**11.MINI PROJECT**

**AIM:**

To develop Library Management application using the concepts of JAVA.

**ALGORITHM:**

Step 1: Start

Step 2: Create arrays for storing book details.

Step 3: Initialize variables for invoice details.

Step 4: Set up the scanner for user input.

Step 5: Display Book Information like List of available books with their codes, names, prices, and quantities.

Step 6: Loop: Use a do-while loop to repeatedly ask the user for book codes and quantities.

Step 7: If book code is valid, prompt for the required quantity. If the book code is not valid, throw InvalidBookCodeException.

Step 7(1): If the quantity is valid, update the invoice details and decrement the available quantity. If invalid, throw InvalidQuantityException.

Step 8: Print the invoice details for the selected books, including book code, name, quantity, amount, grand total and cost.

Step 9: Use a 'BufferedWriter' to write the invoice details to a file named “invoice.txt”.

Step 10: Catch & handle the exception ‘InvalidBookCodeException’ & ‘InvalidQuantityException’

Step 11: Print corresponding error messages.

Step 12: Display a thank-you message.

Step 13: End.

**Flowchart:**

Is book code valid?

Input book code

Start

Print Book details

No

Print error message

Get required quantity

Yes

No

Is quantity valid & available?

Print error message

Yes

Save these details in the array

Yes

Is ch = y ?

End

Write these invoice details in invoice.txt

Calculating Cost = Quantity\*Amount, Grand Total = Cost of all books

Print the Invoice

with all necessary details

**Program:**

//Import necessary packages

import java.util.Scanner;

import java.io.\*;

//User-defined Exception for Invalid Book Code Exception

class InvalidBookCodeException extends Exception{

InvalidBookCodeException(String message){

super(message);

}

}

//User-defined Exception for Invalid Quantity Exception

class InvalidQuantityException extends Exception{

InvalidQuantityException(String message){

super(message);

}

}

class LibraryManagementSystem {

//Library books,books code,Amount,available quantity

public static void main(String [] args){

String Books [] = {"DPCO ","DM ","OOPS ","DS ","FDS ","PYTHON ","C ","C++ ","TAMIL ","ENGLISH"};

int Books\_Code [] = {1000,1001,1002,1003,1004,1005,1006,1007,1008,1009};

int Amount [] = {200,250,750,300,400,200,125,450,300,200};

int[] Quantity = {50,15,100,25,10,100,75,25,40,30};

int CurrentQuantity [] = new int [10];

//Updating Current quantity to the available quantity

for (int Current = 0; Current < CurrentQuantity.length; Current++) {

CurrentQuantity[Current] = Quantity[Current];

}

//Invoice arrays

String InvoiceBooks [] = new String[10];

int InvoiceBooks\_Code [] = new int[10];

int InvoiceAmount [] = new int[10];

int InvoiceQuantity [] = new int[10];

int InvoiceCost [] = new int[10];

//Variable declaration

int book\_position;

int RequiredBooks,BookCode,GrandTotal = 0, count = 0;

char ch;

//Printing the book name, book code, Price, Quantity

Scanner scanner = new Scanner(System.in);

System.out.println("List of items:");

System.out.println("Book Code\tBook Name\t\tPrice\t\tQuantity");

for(int i = 0;i < Books.length;i++){

System.out.println(Books\_Code[i]+"\t\t"+Books[i]+"\t\t\t"+Amount[i]+"\t\t"+Quantity[i]);

}

try {

do {

System.out.println("Enter the Book code:");

BookCode = scanner.nextInt();

boolean flag = false;

//Checking whether the given book code is valid or not

for (int i = 0; i < Books\_Code.length; i++) {

if (Books\_Code[i] == BookCode) {

flag = true;

//If that book is available then ask for required quantity

System.out.println("Enter the Required Quantity:");

RequiredBooks = scanner.nextInt();

if (RequiredBooks < 0) {

throw new InvalidQuantityException("The quantity must be greater than zero.");

} else if (RequiredBooks > CurrentQuantity[i]) {

throw new InvalidQuantityException("The requested quantity is not available.");

} else {

//Making invoice

InvoiceBooks[i] = Books[i];

InvoiceBooks\_Code[i] = Books\_Code[i];

InvoiceQuantity[i] = RequiredBooks;

InvoiceAmount[i] = Amount[i];

InvoiceCost[i] = RequiredBooks \* Amount[i];

// Then decrement the available quantity

CurrentQuantity[i] -= RequiredBooks;

count++;

break;

}

}

}

if (!flag) {

throw new InvalidBookCodeException("Invalid book code has been entered.");

}

System.out.println("Do you need any other books?\n(Press y or Y for yes/ Anyother key for no)");

ch = scanner.next().charAt(0);

