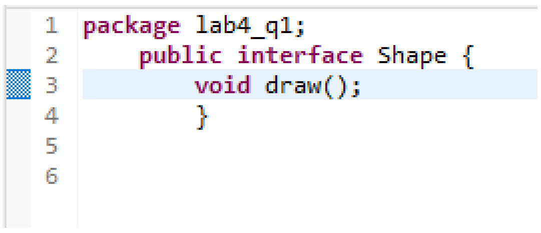
Lab Session 04

Use Façade Design Pattern to simplify a complex software system

EXERCISES

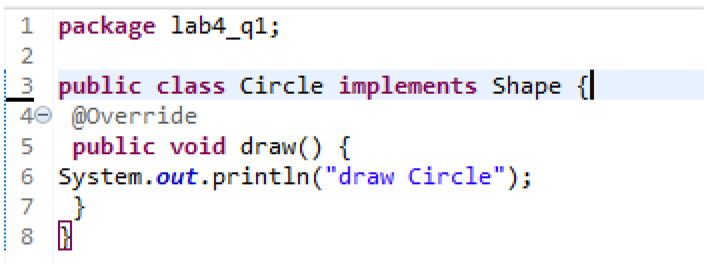
1. Implement the façade design pattern for the diagram given below:

Shape.java



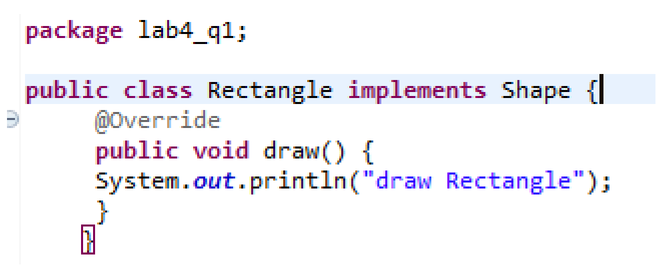
```java
package lab4_q1;
    public interface Shape {
        void draw();
        }
```

Circle.java



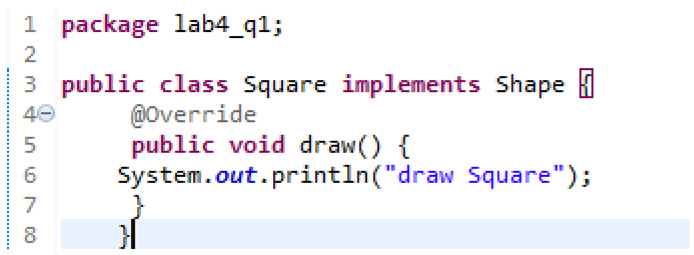
```java
package lab4_q1;

public class Circle implements Shape {
 @Override
 public void draw() {
System.out.println("draw Circle");
 }
}
```

Rectangle.java



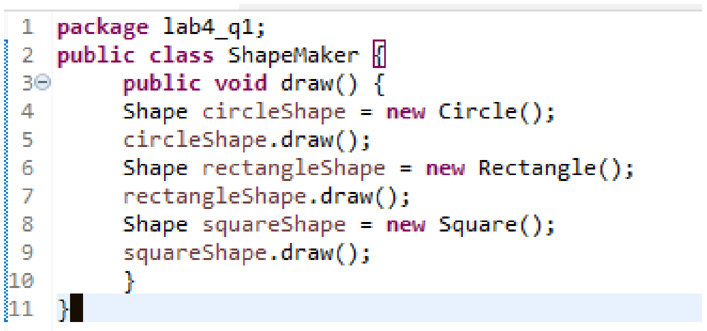
```java
package lab4_q1;

public class Rectangle implements Shape {
     @Override
     public void draw() {
     System.out.println("draw Rectangle");
     }
    }
```

Square.java



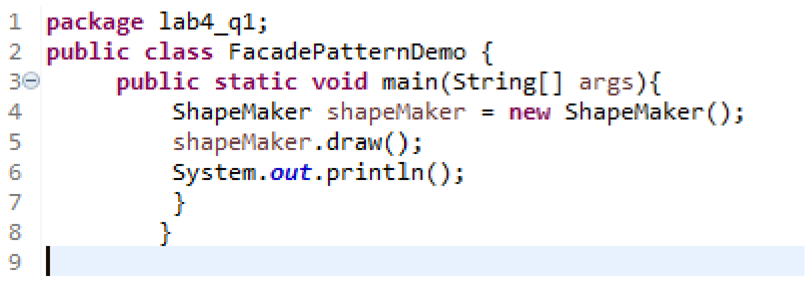
```java
package lab4_q1;

public class Square implements Shape {
     @Override
     public void draw() {
    System.out.println("draw Square");
     }
    }
```

