1.Write a Java program that performs the following operations on a given string: find its length, convert it to uppercase, extract a substring, and replace a character.​

public class strings{

public static void main(String[] args){

String str = "Javaprograms";

System.out.println("The given string is : " + str);

int len = str.length();

System.out.println("The size of the string " + str + " : " + len);

String str1 = str.toUpperCase();

System.out.println("Converting the string to uppercase : " + str1);

String sub = str.substring(4,11);

System.out.println("Extracting a substing : " + sub);

String rep = str.replace("v","w");

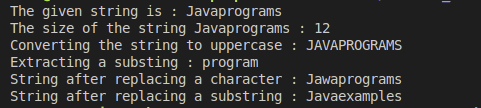
System.out.println("String after replacing a character : " + rep);

String reps = str.replace("programs","examples");

System.out.println("String after replacing a substring : " + reps);

}

}



2. Write a Java program to parse a string into different primitive data types using wrapper class methods like parseInt, parseDouble, parseBoolean, etc., and convert primitive types to strings using valueOf.

public class wrapper{

public static void main(String[] args){

String str = "12435";

System.out.println("The given string is : " + str);

int i = Integer.parseInt(str);

System.out.println("Converting string to integer : " + i);

double d = Double.parseDouble(str);

System.out.println("Converting string to double : " + d);

float f = Float.parseFloat(str);

System.out.println("Converting string to float : " + f);

String s = "true";

System.out.println("The string : " +s);

boolean b = Boolean.parseBoolean(s);

System.out.println("Converting string to boolean : " +b);

String n = String.valueOf(i);

System.out.println("Converting int " + i + " back to string : " +n);

String m = String.valueOf(d);

System.out.println("Converting double " + d + " back to string : " +m);

String x = String.valueOf(f);

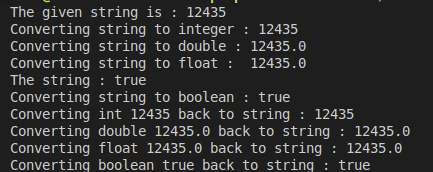
System.out.println("Converting float " + f + " back to string : " +x);

String y = String.valueOf(b);

System.out.println("Converting boolean " + b + " back to string : " +y);

}

}



3. Write a Java program to sort an array of integers in ascending order using a sorting algorithm of your choice.​

public class sort{

public static void main(String[] args){

int[] a = {45,78,23,17,83,56};

System.out.println("Unsorted array : ");

for(int i : a){

System.out.print(i + " ");

}

System.out.println();

sort(a);

}

public static void sort(int[] a){

int n = a.length;

boolean swap;

for( int i=0; i<n-1; i++){

swap=false;

for( int j=0; j<n-1-i; j++){

if(a[j]>a[j+1]){

int temp;

temp=a[j];

a[j]=a[j+1];

a[j+1]=temp;

swap=true;

}

}

if(!swap){

break;

}

}

System.out.println("Sorted array : ");

for(int i : a){

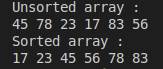
System.out.print(i + " ");

}

System.out.println();

}

}



4. Use an ArrayList to store the list of books. Each book should have attributes such as title, author, ISBN, and price.Implement functionalities to add new books, remove existing books, and display all books in the library.​

import java.util.ArrayList;

import java.util.Scanner;

public class arraylist{

public static void main(String[] args){

Library l = new Library();

Scanner sc = new Scanner(System.in);

while(true){

System.out.println("Library Menu : ");

System.out.println("1.Add a book");

System.out.println("2.Remove a book");

System.out.println("3.Display all books");

System.out.println("4.Exit");

System.out.println("Enter a choice : ");

int ch = sc.nextInt();

sc.nextLine();

switch(ch){

case 1 :

System.out.println("Enter the Title : ");

String title = sc.nextLine();

System.out.println("Enter the Author : ");

String author = sc.nextLine();

System.out.println("Enter the ISBN : ");

String isbn = sc.nextLine();

System.out.println("Enter the price : ");

