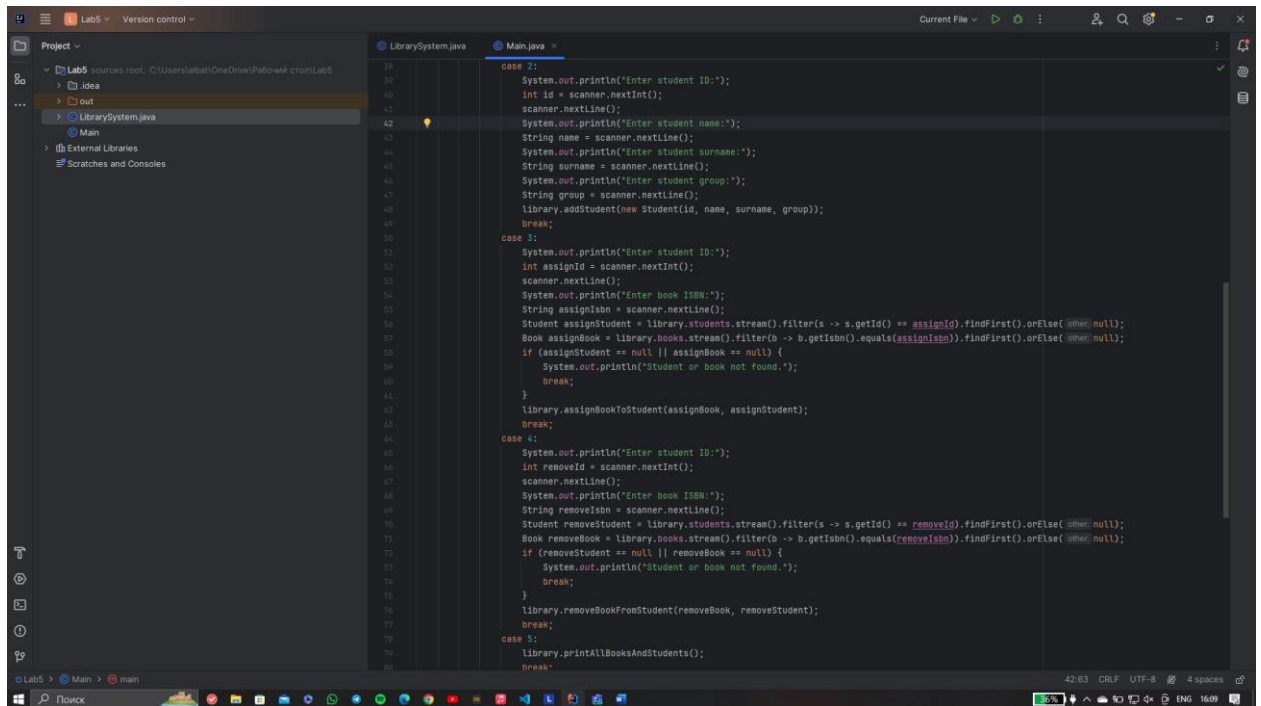
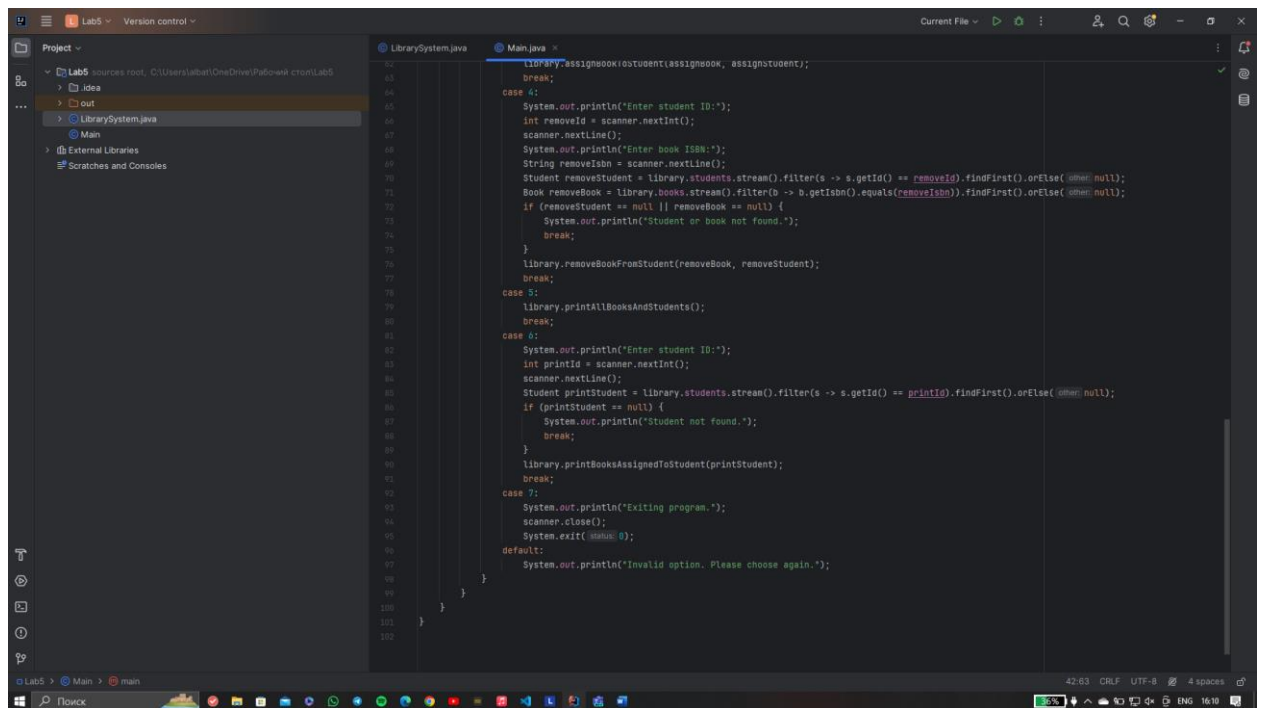


```
1 import java.util.Scanner;
2
3 public class Main {
4     public static void main(String[] args) {
5         Library library = new Library();
6         Scanner scanner = new Scanner(System.in);
7
8         System.out.println("Welcome to the Library System!");
9
10        while (true) {
11            System.out.println("\nChoose an option:");
12            System.out.println("1. Add Book");
13            System.out.println("2. Add Student");
14            System.out.println("3. Assign Book to Student");
15            System.out.println("4. Remove Book from Student");
16            System.out.println("5. Print All Books and Students");
17            System.out.println("6. Print Books Assigned to Student");
18            System.out.println("7. Exit");
19
20            int choice = scanner.nextInt();
21            scanner.nextLine();
22
23            switch (choice) {
24                case 1:
25                    System.out.println("Enter book title:");
26                    String title = scanner.nextLine();
27                    System.out.println("Enter author name:");
28                    String author = scanner.nextLine();
29                    System.out.println("Enter ISBN:");
30                    String isbn = scanner.nextLine();
31                    System.out.println("Enter publication year:");
32                    int year = scanner.nextInt();
33                    System.out.println("Enter quantity:");
34                    int quantity = scanner.nextInt();
35                    scanner.nextLine();
36                    library.addBook(new Book(title, author, isbn, year, quantity));
37                    break;
38                case 2:
39                    System.out.println("Enter student ID:");
40                    int id = scanner.nextInt();
41                    scanner.nextLine();
42                    System.out.println("Enter student name:");
43                    String name = scanner.nextLine();
44                    System.out.println("Enter student surname:");
45                    String surname = scanner.nextLine();
46                    System.out.println("Enter student group:");
47                    String group = scanner.nextLine();
48                    library.addStudent(new Student(id, name, surname, group));
49                    break;
50                case 3:
51                    System.out.println("Enter student ID:");
52                    int assignId = scanner.nextInt();
53                    scanner.nextLine();
54                    System.out.println("Enter book ISBN:");
55                    String assignIsbn = scanner.nextLine();
56                    Student assignStudent = library.students.stream().filter(s -> s.getId() == assignId).findFirst().orElse(null);
57                    Book assignBook = library.books.stream().filter(b -> b.getIsbn().equals(assignIsbn)).findFirst().orElse(null);
58                    if (assignStudent == null || assignBook == null) {
59                        System.out.println("Student or book not found.");
60                        break;
61                    }