ShapeMaker.java



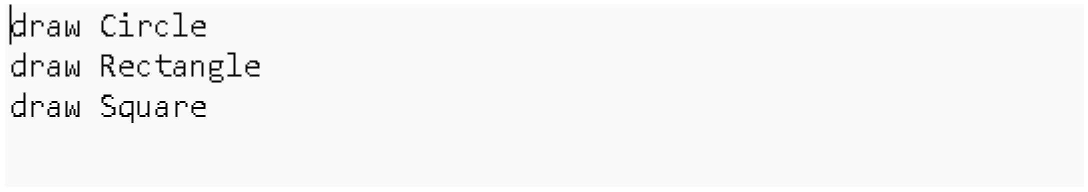
```java
package lab4_q1;
public class ShapeMaker {
    public void draw() {
    Shape circleShape = new Circle();
    circleShape.draw();
    Shape rectangleShape = new Rectangle();
    rectangleShape.draw();
    Shape squareShape = new Square();
    squareShape.draw();
    }
}
```

FacadeDemoPattern.java



```java
package lab4_q1;
public class FacadePatternDemo {
    public static void main(String[] args){
        ShapeMaker shapeMaker = new ShapeMaker();
        shapeMaker.draw();
        System.out.println();
        }
    }
```

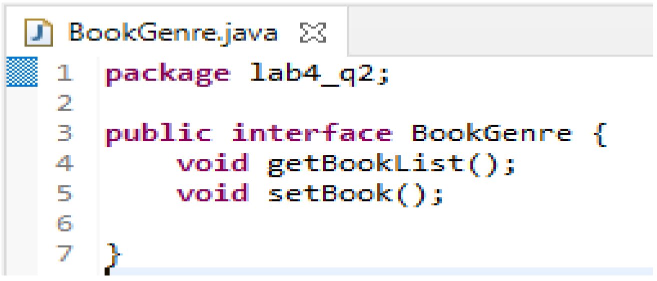
Output:



```
draw Circle
draw Rectangle
draw Square
```

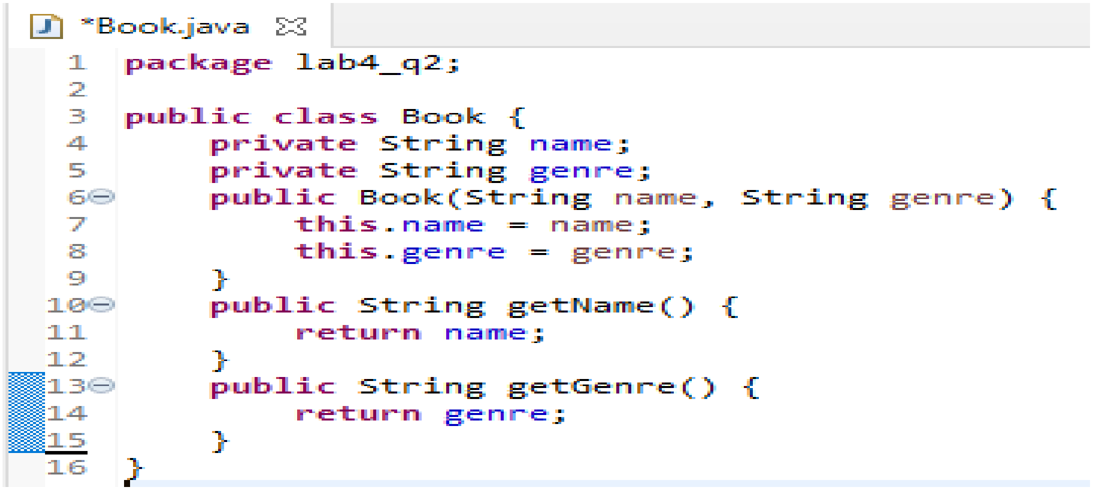
2. For a library system, there are three classes namely Fiction, Non Fiction and Technology. All of these classes consists of getBookList() method to get the books of the respective genre. All of these class implement BookGenre interface. Implement a façade of LibraryService containing a method of BookBorrow(). Also demonstrate the working of façade using a client.