Double price = sc.nextDouble();

sc.nextLine();

Book b = new Book(title,author,isbn,price);

l.addbook(b);

break;

case 2 :

System.out.println("Enter the ISBN of the book to remove : ");

String remisbn = sc.nextLine();

l.removebook(remisbn);

break;

case 3 :

l.displaybooks();

break;

case 4 :

System.out.println("Exit..");

sc.close();

return;

case 5 :

System.out.println("Invalid choice.");

}

}

}

}

class Library{

private ArrayList<Book> books;

public Library(){

books = new ArrayList<>();

}

public void addbook(Book book){

books.add(book);

System.out.println("Book added : " + book);

}

public void removebook(String isbn){

Book rem = null;

for(Book book : books){

if(book.getISBN().equals(isbn)){

rem = book;

break;

}

}

if(rem!=null){

books.remove(rem);

System.out.println("The Book removed : " + rem);

}

else{

System.out.println("The book with ISBN " + isbn + " not found");

}

}

public void displaybooks(){

if(books.isEmpty()){

System.out.println("No books in the library.");

}

else{

System.out.println("The Books in the library are : ");

for(Book book : books){

System.out.println(book);

}

}

}

}

class Book{

private String title;

private String author;

private String isbn;

private double price;

public Book(String title, String author, String isbn, double price){

this.title = title;

this.author = author;

this.isbn = isbn;

this.price = price;

}

public String getTitle(){

return title;

}

public String getAuthor(){

return author;

}

public String getISBN(){

return isbn;

}

public double getPrice(){

return price;

