

INPUT

```
import java.util.ArrayList;
```

```
import java.util.Scanner;
```

14 usages

```
class Books {
```

2 usages

```
static ArrayList<String> Author = new ArrayList<>();
```

6 usages

```
static ArrayList<String> Title = new ArrayList<>();
```

2 usages

```
static ArrayList<Integer> Price = new ArrayList<>();
```

```
}
```

7 usages

```
class LibraryMember {
```

3 usages

```
static ArrayList<String> Username = new ArrayList<>();
```

4 usages

```
static ArrayList<String> Password = new ArrayList<>();
```

no usages

```
LibraryMember(ArrayList<String> Username, ArrayList<String> Password) {
```

```
    this.Username = Username;
```

```
    this.Password = Password;
```

```
}
```

```
}
```

4 usages

```
class LibraryStaff {
```

3 usages

```
static ArrayList<String> staffName = new ArrayList<>();
```

2 usages

```
static ArrayList<String> Role = new ArrayList<>();
```

no usages

```
LibraryStaff(ArrayList<String> staffName, ArrayList<String> Role) {
```

```
    this.staffName = staffName;
```

```
    this.Role = Role;
```

```
}
```

```
}
```

1 usage

```
class AddBook implements Runnable {
```

2 usages

```
String staffName;
```

3 usages

```
String bookTitle;
```

2 usages

```
String author;
```

2 usages

```
int price;
```

1 usage

```
public AddBook(String staffName, String bookTitle, String author, int price) {  
    this.staffName = staffName;  
    this.bookTitle = bookTitle;  
    this.author = author;  
    this.price = price;  
}  
  
public void run() {  
    synchronized (Books.class) {  
        Books.Title.add(bookTitle);  
        Books.Author.add(author);  
        Books.Price.add(price);  
        System.out.println("Book '" + bookTitle + "' added to the library by " + staffName);  
    }  
}  
}
```

1 usage

```
class SearchBook implements Runnable {  
    1 usage  
    String staffName;  
    2 usages  
    String bookTitle;
```

! usage

```
public SearchBook(String staffName, String bookTitle) {  
    this.staffName = staffName;  
    this.bookTitle = bookTitle;  
}  
  
public void run() {  
    synchronized (Books.class) {  
        int index = Books.Title.indexOf(bookTitle);  
        if (index != -1) {  
            System.out.println("Book found:");  
            System.out.println("Title: " + Books.Title.get(index));  
            System.out.println("Author: " + Books.Author.get(index));  
            System.out.println("Price: " + Books.Price.get(index));  
        } else {  
            System.out.println("Book not found.");  
        }  
    }  
}  
}
```

1 usage

```
class BorrowBook implements Runnable {
```

2 usages

```
String memberName;
```

5 usages

```
String bookTitle;
```

1 usage

```
public BorrowBook(String memberName, String bookTitle) {
```

```
    this.memberName = memberName;
```

```
    this.bookTitle = bookTitle;
```

```
}
```

```
public void run() {
```

```
    synchronized (Books.class) {
```

```
        if (Books.Title.contains(bookTitle)) {
```

```
            System.out.println(memberName + " borrowed " + bookTitle);
```

```
            Books.Title.remove(bookTitle);
```

```
        } else {
```

```
            System.out.println("Book " + bookTitle + " not available.");
```

```
        }
```

```
    }
```

```
}
```

```
}
```

```
class ReturnBook implements Runnable {  
    2 usages  
    String memberName;  
    3 usages  
    String bookTitle;  
  
    1 usage  
    public ReturnBook(String memberName, String bookTitle) {  
        this.memberName = memberName;  
        this.bookTitle = bookTitle;  
    }  
  
    public void run() {  
        synchronized (Books.class) {  
            System.out.println(memberName + " returned " + bookTitle);  
            Books.Title.add(bookTitle);  
        }  
    }  
}
```

```
public class Main {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        boolean exit = false;  
  
        while (!exit) {  
            System.out.println("Choose an option:");  
            System.out.println("1. Member");  
            System.out.println("2. Staff");  
            System.out.println("3. Exit");  
            int choice = scanner.nextInt();  
  
            switch (choice) {  
                case 1:  
                    handleMember(scanner);  
                    break;  
                case 2:  
                    handleStaff(scanner);  
                    break;  
                case 3:  
                    exit = true;  
                    break;  
                default:  
                    System.out.println("Invalid choice. Please choose again.");  
            }  
        }  
    }  
}
```


