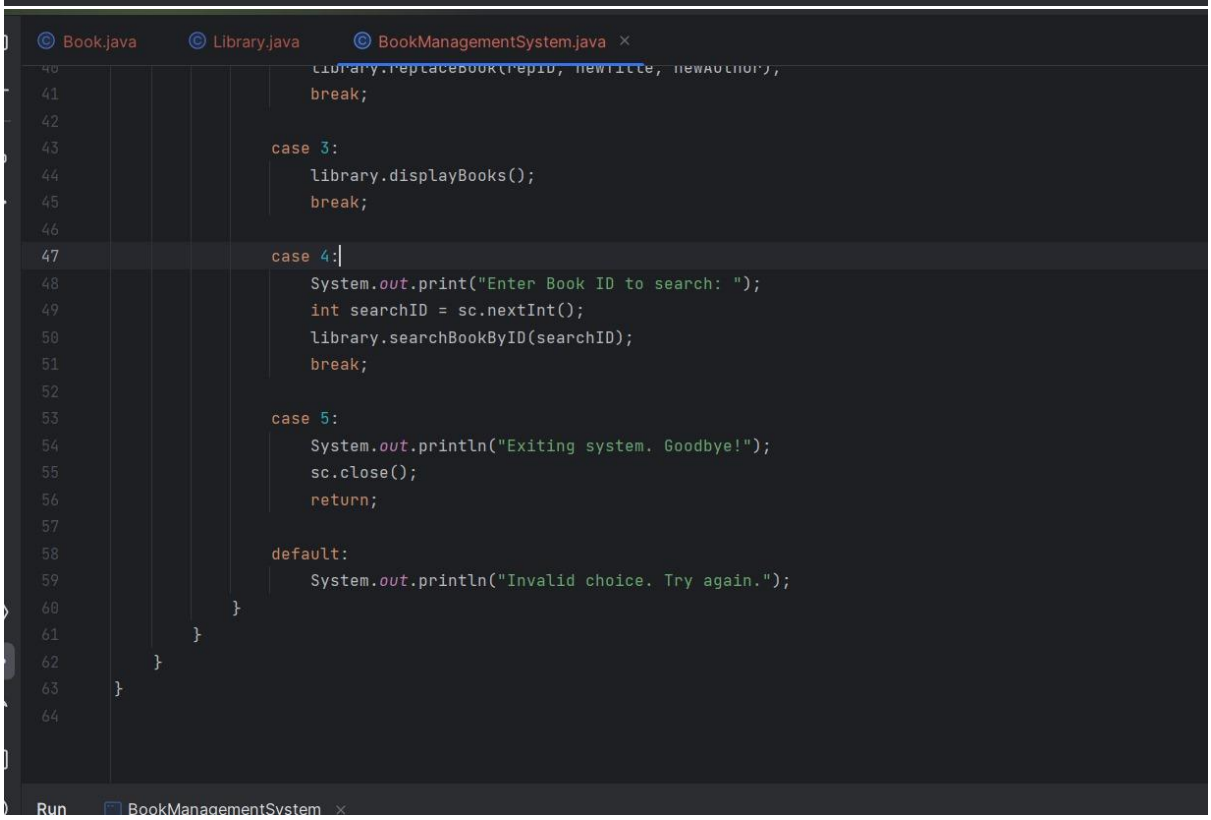
Book.java | Library.java | BookManagementSystem.java ×

```
import java.util.Scanner;

public class BookManagementSystem {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Library library = new Library();

        while (true) {
            System.out.println("1. Add Book");
            System.out.println("2. Replace Book by ID");
            System.out.println("3. Display All Books");
            System.out.println("4. Search Book by ID");
            System.out.println("5. Exit");
            System.out.print("Enter choice: ");
            int choice = sc.nextInt();

            switch (choice) {
                case 1:
                    System.out.print("Enter Book ID: ");
                    int id = sc.nextInt();
                    sc.nextLine();
                    System.out.print("Enter Title: ");
                    String title = sc.nextLine();
                    System.out.print("Enter Author: ");
                    String author = sc.nextLine();
                    System.out.print("Is Available (true/false): ");
```