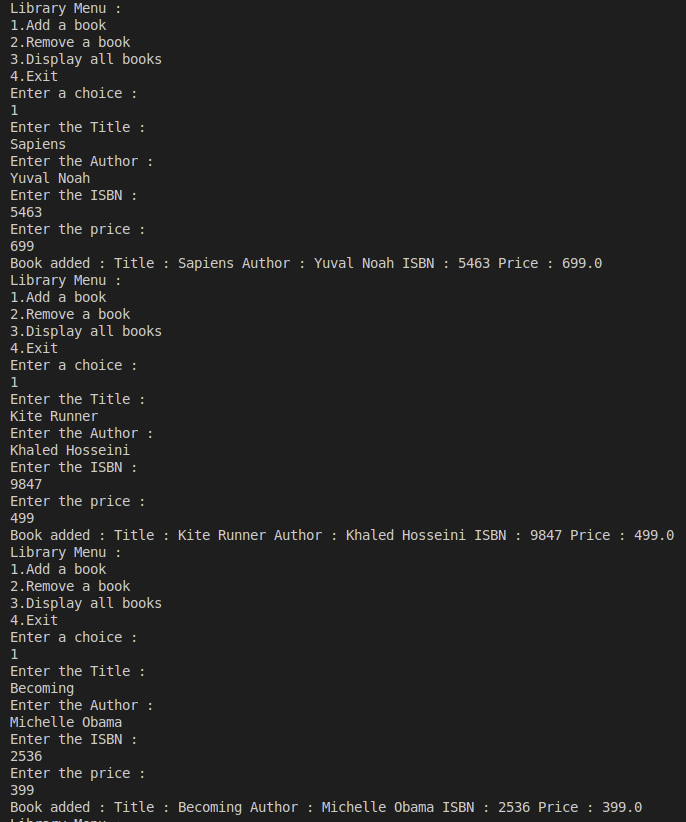
}

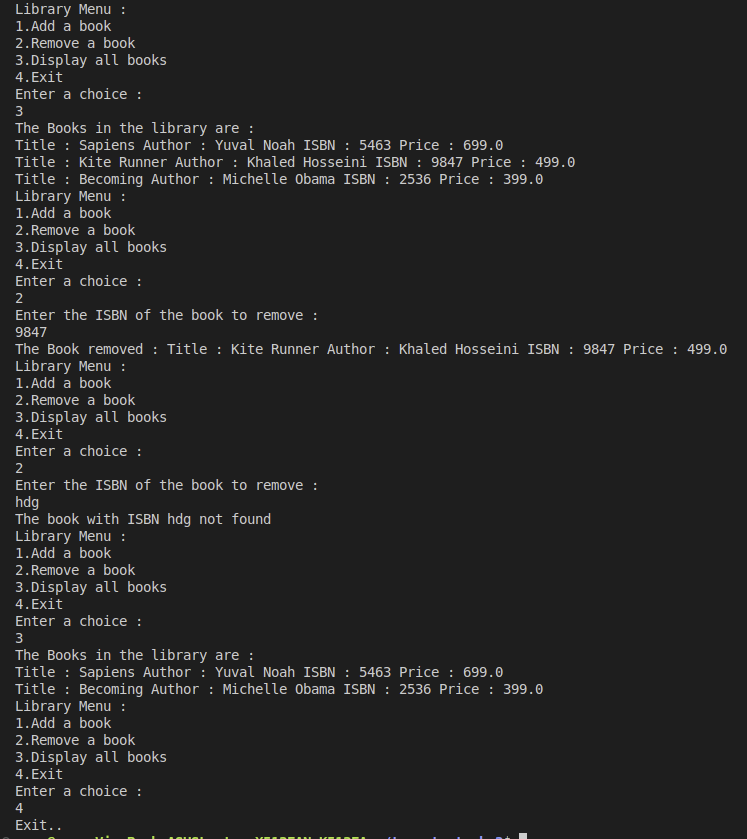
public String toString(){

return "Title : " + title + " Author : " + author + " ISBN : " + isbn + " Price : " + price;

}

}





5. Use a HashSet to manage the unique genres available in the library.Ensure that new genres can be added without duplicating existing genres.​

import java.util.HashSet;

import java.util.Set;

import java.util.Scanner;

import java.util.Objects;

public class hashset{

public static void main(String[] args){

Library l = new Library();

Scanner sc = new Scanner(System.in);

while(true){

System.out.println("Library Menu : ");

System.out.println("1.Add a book");

System.out.println("2.Remove a book");

System.out.println("3.Display all genres");

System.out.println("4.Display all books");

System.out.println("5.Exit");

System.out.println("Enter a choice : ");

int ch = sc.nextInt();

sc.nextLine();

switch(ch){

case 1 :

System.out.println("Enter the Title : ");

String title = sc.nextLine();

System.out.println("Enter the Author : ");

String author = sc.nextLine();

System.out.println("Enter the ISBN : ");

String isbn = sc.nextLine();

System.out.println("Enter the Genre : ");

String genre = sc.nextLine();

System.out.println("Enter the price : ");

Double price = sc.nextDouble();

sc.nextLine();

Book b = new Book(title,author,isbn,genre,price);

l.addbook(b);

break;

case 2 :

System.out.println("Enter the ISBN of the book to remove : ");

String remisbn = sc.nextLine();

l.removebook(remisbn);

break;

case 3 :

l.displaygenres();

break;

case 4 :

l.displaybooks();

break;

case 5 :

System.out.println("Exit..");

sc.close();

return;

default :

System.out.println("Invalid choice.");

}

}

}

}

class Book{

private String title;

private String author;

private String isbn;

private double price;

public String genre;

public Book(String title, String author, String isbn, String genre, double price){

this.title = title;

this.author = author;

this.isbn = isbn;

this.genre = genre;

this.price = price;

}

public String getTitle(){

return title;

}

public String getAuthor(){

return author;

}

public String getISBN(){

return isbn;

}

public String getGenre(){

return genre;

}

public double getPrice(){

return price;

}

public boolean equals(Object o){

if(this == o){

return true;

}

if(o==null || getClass()!=o.getClass()){

return false;

}

Book book = (Book) o;

return isbn.equals(book.isbn);

}

public int hashCode(){

return Objects.hash(isbn);

}

public String toString(){

return "Title : " + title + ", Author : " + author + ", ISBN : " + isbn + ", Genre : " + genre + ", Price : " + price;