62                    library.assignBookToStudent(assignBook, assignStudent);
63                    break;
64                case 4:
65                    System.out.println("Enter student ID:");
66                    int removeId = scanner.nextInt();
67                    scanner.nextLine();
68                    System.out.println("Enter book ISBN:");
69                    String removeIsbn = scanner.nextLine();
70                    Student removeStudent = library.students.stream().filter(s -> s.getId() == removeId).findFirst().orElse(null);
71                    Book removeBook = library.books.stream().filter(b -> b.getIsbn().equals(removeIsbn)).findFirst().orElse(null);
72                    if (removeStudent == null || removeBook == null) {
73                        System.out.println("Student or book not found.");
74                        break;
75                    }
76                    library.removeBookFromStudent(removeBook, removeStudent);
77                    break;
78                case 5:
79                    library.printAllBooksAndStudents();
80                    break;
81                case 7:
82                    System.out.println("Exiting Library System. Goodbye!");
83                    return;
84            }
85        }
86    }
87 }
```



```
38
39
40 int id = scanner.nextInt();
41 scanner.nextLine();
42 System.out.println("Enter student name:");
43 String name = scanner.nextLine();
44 System.out.println("Enter student surname:");
45 String surname = scanner.nextLine();
46 System.out.println("Enter student group:");
47 String group = scanner.nextLine();
48 library.addStudent(new Student(id, name, surname, group));
49 break;
50
51 case 3:
52     System.out.println("Enter student ID:");
53     int assignId = scanner.nextInt();
54     scanner.nextLine();
55     System.out.println("Enter book ISBN:");
56     String assignIsbn = scanner.nextLine();
57     Student assignStudent = library.students.stream().filter(s -> s.getId() == assignId).findFirst().orElse(null);
58     Book assignBook = library.books.stream().filter(b -> b.getIsbn().equals(assignIsbn)).findFirst().orElse(null);
59     if (assignStudent == null || assignBook == null) {
60         System.out.println("Student or book not found.");
61         break;
62     }
63     library.assignBookToStudent(assignBook, assignStudent);
64     break;
65
66 case 4:
67     System.out.println("Enter student ID:");
68     int removeId = scanner.nextInt();
69     scanner.nextLine();
70     System.out.println("Enter book ISBN:");
71     String removeIsbn = scanner.nextLine();
72     Student removeStudent = library.students.stream().filter(s -> s.getId() == removeId).findFirst().orElse(null);
73     Book removeBook = library.books.stream().filter(b -> b.getIsbn().equals(removeIsbn)).findFirst().orElse(null);
74     if (removeStudent == null || removeBook == null) {
75         System.out.println("Student or book not found.");
76         break;
77     }
78     library.removeBookFromStudent(removeBook, removeStudent);
79     break;
80
81 case 5:
82     library.printAllBooksAndStudents();
83     break;
84
85 case 7:
86     System.out.println("Exiting Library System. Goodbye!");
87     return;
88 }
```