} while (ch == 'y' || ch == 'Y');

System.out.println("\t\t\t\t\tINVOICE\t\t\t\t");

System.out.println("\t\t\t\tLibrary Management System\t\t\t");

System.out.println("S.No \tBook Code \t\tBook Name \tQuantity \tAmount \tCost");

//Calculating Grand Total

for (int i = 0; i < InvoiceCost.length; i++) {

GrandTotal += InvoiceCost[i];

}

//Printing Invoice...

for (int i = 0; i < InvoiceBooks.length; i++) {

if (InvoiceBooks[i] != null) {

System.out.println((++count) + "\t" + InvoiceBooks\_Code[i] + "\t\t\t" + InvoiceBooks[i] + "\t\t" + InvoiceQuantity[i] + "\t\t" + InvoiceAmount[i] + "\t" + InvoiceCost[i]);

}

}

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

System.out.println("Grand Total: " + GrandTotal);

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

System.out.println("Thanks for purchasing our Books!!!\nVisit Again\n");

count = 0;

//Writing this invoice to invoice.txt

try (BufferedWriter writer = new BufferedWriter(new FileWriter("invoice.txt"))) {

writer.write("\t\t\t\t\t\tINVOICE\t\t\t\t\n");

writer.write("\t\t\t\t\tLibrary Management System\t\t\t");

writer.write("\nS.No \tBook Code \tBook Name \tQuantity \t Amount \t Cost\n");

for (int i = 0; i < InvoiceBooks.length; i++) {

if (InvoiceBooks[i] != null) {

// Writing to the file

writer.write((++count) + "\t" + InvoiceBooks\_Code[i] + "\t\t" + InvoiceBooks[i] + "\t\t" + InvoiceQuantity[i] + "\t\t " + InvoiceAmount[i] + "\t\t " + InvoiceCost[i]);

writer.newLine();

}

}

writer.write("-----------------------------------------------------------------------------------------");

writer.newLine();

writer.write("Grand Total: " + GrandTotal);

writer.newLine();

writer.write("-----------------------------------------------------------------------------------------");

writer.newLine();

writer.write("Thanks for purchasing our Books!!!\nVisit Again\n");

} catch (IOException e) {

e.printStackTrace();

}

}

catch(InvalidQuantityException invalidQuantityException){

System.out.println(invalidQuantityException);

}

catch(InvalidBookCodeException invalidBookCodeException){

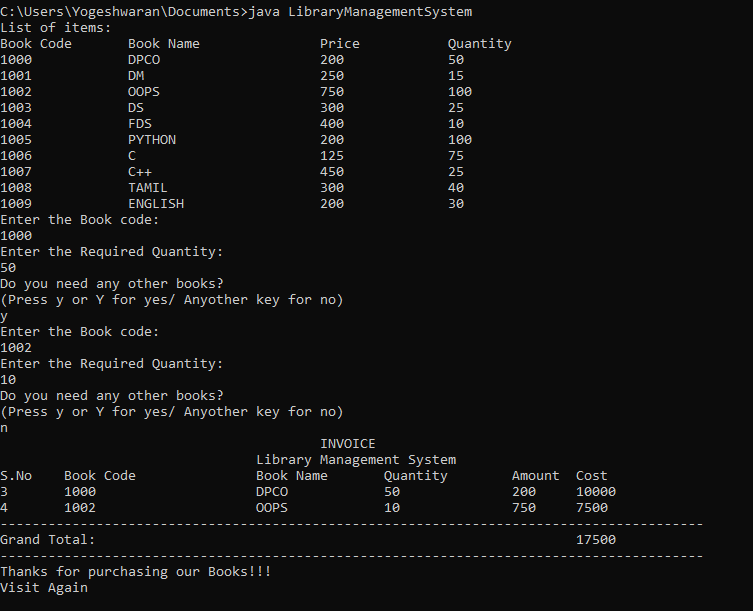
System.out.println(invalidBookCodeException);

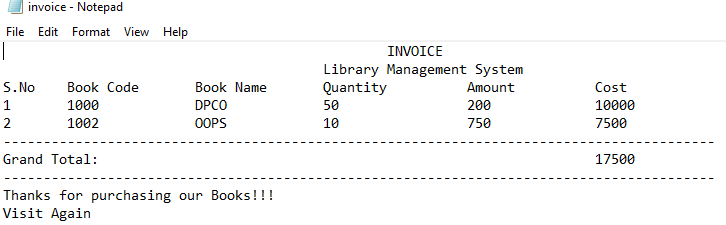
}

}

}

**Output:**

****



**Result:**

Thus, the mini project (Library Management) using Java was created successfully and output has been verified.