}

}

class Library{

HashSet<Book> books = new HashSet<>();

HashSet<String> genres = new HashSet<>();

public void addbook(Book book){

books.add(book);

genres.add(book.genre);

System.out.println("Book added : " + book);

}

public void removebook(String isbn){

Book rem = null;

for(Book book : books){

if(book.getISBN().equals(isbn)){

rem = book;

break;

}

}

if(rem!=null){

books.remove(rem);

System.out.println("The Book removed : " + rem);

}

else{

System.out.println("The book with ISBN " + isbn + " not found");

}

}

public void displaybooks(){

if(books.isEmpty()){

System.out.println("No books in the library.");

}

else{

System.out.println("The Books in the library are : ");

for(Book book : books){

System.out.println(book);

}

}

}

public void displaygenres(){

String[] genrelist = genres.toArray(new String[0]);

if(genres.isEmpty()){

System.out.println("No genres in the library.");

}

else{

System.out.println("The genres in the library are : ");

for(String genre : genres){

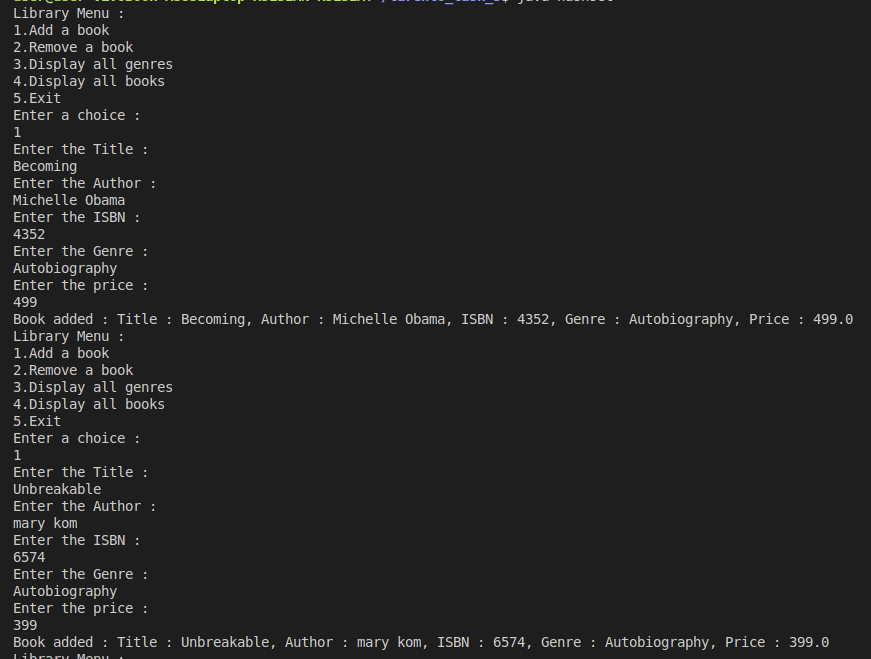
System.out.println(genre);