1) The 'Main' class in this code snippet is the entry point for the library management system. It uses a 'Scanner' object to read user input from the console and interacts with the 'Library' class to perform various operations.

1. Initialization:

- Creates an instance of the 'Library' class to manage the library operations.
- Creates a 'Scanner' object to read user input.

2. Menu Display and User Input:

- Displays a menu of options for the user to choose from.
- Reads the user's choice using `scanner.nextInt()` and consumes the newline character with `scanner.nextLine()`.

3. *Switch Statement:

- Handles different user choices by executing the corresponding case.
- For each case:
 - Prompts the user for necessary information (e.g., book title, author, ISBN, student ID, etc.).
 - Reads the input using `scanner.nextLine()` or `scanner.nextInt()` based on the data type.
 - Calls the corresponding method of the `'Library'` class to perform the operation (e.g., `addBook`, `addStudent` and so on)

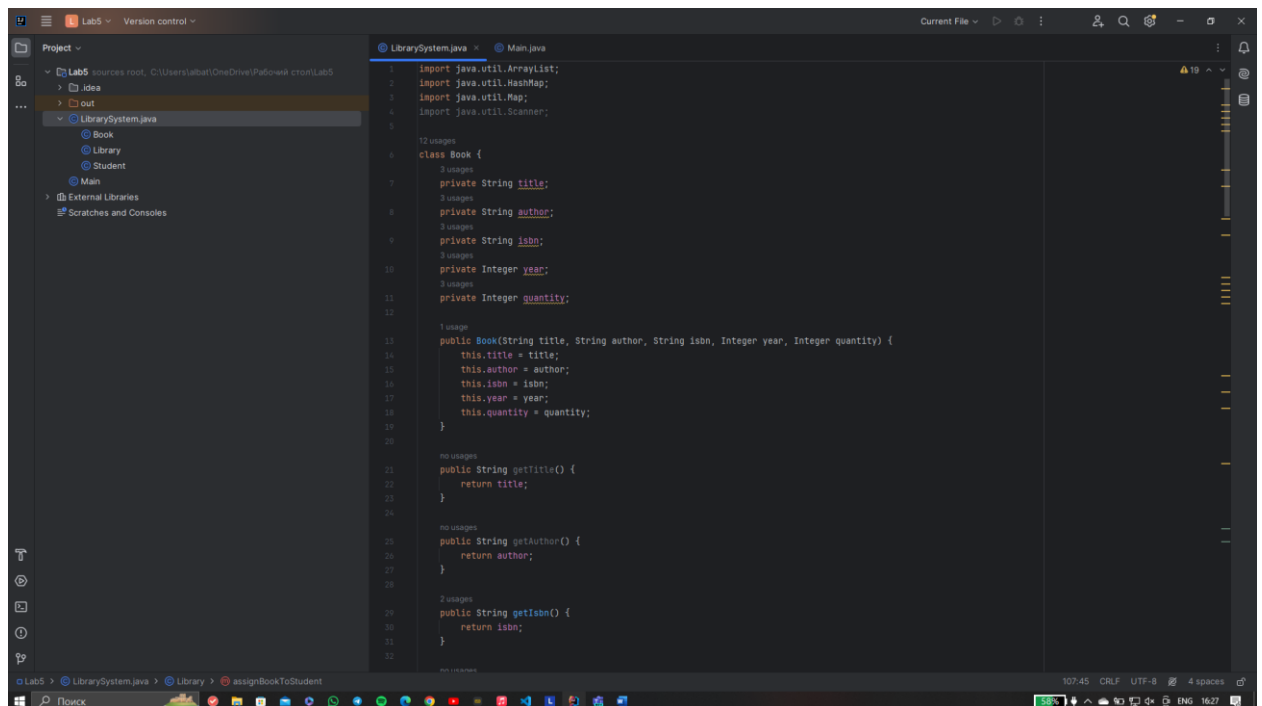
4. Loop:

- Uses a `while (true)` loop to keep the program running until the user chooses to exit (option `q`).
- Ensures that the user can perform multiple operations without restarting the program.

5. Resource Management:

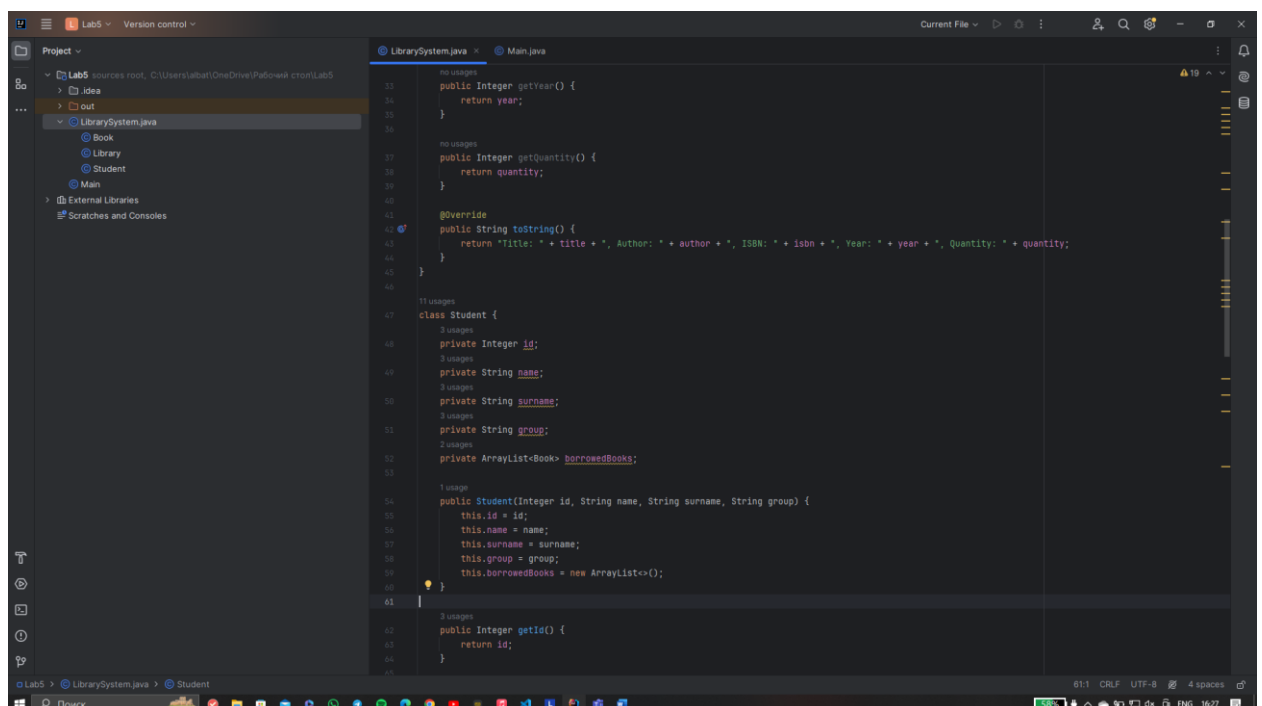
- Closes the 'Scanner' object ('scanner.close()') and exits the program ('System.exit(0)') when the user chooses to exit.