Step 1: Create the BookGenre interface.


BookGenre.java

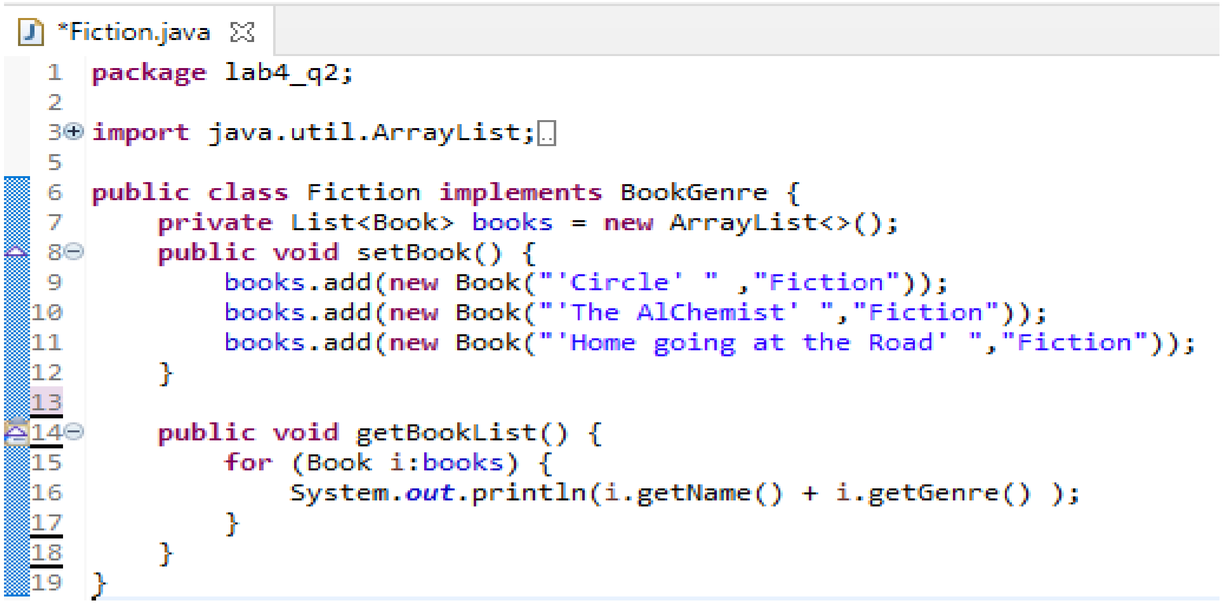
```java
package lab4_q2;

public interface BookGenre {
    void getBookList();
    void setBook();

}
```

Step 2: Create a simple Book class that contains genre and name.