n BookManagementSystem ×

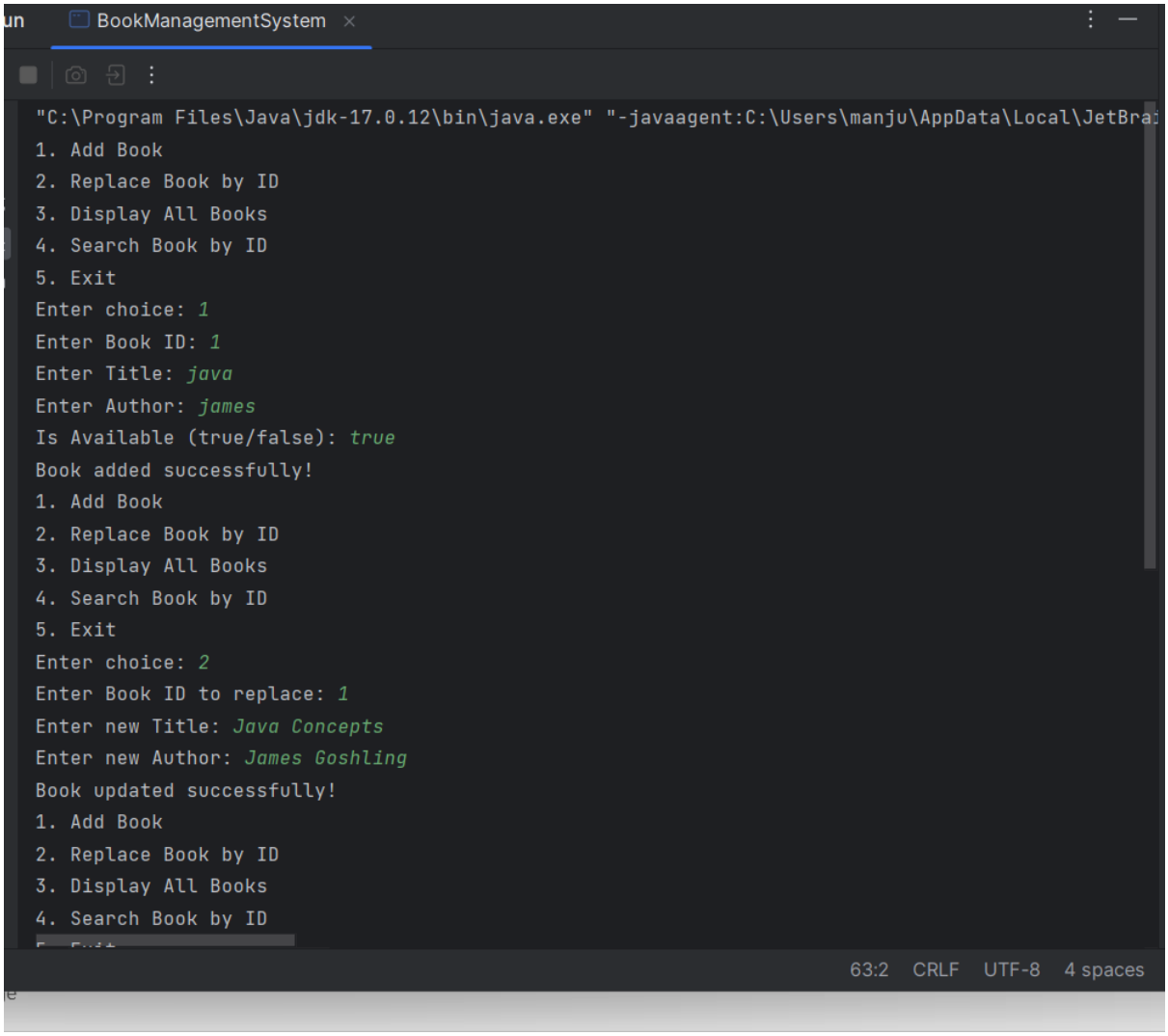


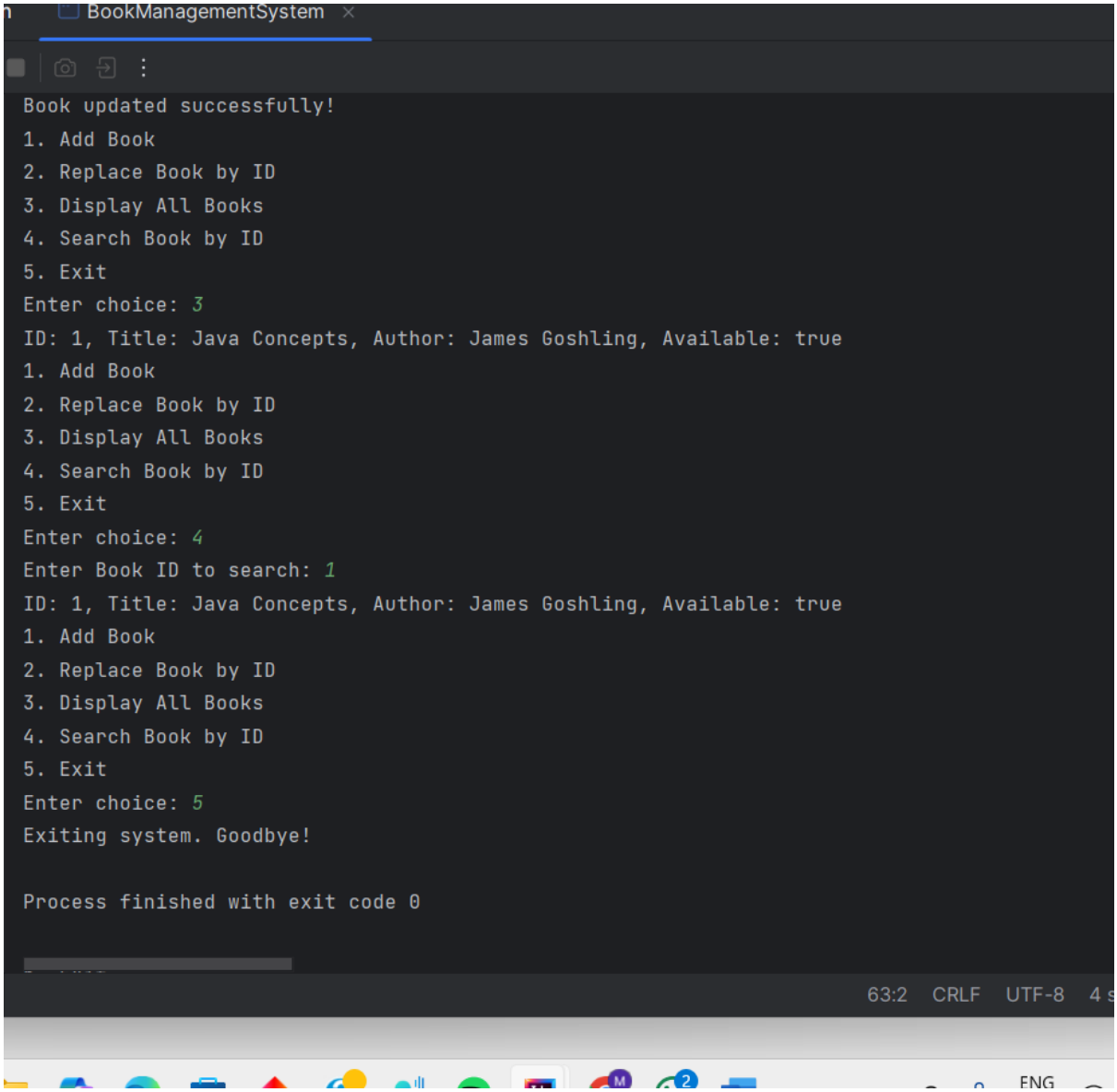
```
Book.java  Library.java  BookManagementSystem.java

16
17              switch (choice) {
18                  case 1:
19                      System.out.print("Enter Book ID: ");
20                      int id = sc.nextInt();
21                      sc.nextLine();
22                      System.out.print("Enter Title: ");
23                      String title = sc.nextLine();
24                      System.out.print("Enter Author: ");
25                      String author = sc.nextLine();
26                      System.out.print("Is Available (true/false): ");
27                      boolean isAvailable = sc.nextBoolean();
28                      Book newBook = new Book(id, title, author, isAvailable);
29                      library.addBook(newBook);
30                      break;
31
32                  case 2:
33                      System.out.print("Enter Book ID to replace: ");
34                      int repID = sc.nextInt();
35                      sc.nextLine();
36                      System.out.print("Enter new Title: ");
37                      String newTitle = sc.nextLine();
38                      System.out.print("Enter new Author: ");
39                      String newAuthor = sc.nextLine();
40                      library.replaceBook(repID, newTitle, newAuthor);
41                      break;
```



```
Book.java  Library.java  BookManagementSystem.java

41                      break;
42
43                  case 3:
44                      library.displayBooks();
45                      break;
46
47                  case 4:
48                      System.out.print("Enter Book ID to search: ");
49                      int searchID = sc.nextInt();
50                      library.searchBookByID(searchID);
51                      break;
52
53                  case 5:
54                      System.out.println("Exiting system. Goodbye!");
55                      sc.close();
56                      return;
57
58                  default:
59                      System.out.println("Invalid choice. Try again.");
60              }
61          }
62      }
63  }
64

Run  BookManagementSystem
```