'Main' class provides a user-friendly interface for managing a library, allowing users to add books and students, assign books to students, remove books from students, and view information about books and students.



The screenshot shows the IntelliJ IDEA IDE with the 'Lab5' project open. The 'Project' view on the left shows the file structure: 'Lab5' > 'src' > 'LibrarySystem.java'. The 'LibrarySystem.java' file is open, showing the 'Book' class. The code includes imports for 'java.util.ArrayList', 'java.util.HashMap', 'java.util.Map', and 'java.util.Scanner'. The 'Book' class has private attributes: 'title', 'author', 'isbn', 'year', and 'quantity'. It has a constructor 'Book(String title, String author, String isbn, Integer year, Integer quantity)' and three getter methods: 'getTitle()', 'getAuthor()', and 'getIsbn()'. The status bar at the bottom indicates '107:45 CRLF UTF-8 4 spaces'.

```
1 import java.util.ArrayList;
2 import java.util.HashMap;
3 import java.util.Map;
4 import java.util.Scanner;
5
6 class Book {
7     private String title;
8     private String author;
9     private String isbn;
10    private Integer year;
11    private Integer quantity;
12
13    public Book(String title, String author, String isbn, Integer year, Integer quantity) {
14        this.title = title;
15        this.author = author;
16        this.isbn = isbn;
17        this.year = year;
18        this.quantity = quantity;
19    }
20
21    public String getTitle() {
22        return title;
23    }
24
25    public String getAuthor() {
26        return author;
27    }
28
29    public String getIsbn() {
30        return isbn;
31    }
32}
```