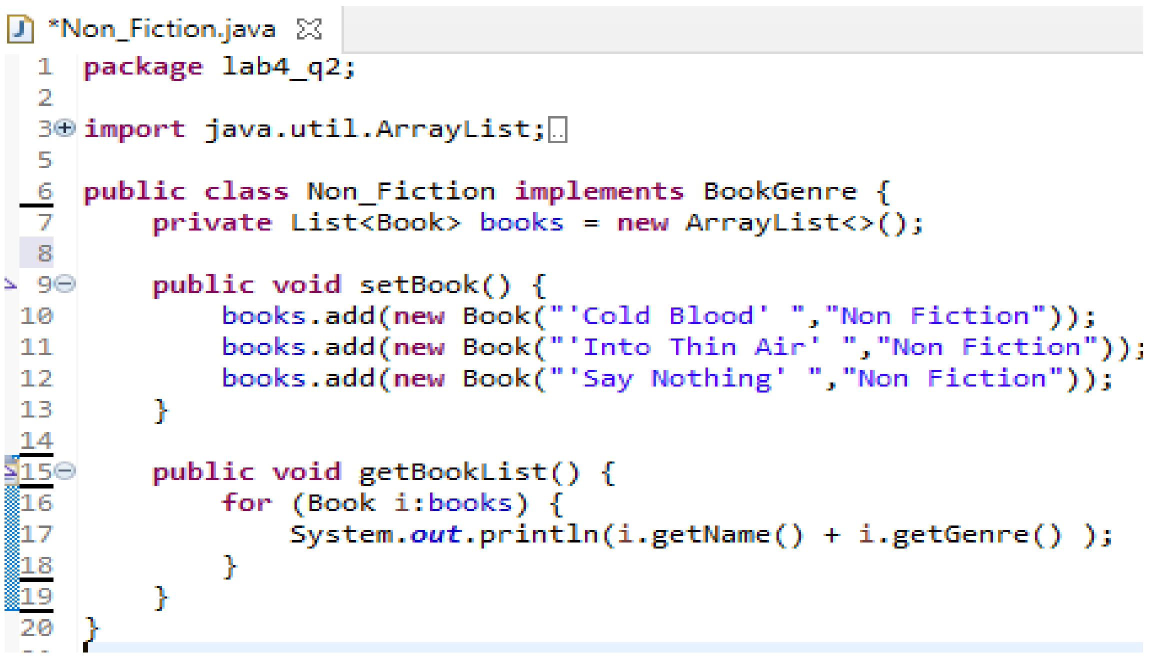

*Book.java

```java
package lab4_q2;

public class Book {
    private String name;
    private String genre;
    public Book(String name, String genre) {
        this.name = name;
        this.genre = genre;
    }
    public String getName() {
        return name;
    }
    public String getGenre() {
        return genre;
    }
}
```

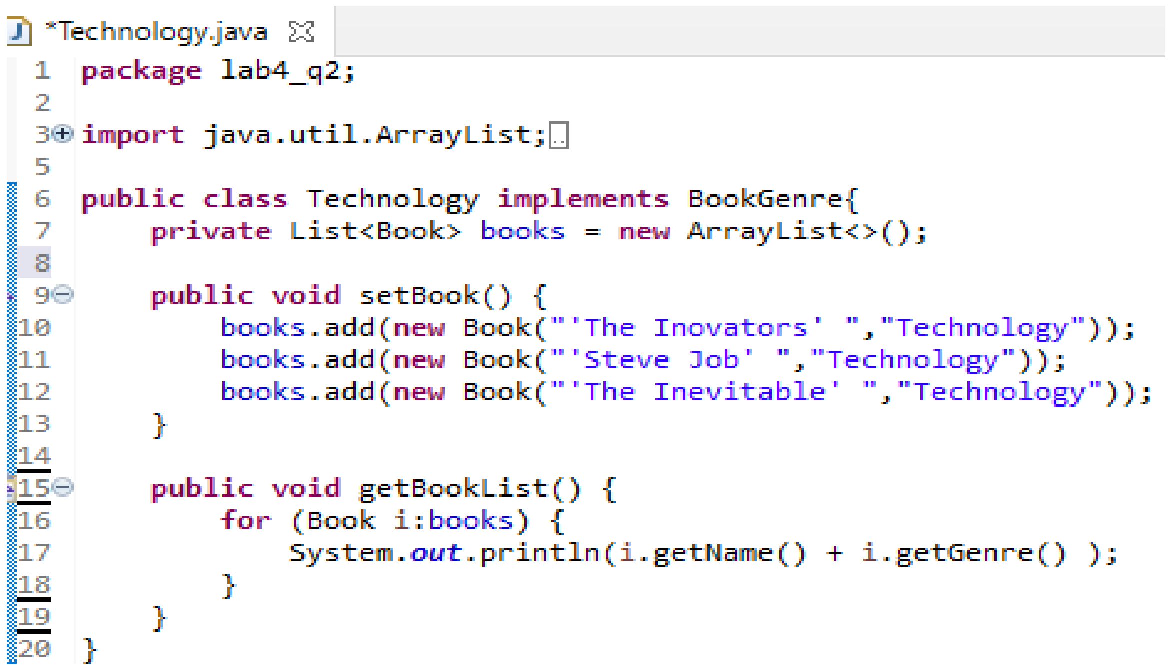
Step 3: Create the Concrete classes Fiction, Non_Fiction and Technology that inherit BookGenre.