Output:


un
BookManagementSystem
"C:\Program Files\Java\jdk-17.0.12\bin\java.exe" "-javaagent:C:\Users\manju\AppData\Local\JetBra
1. Add Book
2. Replace Book by ID
3. Display All Books
4. Search Book by ID
5. Exit
Enter choice: 1
Enter Book ID: 1
Enter Title: java
Enter Author: james
Is Available (true/false): true
Book added successfully!
1. Add Book
2. Replace Book by ID
3. Display All Books
4. Search Book by ID
5. Exit
Enter choice: 2
Enter Book ID to replace: 1
Enter new Title: Java Concepts
Enter new Author: James Goshling
Book updated successfully!
1. Add Book
2. Replace Book by ID
3. Display All Books
4. Search Book by ID
63:2
CRLF
UTF-8
4 spaces



2.Create Interface Taxable with members sales Tax=7% and income Tax=10.5%. create abstract method calc Tax().

a. Create class Employee(empId,name,salary) and implement Taxable to calculate income Tax on yearly salary.

b. Create class Product(pid.price,quantity) and implement Taxable to calculate sales Tax on unit price of product.

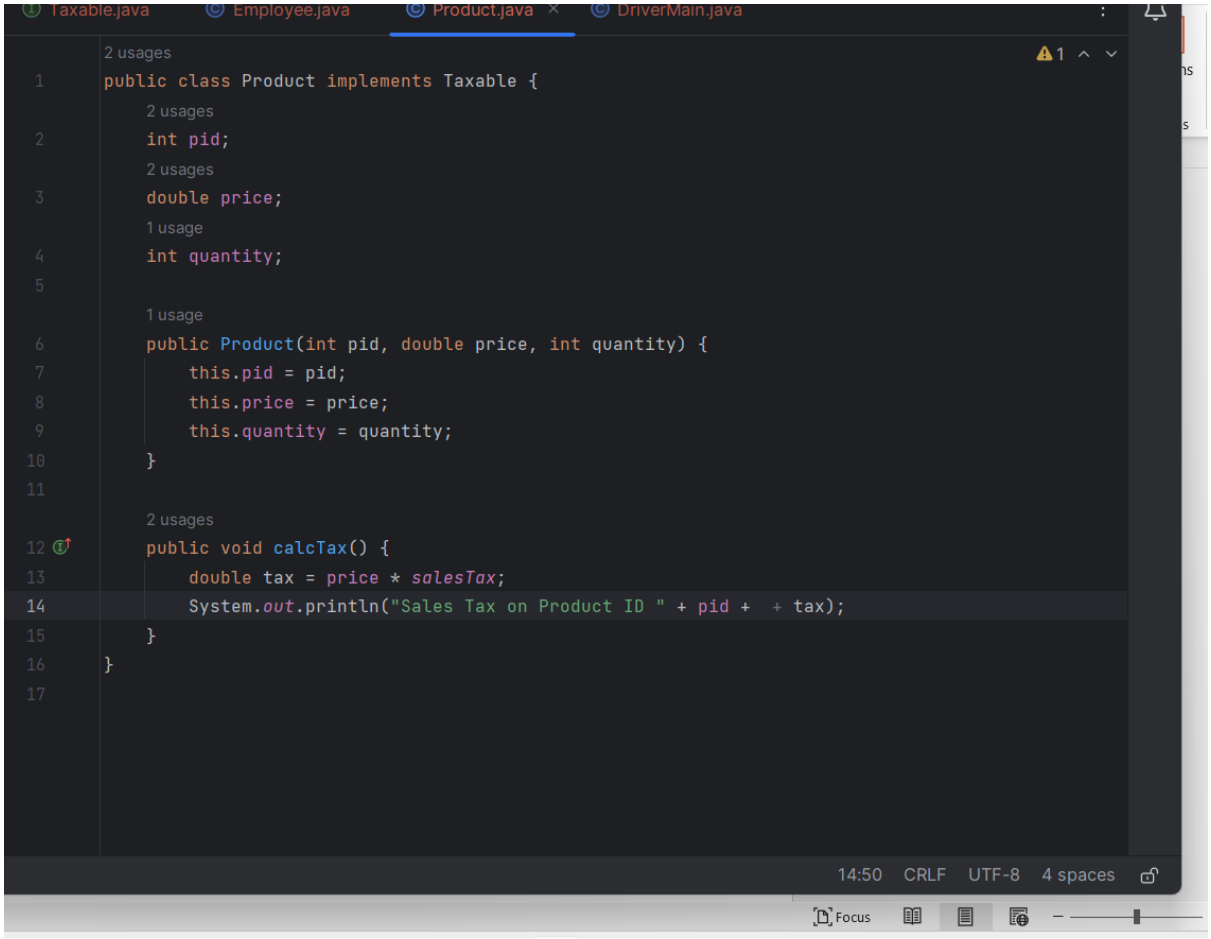
c. Create class for main method(Say DriverMain), accept employee information and a product information from user and print income tax and sales tax respectively