1 usage

```
static void handleMember(Scanner scanner) {  
    int k = 0;  
  
    System.out.println("1. Have an account \n2. New member");  
    int ch = scanner.nextInt();  
    System.out.println("Enter Username:");  
    String username = scanner.next();  
    System.out.println("Enter Password:");  
    String password = scanner.next();  
    if (ch == 2) {  
        synchronized (LibraryMember.class) {  
            LibraryMember.Username.add(username);  
            LibraryMember.Password.add(password);  
        }  
    }  
    ch = 1;  
    if (ch == 1) {  
        synchronized (LibraryMember.class) {  
            if (LibraryMember.Username.contains(username)) {  
                if (LibraryMember.Password.contains(password)) {  
                    System.out.println("You are welcome");  
                    k = 1;  
                } else {  
                    while (!LibraryMember.Password.contains(password)) {  
                        System.out.println("\nEnter Password:");  
                        password = scanner.next();  
                    }  
                }  
            }  
        }  
    }  
}
```

```
        System.out.println("You are welcome");
        k = 1;
    }
}
}
}
if (k == 1) {
    System.out.println("What operation do you want to perform?");
    System.out.println("1. Borrow a book");
    System.out.println("2. Return a book");
    int operationChoice = scanner.nextInt();

    if (operationChoice == 1) {
        System.out.println("Enter the title of the book you want to borrow:");
        String bookTitle = scanner.next();

        // Simulate borrowing the book
        Thread borrowThread = new Thread(new BorrowBook(username, bookTitle));
        borrowThread.start();
        try
        {
            borrowThread.join();
        }
        catch (Exception e){}
    } else if (operationChoice == 2) {
        System.out.println("Enter the title of the book you want to return:");
        String bookTitle = scanner.next();
    }
}
```

```

        // Simulate returning the book
        Thread returnThread = new Thread(new ReturnBook(username, bookTitle));
        returnThread.start();
        try
        {
            returnThread.join();
        }
        catch(Exception e){}
    }
}

```

1 usage

```

public static void handleStaff(Scanner scanner) {
    System.out.println("Enter Staff Name:");
    String staffName = scanner.next();
    System.out.println("Enter Role:");
    String role = scanner.next();

    synchronized (LibraryStaff.class) {
        if(LibraryStaff.staffName.contains(staffName)) {
            System.out.println("Staff exists");
        }
        else {
            LibraryStaff.staffName.add(staffName);
            LibraryStaff.Role.add(role);
        }
    }
}

```

```
// Staff operations
System.out.println("What operation do you want to perform?");
System.out.println("1. Add a book");
System.out.println("2. Search for a book");
System.out.println("3. no operation");
int operationChoice = scanner.nextInt();

switch (operationChoice) {
    case 1:
        System.out.println("Enter the title of the book:");
        String title = scanner.next();
        System.out.println("Enter the author of the book:");
        String author = scanner.next();
        System.out.println("Enter the price of the book:");
        int price = scanner.nextInt();
        Thread addBookThread = new Thread(new AddBook(staffName, title, author, price));
        addBookThread.start();
        break;
    case 2:
        System.out.println("Enter the title of the book you want to search:");
        String searchTitle = scanner.next();
        Thread searchBookThread = new Thread(new SearchBook(staffName, searchTitle));
        searchBookThread.start();
        try
        {
            searchBookThread.join();
        }
}
```

```
        catch(Exception e){}
        break;
    case 3:
        break;
    default:
        System.out.println("Invalid choice.");
    }
}
}
```

OUTPUT


```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA
```

Choose an option:

1. Member
2. Staff
3. Exit

2

Enter Staff Name:

ABHI

Enter Role:

OWNER

What operation do you want to perform?

1. Add a book
2. Search for a book
3. no operation

1

Enter the title of the book:

a

Enter the author of the book:

A

Enter the price of the book:

12

Choose an option:

1. Member
2. Staff
3. Exit

Book 'a' added to the library by ABHI

2

Enter Staff Name:

JITHU

Enter Role:

MANAGER

What operation do you want to perform?

1. Add a book
2. Search for a book
3. no operation

2

Enter the title of the book you want to search:

A

Book not found.

Choose an option:

1. Member
2. Staff
3. Exit

2

Enter Staff Name:

JITHU

Enter Role:

MANAGER

Staff exists

What operation do you want to perform?

1. Add a book
2. Search for a book
3. no operation

2

Enter the title of the book you want to search:

a

Book found:

Title: a

Author: A

Price: 12

Choose an option:

1. Member

2. Staff

3. Exit

1

1. Have an account

2. New member

2

Enter Username:

KAIL

Enter Password:

KKK

You are welcome

What operation do you want to perform?

1. Borrow a book

2. Return a book

1

Enter the title of the book you want to borrow:

a

KAIL borrowed a

Choose an option:

1. Member
2. Staff
3. Exit

1

1. Have an account

2. New member

1

Enter Username:

KAIL

Enter Password:

KKK

You are welcome

What operation do you want to perform?

1. Borrow a book
2. Return a book

2

Enter the title of the book you want to return:

a

KAIL returned a

Choose an option:

1. Member
2. Staff
3. Exit

3

Process finished with exit code 0