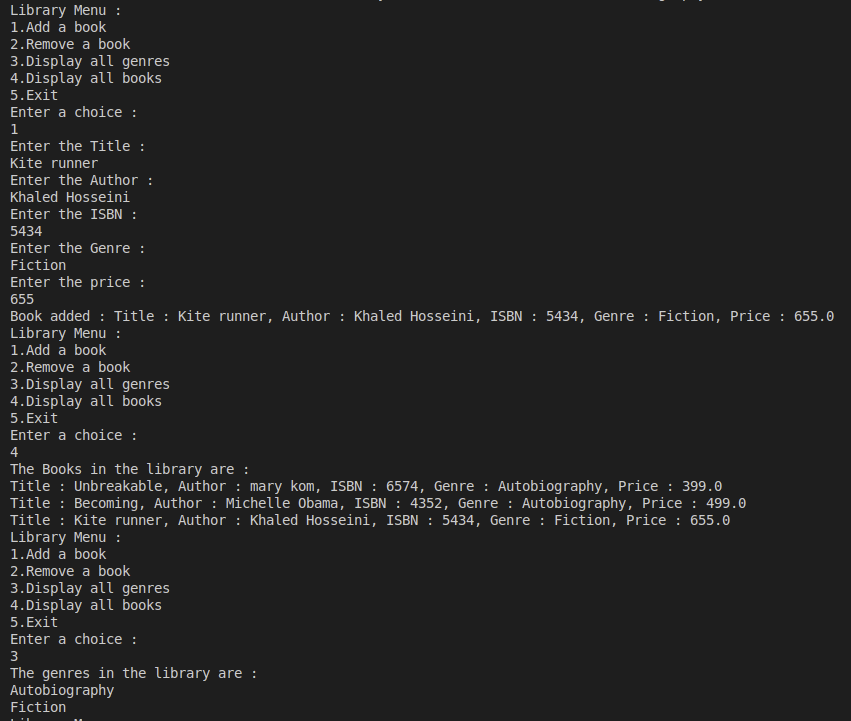
}

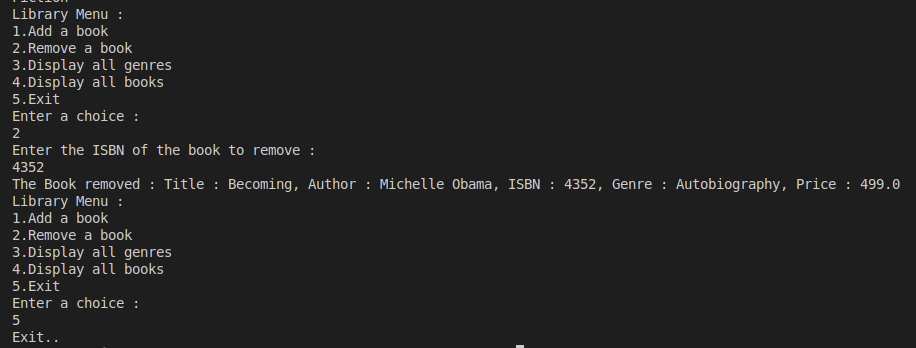
}

}

}







6. Use a HashMap to map ISBN numbers to books for quick lookup.Implement functionalities to add, update, and retrieve book details using ISBN.​

import java.util.HashMap;

import java.util.Map;

import java.util.Scanner;

public class hashmap{

public static void main(String[] args){

Library l = new Library();

Scanner sc = new Scanner(System.in);

while(true){

System.out.println("Library Menu : ");

System.out.println("1.Add a book");

System.out.println("2.Update a book");

System.out.println("3.Retrieve a book");

System.out.println("4.Display all books");

System.out.println("5.Exit");

System.out.println("Enter a choice : ");

int ch = sc.nextInt();

sc.nextLine();

switch(ch){

case 1 :

System.out.println("Enter the Title : ");

String title = sc.nextLine();

System.out.println("Enter the Author : ");

String author = sc.nextLine();

System.out.println("Enter the ISBN : ");

String isbn = sc.nextLine();

System.out.println("Enter the price : ");

Double price = sc.nextDouble();

sc.nextLine();

Book b = new Book(title,author,isbn,price);

l.addbook(b);

break;

case 2 :

System.out.println("Enter the ISBN of the book to update : ");

String upisbn = sc.nextLine();

System.out.println("Enter new Title : ");

String newtitle = sc.nextLine();

System.out.println("Enter new Author : ");

String newauthor = sc.nextLine();

System.out.println("Enter new price : ");

double newprice = sc.nextDouble();

sc.nextLine();

l.updatebook(upisbn, newtitle, newauthor, newprice);

break;

case 3 :

System.out.println("Enter ISBN of the book to retrieve : ");

String retisbn = sc.nextLine();

l.retrievebook(retisbn);

break;

case 4 :

l.displaybooks();

break;

case 5 :

System.out.println("Exit..");

sc.close();

return;

default :

System.out.println("Invalid choice.");