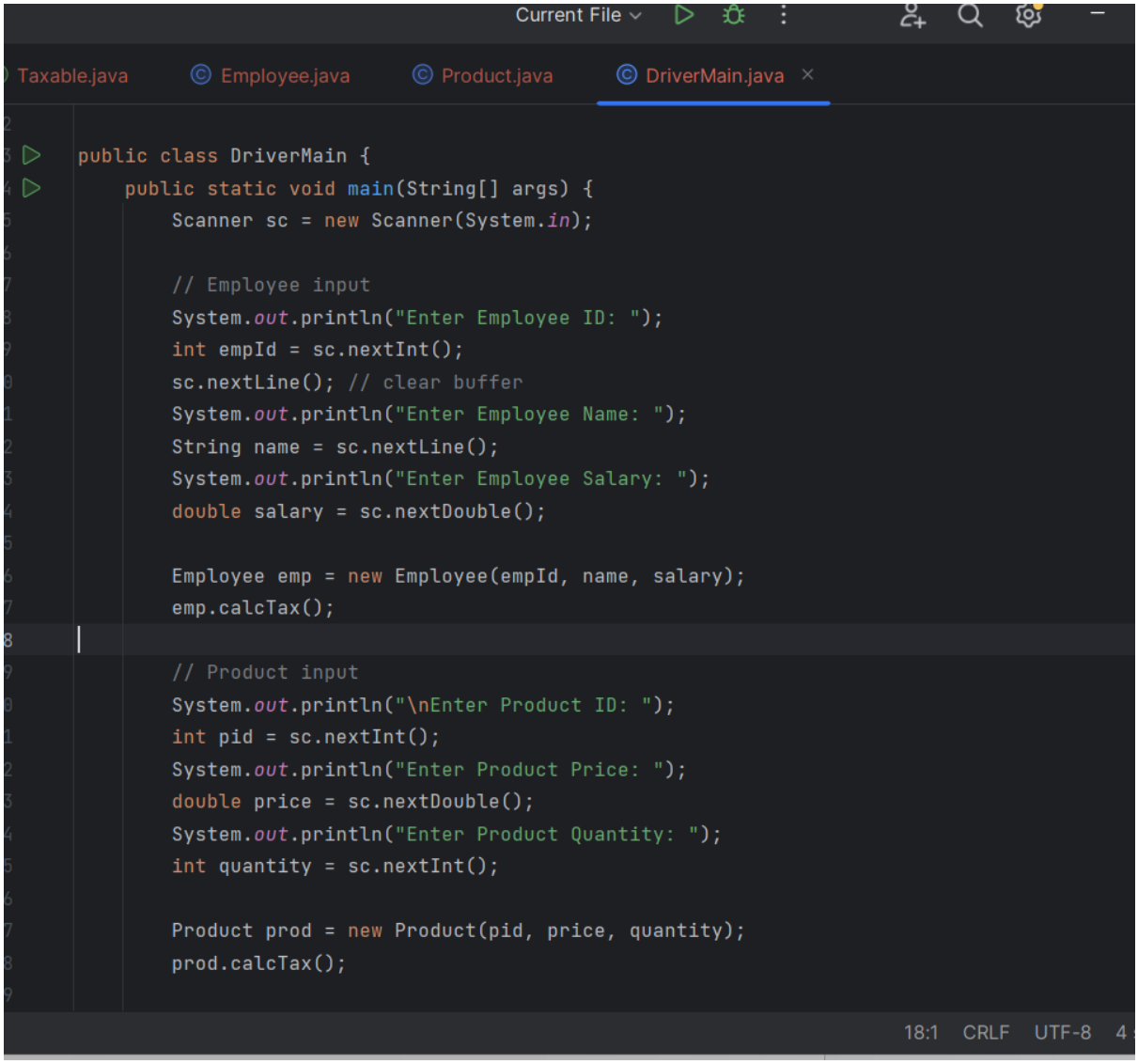
Taxable.java × Employee.java Product.java DriverMain.java

```java
2 usages  2 implementations
public interface Taxable {
    1 usage
    double salesTax = 0.07;     // 7%
    1 usage
    double incomeTax = 0.105;   // 10.5%

    2 usages  2 implementations
    void calcTax(); // abstract method
}
```