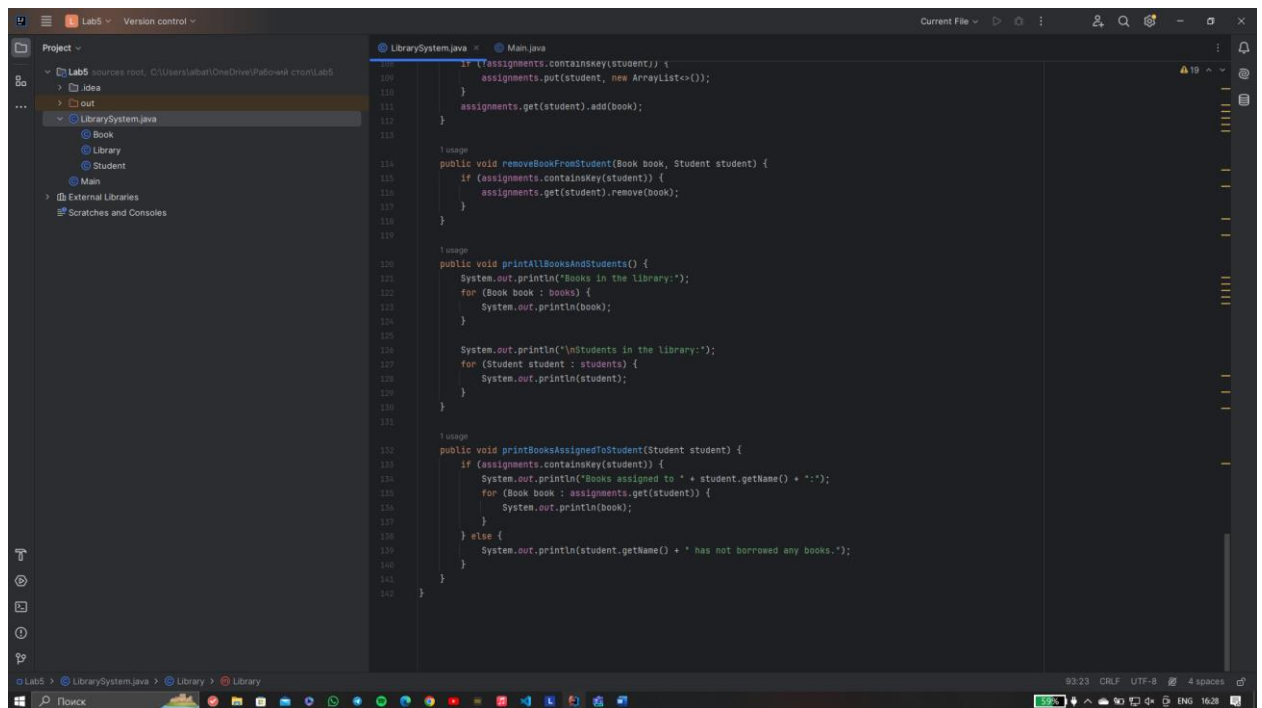


The screenshot shows the IntelliJ IDEA IDE with the 'Lab5' project open. The 'Project' view on the left shows the file structure: 'Lab5' > 'src' > 'LibrarySystem.java'. The 'LibrarySystem.java' file is open, showing the 'Student' class. The code includes imports for 'java.util.ArrayList', 'java.util.HashMap', 'java.util.Map', and 'java.util.Scanner'. The 'Student' class has private attributes: 'id', 'name', 'surname', 'group', and 'borrowedBooks'. It has a constructor 'Student(Integer id, String name, String surname, String group)' and two getter methods: 'getId()' and 'getQuantity()'. The status bar at the bottom indicates '81:1 CRLF UTF-8 4 spaces'.

```
33 public Integer getYear() {
34     return year;
35 }
36
37 public Integer getQuantity() {
38     return quantity;
39 }
40
41 @Override
42 public String toString() {
43     return "Title: " + title + ", Author: " + author + ", ISBN: " + isbn + ", Year: " + year + ", Quantity: " + quantity;
44 }
45
46 class Student {
47     private Integer id;
48     private String name;
49     private String surname;
50     private String group;
51     private ArrayList<Book> borrowedBooks;
52
53     public Student(Integer id, String name, String surname, String group) {
54         this.id = id;
55         this.name = name;
56         this.surname = surname;
57         this.group = group;
58         this.borrowedBooks = new ArrayList<>();
59     }
60
61     public Integer getId() {
62         return id;
63     }
64 }
```

```
62 public Integer getId() {
63     return id;
64 }
65
66 2 usages
67 public String getName() {
68     return name;
69 }
70
71 no usages
72 public String getSurname() {
73     return surname;
74 }
75
76 no usages
77 public String getGroup() {
78     return group;
79 }
80
81 no usages
82 public ArrayList<Book> getBorrowedBooks() {
83     return borrowedBooks;
84 }
85
86 @Override
87 public String toString() {
88     return "ID: " + id + ", Name: " + name + ", Surname: " + surname + ", Group: " + group;
89 }
90
91 2 usages
92 class Library {
93     5 usages
94     ArrayList<Book> books;
95     6 usages
96     ArrayList<Student> students;
97     8 usages
98     private Map<Student, ArrayList<Book>> assignments;
99
100 1 usage
101 public Library() {
102     this.books = new ArrayList<>();
103     this.students = new ArrayList<>();
104 }
```

```
105 1 usage
106 public Library() {
107     this.books = new ArrayList<>();
108     this.students = new ArrayList<>();
109     this.assignments = new HashMap<>();
110 }
111
112 1 usage
113 public void addBook(Book book) {
114     books.add(book);
115 }
116
117 1 usage
118 public void addStudent(Student student) {
119     students.add(student);
120 }
121
122 1 usage
123 public void assignBookToStudent(Book book, Student student) {
124     if (!assignments.containsKey(student)) {
125         assignments.put(student, new ArrayList<>());
126     }
127     assignments.get(student).add(book);
128 }
129
130 1 usage
131 public void removeBookFromStudent(Book book, Student student) {
132     if (assignments.containsKey(student)) {
133         assignments.get(student).remove(book);
134     }
135 }
136
137 1 usage
138 public void printAllBooksAndStudents() {
139     System.out.println("Books in the library:");
140     for (Book book : books) {
141         System.out.println(book);
142     }
143
144     System.out.println("\nStudents in the library:");
145     for (Student student : students) {
146         System.out.println(student);
147     }
148 }
```