}

}

}

}

class Book{

private String title;

private String author;

private String isbn;

private double price;

public Book(String title, String author, String isbn, double price){

this.title = title;

this.author = author;

this.isbn = isbn;

this.price = price;

}

public String getTitle(){

return title;

}

public String getAuthor(){

return author;

}

public String getISBN(){

return isbn;

}

public double getPrice(){

return price;

}

public void setTitle(String title){

this.title = title;

}

public void setAuthor(String author){

this.author = author;

}

public void setPrice(double price){

this.price = price;

}

public String toString(){

return "Title : " + title + " Author : " + author + " ISBN : " + isbn + " Price : " + price;

}

}

class Library{

private Map<String, Book> books;

public Library(){

books = new HashMap<>();

}

public void addbook(Book book){

if(books.containsKey(book.getISBN())){

System.out.println("The book with ISBN " + book.getISBN() + " already exists.");

}

else{

books.put(book.getISBN(), book);

System.out.println("Book added : " + book);

}

}

public void updatebook(String isbn, String title, String author, double price){

Book book = books.get(isbn);

if(book!=null){

book.setTitle(title);

book.setAuthor(author);

book.setPrice(price);

System.out.println("Book updated : " + book);

}

else{

System.out.println("Book with ISBN " + isbn + " not found.");

}

}

public void retrievebook(String isbn){

Book book = books.get(isbn);

if(book!=null){

System.out.println("Book details : "+book);

}

else{

System.out.println("Book with ISBN "+ isbn + " not found.");

}

}

public void displaybooks(){

if(books.isEmpty()){

System.out.println("No books in the library.");

}

else{

System.out.println("The Books in the library are : ");

for(Book book : books.values()){

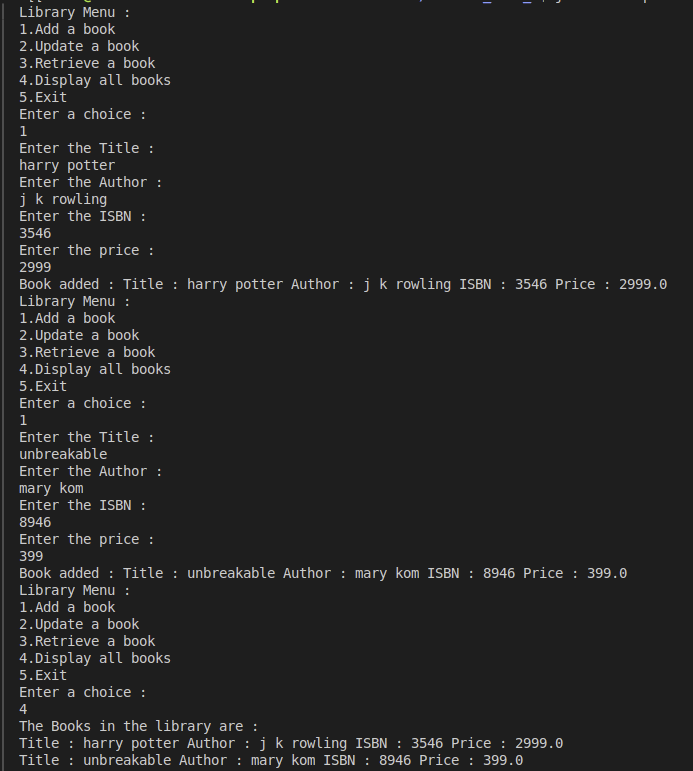
System.out.println(book);