*Fiction.java

```java
package lab4_q2;

import java.util.ArrayList;

public class Fiction implements BookGenre {
    private List<Book> books = new ArrayList<>();
    public void setBook() {
        books.add(new Book("'Circle' " ,"Fiction"));
        books.add(new Book("'The AlChemist' ","Fiction"));
        books.add(new Book("'Home going at the Road' ","Fiction"));
    }

    public void getBookList() {
        for (Book i:books) {
            System.out.println(i.getName() + i.getGenre() );
        }
    }
}
```

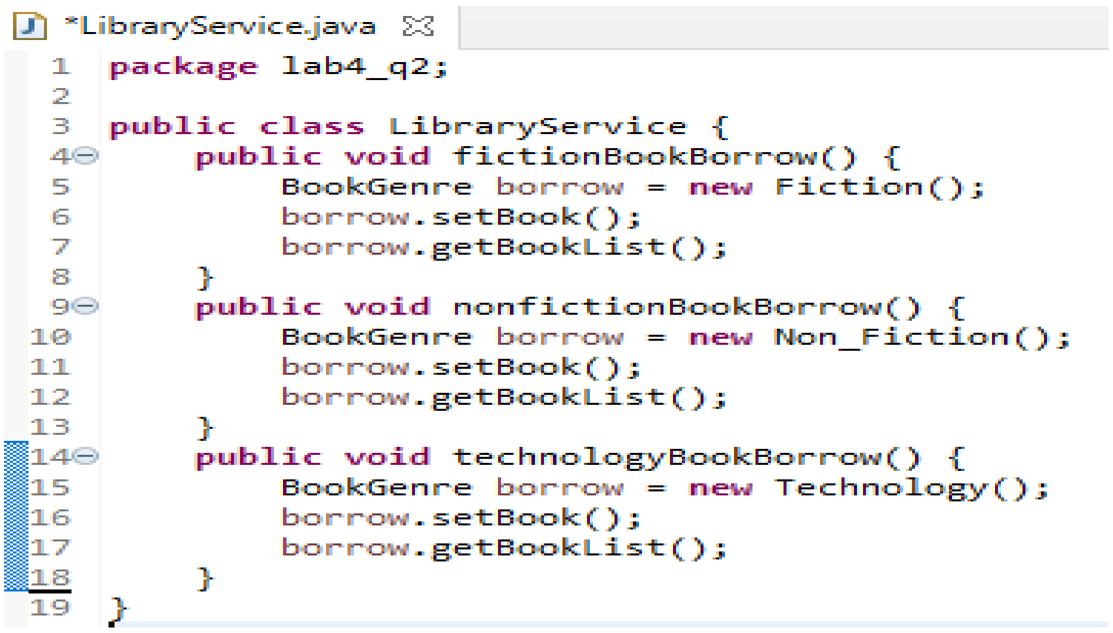

*Non_Fiction.java

```
package lab4_q2;

import java.util.ArrayList;

public class Non_Fiction implements BookGenre {
    private List<Book> books = new ArrayList<>();

    public void setBook() {
        books.add(new Book("'Cold Blood' ","Non Fiction"));
        books.add(new Book("'Into Thin Air' ","Non Fiction"));
        books.add(new Book("'Say Nothing' ","Non Fiction"));
    }

    public void getBookList() {
        for (Book i:books) {
            System.out.println(i.getName() + i.getGenre() );
        }
    }
}
```


*Technology.java

```
package lab4_q2;

import java.util.ArrayList;

public class Technology implements BookGenre{
    private List<Book> books = new ArrayList<>();

    public void setBook() {
        books.add(new Book("'The Inovators' ","Technology"));
        books.add(new Book("'Steve Job' ","Technology"));
        books.add(new Book("'The Inevitable' ","Technology"));
    }

    public void getBookList() {
        for (Book i:books) {
            System.out.println(i.getName() + i.getGenre() );
        }
    }
}
```