This part of code defines three classes: `Book`, `Student`, and `Library`, which together form a basic library management system:

1. *Book Class:

- Represents a book with properties such as title, author, ISBN, publication year, and quantity.
- Provides a constructor to initialize these properties.
- Includes getter methods to access the properties.
- Overrides the `toString` method to return a formatted string representation of the book.

2. Student Class:

- Represents a student with properties such as ID, name, surname, group, and a list of borrowed books.
- Provides a constructor to initialize these properties, including an empty list for borrowed books.
- Includes getter methods to access the properties.
- Overrides the `toString` method to return a formatted string representation of the student.

3. Library Class:

- Manages the books, students, and their assignments.
- Contains ArrayLists to store books and students, and a HashMap to store assignments (books assigned to students).
- Provides methods to add books and students to the library, assign books to students, remove books from students, and print information about books and students.

- The `printAllBooksAndStudents` method iterates over the books and students to print their information.
- The `printBooksAssignedToStudent` method prints the books assigned to a specific student, if any.

So, these classes work together to create a simple library management system that allows adding books and students, assigning books to students, removing books from students, and viewing information about books and students.

Here is output:

```

C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2023.3.2\lib\idea_rt.jar=S4473:C:\Program Files\JetBrains\IntelliJ IDEA 2023.3.2\bin" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8
Welcome to the Library System!

Choose an option:
1. Add Book
2. Add Student
3. Assign Book to Student
4. Remove Book from Student
5. Print All Books and Students
6. Print Books Assigned to Student
7. Exit
1
Enter book title:
Ikigai
Enter author name:
Ken Mogi
Enter ISBN:
1234
Enter publication year:
2018
Enter quantity:
8

Choose an option:
1. Add Book
2. Add Student
3. Assign Book to Student
4. Remove Book from Student
5. Print All Books and Students
6. Print Books Assigned to Student
7. Exit
2
Enter student ID:
24592
Enter student name:
Albatno
Enter student surname:
Subit
Enter student group:
IT1-2202
  
```

```

IT1-2202

Choose an option:
1. Add Book
2. Add Student
3. Assign Book to Student
4. Remove Book from Student
5. Print All Books and Students
6. Print Books Assigned to Student
7. Exit
3
Enter student ID:
24592
Enter book ISBN:
1234

Choose an option:
1. Add Book
2. Add Student
3. Assign Book to Student
4. Remove Book from Student
5. Print All Books and Students
6. Print Books Assigned to Student
7. Exit
4
Enter student ID:
24592
Books assigned to Albatno:
Title: Ikigai, Author: Ken Mogi, ISBN: 1234, Year: 2018, Quantity: 8

Choose an option:
1. Add Book
2. Add Student
3. Assign Book to Student
4. Remove Book from Student
5. Print All Books and Students
6. Print Books Assigned to Student
7. Exit
4
Enter student ID:
24592
  
```

```
Lab5 - Version control
Project
Run Main
LibrarySystem.java Main.java
Enter student ID:
34592
Enter book ISBN:
1234
Choose an option:
1. Add Book
2. Add Student
3. Assign Book to Student
4. Remove Book from Student
5. Print All Books and Students
6. Print Books Assigned to Student
7. Exit
0
Enter student ID:
34592
Books assigned to Albatnu:
Choose an option:
1. Add Book
2. Add Student
3. Assign Book to Student
4. Remove Book from Student
5. Print All Books and Students
6. Print Books Assigned to Student
7. Exit
5
Books in the library:
Title: Ikigai, Author: Ken Mogi, ISBN: 1234, Year: 2018, Quantity: 8
Students in the library:
ID: 34592, Name: Albatnu, Surname: Sabit, Group: ITI-2202
Choose an option:
1. Add Book
2. Add Student
3. Assign Book to Student
4. Remove Book from Student
5. Print All Books and Students
5:Print All Books and Students
Lab5 > Main > main
12:47 CRLF UTF-8 4 spaces
```