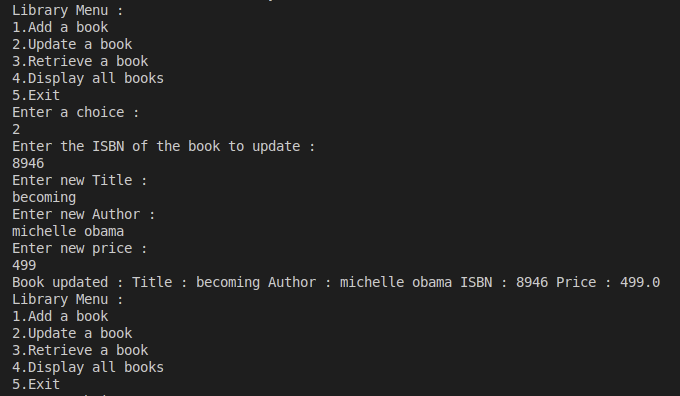
}

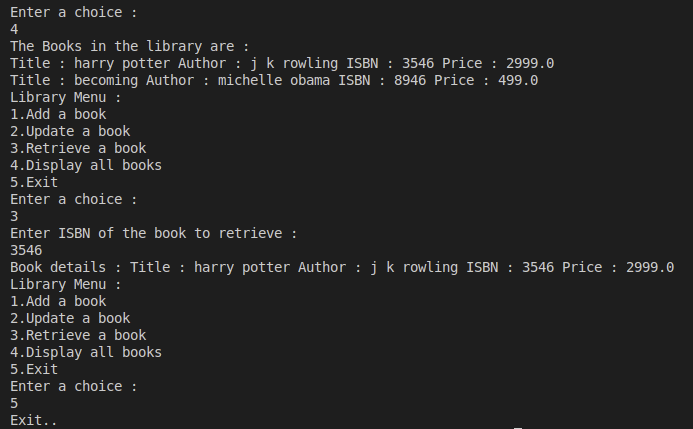
}

}

}







7.Implement a custom exception called ProductNotFoundException that is thrown when a product is not found in the inventory.Use try, catch, finally, throw, and throws to handle exceptions appropriately.

​

import java.util.HashSet;

import java.util.Set;

import java.util.Scanner;

import java.util.Objects;

public class productexception{

public static void main(String[] args){

Library l = new Library();

Scanner sc = new Scanner(System.in);

while(true){

System.out.println("Library Menu : ");

System.out.println("1.Add a book");

System.out.println("2.Remove a book");

System.out.println("3.Display all books");

System.out.println("4.Exit");

System.out.println("Enter a choice : ");

int ch = sc.nextInt();

sc.nextLine();

switch(ch){

case 1 :

System.out.println("Enter the Title : ");

String title = sc.nextLine();

System.out.println("Enter the Author : ");

String author = sc.nextLine();

System.out.println("Enter the ISBN : ");

String isbn = sc.nextLine();

System.out.println("Enter the price : ");

Double price = sc.nextDouble();

sc.nextLine();

Book b = new Book(title,author,isbn,price);

l.addbook(b);

break;

case 2 :

try{

System.out.println("Enter the ISBN of the book to remove : ");

String remisbn = sc.nextLine();

l.removebook(remisbn);

}

catch(ProductNotFoundException e){

System.out.println(e.getMessage());

}

break;

case 3 :

l.displaybooks();

break;

case 4 :

System.out.println("Exit..");

sc.close();

return;

case 5 :

System.out.println("Invalid choice.");

}

}

}

}

class Book{

private String title;

private String author;

private String isbn;

private double price;

public Book(String title, String author, String isbn, double price){

this.title = title;

this.author = author;

this.isbn = isbn;

this.price = price;

}

public String getTitle(){

return title;

}

public String getAuthor(){

return author;

}

public String getISBN(){

return isbn;

}

public double getPrice(){

return price;

}

public boolean equals(Object o){

if(this == o){

return true;

}

if(o==null || getClass()!=o.getClass()){

return false;

}

Book book = (Book) o;

return isbn.equals(book.isbn);

}

public int hashCode(){

return Objects.hash(isbn);

}

public String toString(){

return "Title : " + title + " Author : " + author + " ISBN : " + isbn + " Price : " + price;

}

}

class Library{

private Set<Book> books;

public Library(){

books = new HashSet<>();

}

public void addbook(Book book){

books.add(book);

System.out.println("Book added : " + book);