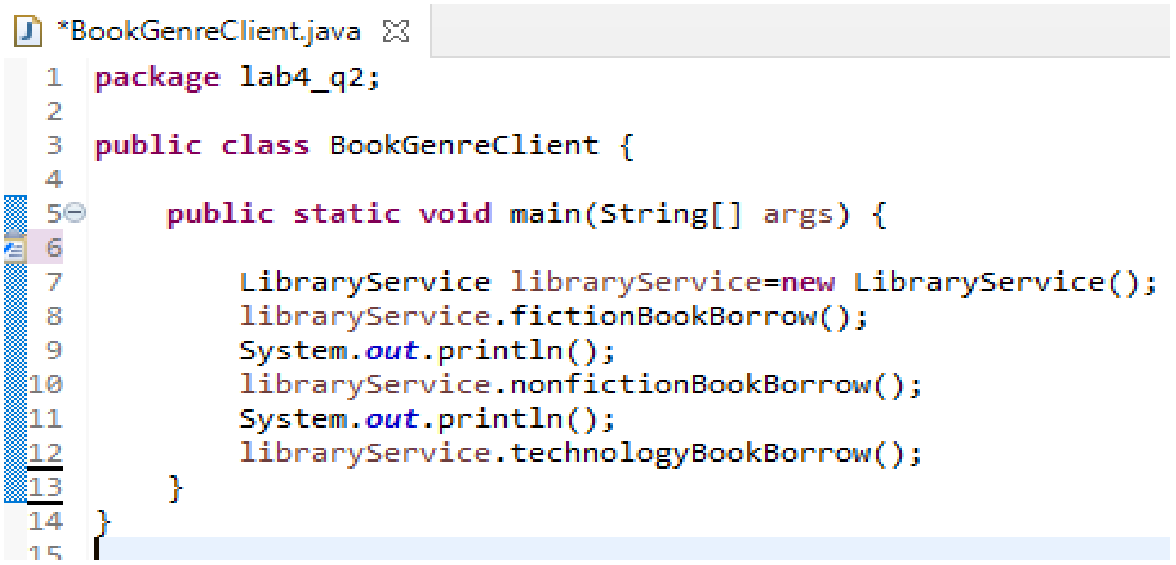
Step 4: Create the LibraryService Facade class that will hide all of the logic from the Client.



*LibraryService.java

```java
package lab4_q2;

public class LibraryService {
    public void fictionBookBorrow() {
        BookGenre borrow = new Fiction();
        borrow.setBook();
        borrow.getBookList();
    }
    public void nonfictionBookBorrow() {
        BookGenre borrow = new Non_Fiction();
        borrow.setBook();
        borrow.getBookList();
    }
    public void technologyBookBorrow() {
        BookGenre borrow = new Technology();
        borrow.setBook();
        borrow.getBookList();
    }
}
```

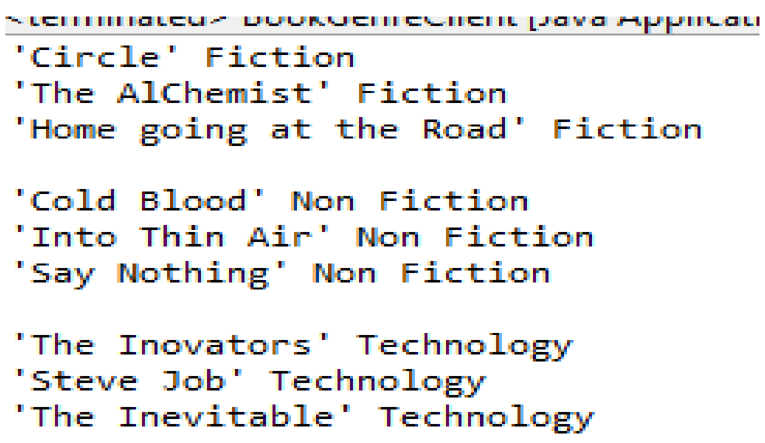
Step 5: Create the BookGenreClient.



*BookGenreClient.java

```java
package lab4_q2;

public class BookGenreClient {

    public static void main(String[] args) {

        LibraryService libraryService=new LibraryService();
        libraryService.fictionBookBorrow();
        System.out.println();
        libraryService.nonfictionBookBorrow();
        System.out.println();
        libraryService.technologyBookBorrow();
    }
}
```

Output:



```
'Circle' Fiction
'The AlChemist' Fiction
'Home going at the Road' Fiction

'Cold Blood' Non Fiction
'Into Thin Air' Non Fiction
'Say Nothing' Non Fiction

'The Inovators' Technology
'Steve Job' Technology
'The Inevitable' Technology
```