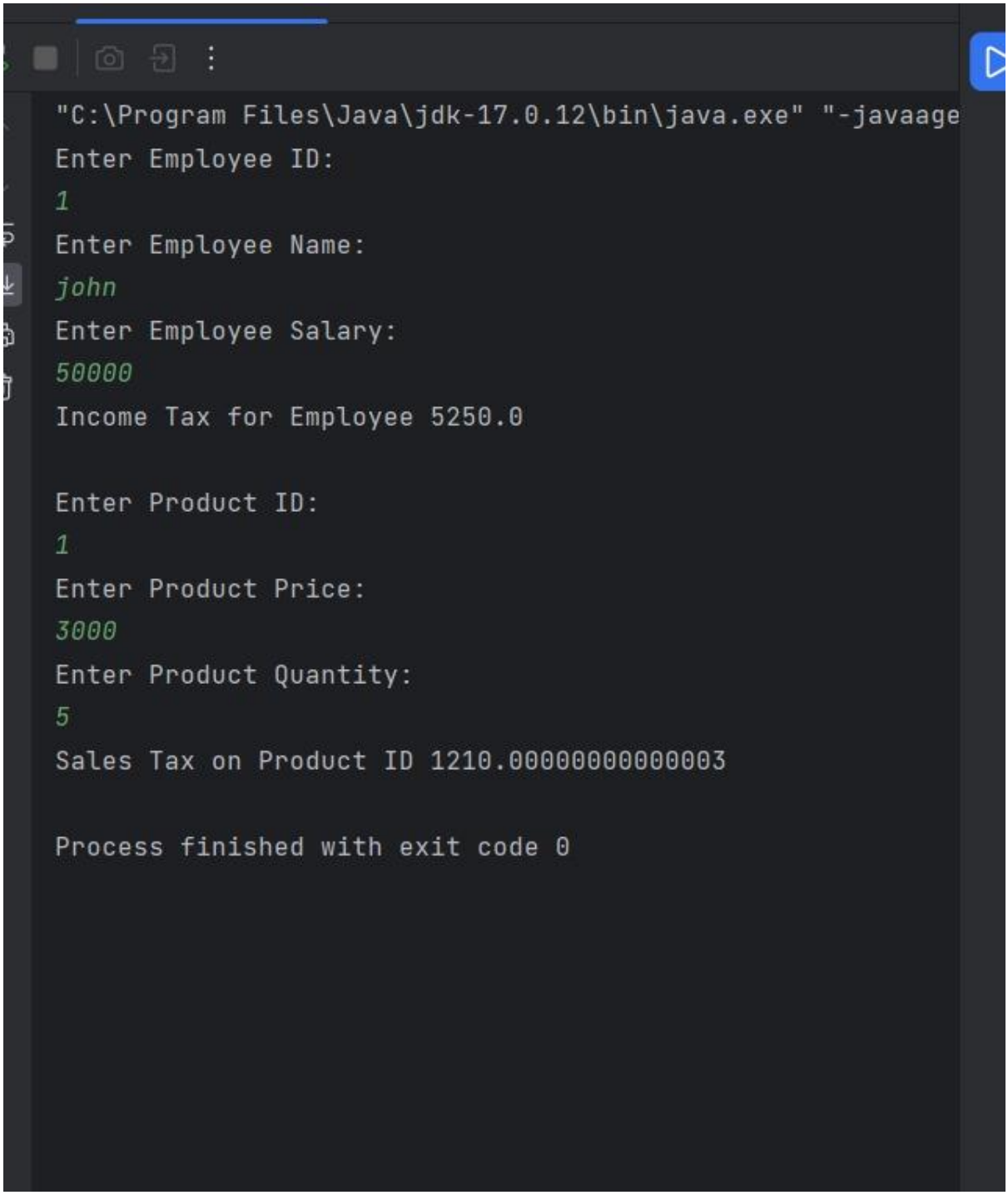
Taxable.java Employee.java × Product.java DriverMain.java

```java
2 usages
public class Employee implements Taxable {
    1 usage
    int empId;
    1 usage
    String name;
    2 usages
    double salary;

    1 usage
    public Employee(int empId, String name, double salary) {
        this.empId = empId;
        this.name = name;
        this.salary = salary;
    }

    2 usages
    public void calcTax() {
        double tax = salary * incomeTax;
        System.out.println("Income Tax for Employee " + tax);
    }
}
```


Taxable.java
Employee.java
Product.java
DriverMain.java
2 usages
1
public class Product implements Taxable {
2 usages
int pid;
2 usages
double price;
1 usage
int quantity;
1 usage
public Product(int pid, double price, int quantity) {
this.pid = pid;
this.price = price;
this.quantity = quantity;
}
2 usages
public void calcTax() {
double tax = price * salesTax;
System.out.println("Sales Tax on Product ID " + pid + + tax);
}
}
14:50
CRLF
UTF-8
4 spaces
Focus


Current File
Taxable.java
Employee.java
Product.java
DriverMain.java
public class DriverMain {
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
// Employee input
System.out.println("Enter Employee ID: ");
int empId = sc.nextInt();
sc.nextLine(); // clear buffer
System.out.println("Enter Employee Name: ");
String name = sc.nextLine();
System.out.println("Enter Employee Salary: ");
double salary = sc.nextDouble();
Employee emp = new Employee(empId, name, salary);
emp.calcTax();
// Product input
System.out.println("\nEnter Product ID: ");
int pid = sc.nextInt();
System.out.println("Enter Product Price: ");
double price = sc.nextDouble();
System.out.println("Enter Product Quantity: ");
int quantity = sc.nextInt();
Product prod = new Product(pid, price, quantity);
prod.calcTax();
18:1
CRLF
UTF-8

Output:


"C:\Program Files\Java\jdk-17.0.12\bin\java.exe" "-javaage
Enter Employee ID:
1
Enter Employee Name:
john
Enter Employee Salary:
50000
Income Tax for Employee 5250.0
Enter Product ID:
1
Enter Product Price:
3000
Enter Product Quantity:
5
Sales Tax on Product ID 1210.00000000000003
Process finished with exit code 0