}

public void removebook(String isbn) throws ProductNotFoundException{

Book rem = null;

for(Book book : books){

if(book.getISBN().equals(isbn)){

rem = book;

break;

}

}

if(rem!=null){

books.remove(rem);

System.out.println("The Book removed : " + rem);

}

else{

throw new ProductNotFoundException("The book with ISBN " + isbn + " not found");

}

}

public void displaybooks(){

if(books.isEmpty()){

System.out.println("No books in the library.");

}

else{

System.out.println("The Books in the library are : ");

for(Book book : books){

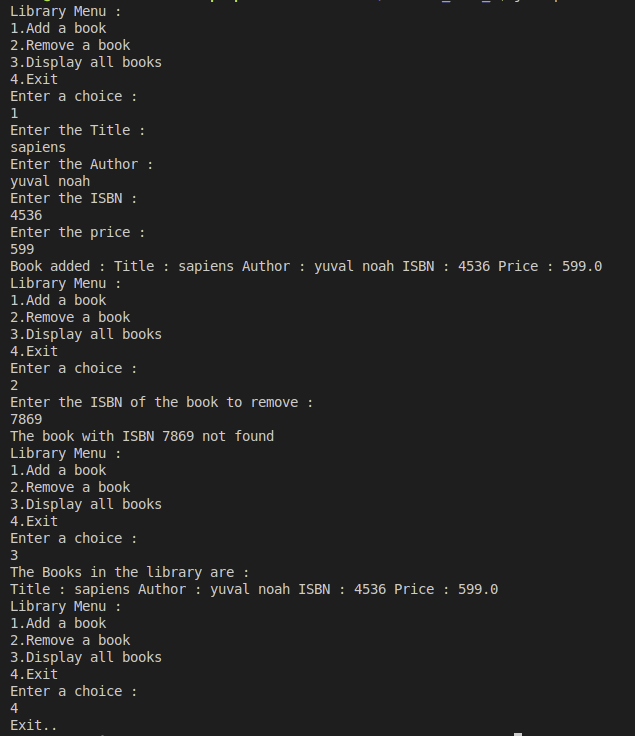
System.out.println(book);

}

}

}

}



8. Use any one of the file handling to read employee records from a text file and write employee records to a text file.

import java.io.\*;

class Employee {

int id;

String name;

String position;

public Employee(int id, String name, String position) {

this.id = id;

this.name = name;

this.position = position;

}

@Override

public String toString() {

return id + "," + name + "," + position;

}

}

public class readWrite {

public static void readEmployee(String fileName) {

try (BufferedReader br = new BufferedReader(new FileReader(fileName))) {

String line;

while ((line = br.readLine()) != null) {

if (line.startsWith("ID")) {

continue;

}

String[] details = line.split(",");

int id = Integer.parseInt(details[0]);

String name = details[1];

String position = details[2];

Employee employee = new Employee(id, name, position);

System.out.println("Read Employee: " + employee);

}

} catch (IOException e) {

System.out.println("An error occurred while reading the file");

e.printStackTrace();

}

}

public static void writeEmployee(String fileName, Employee[] employees) {

try (BufferedWriter bw = new BufferedWriter(new FileWriter(fileName, true))) {

for (Employee employee : employees) {

bw.write(employee.toString());

bw.newLine();

}

System.out.println("Employee details written");

} catch (IOException e) {

System.out.println("An error occurred during writing");

e.printStackTrace();

}

}

public static void main(String[] args) {

String inputFileName = "/home/user/tarento\_task\_3/employee.txt";

String outputFileName = "/home/user/tarento\_task\_3/employee.txt";

readEmployee(inputFileName);

Employee[] newEmployees = {

new Employee(4, "Navjit", "Manager"),

new Employee(5, "Zayn", "Developer")

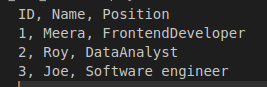
};

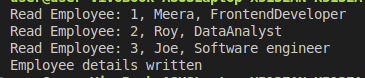
writeEmployee(outputFileName, newEmployees);

}

}

​



employee.txt

