TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO CUỐI KỲ**

**MÔN LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

**Bookstore Management**

*Người hướng dẫn*: **TS. HỒ THỊ THANH TUYẾN**

*Người thực hiện*: **NGUYỄN TRIỀU DƯƠNG – 520H0621**

Lớp **: 20H50301**

Khoá  **: 24**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2021**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO CUỐI KỲ**

**MÔN LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

**Bookstore Management**

*Người hướng dẫn:* **TS. HỒ THỊ THANH TUYẾN**

*Người thực hiện:* **NGUYỄN TRIỀU DƯƠNG – 520H0621**

Lớp : **20H50301**

Khoá  **: 24**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2021**

LỜI CẢM ƠN

In the last semester, I learned Object-Oriented Programming with the dedicated instruction of Ms. Ho Thi Thanh Tuyen and Mr. Pham Thai Ky Trung, which was a solid foundation for me to complete this final report. Thanks to my teachers, I have acquired valuable knowledge of this subject and mastered the important concepts of classes, objects, and four important components of OOP: Inheritance, Encapsulation, Polymorphism, and Abstraction. Although I only studied with teachers for a short semester and had little contact with teachers because of the Covid-19 epidemic, the hours I spent in class with teachers opened my eyes and absorbed a lot of interesting knowledge. In the process of studying, I still faced many difficulties, but thanks to the dedicated teaching of my teachers, I overcame all of them, and I also gained valuable test-taking experiences to complete the subject well as well as myself, this final report.

To be able to study under the teaching of such a teacher, it is impossible not to mention the merits of the Board of Directors of Ton Duc Thang University, the teachers of the Department of Information Technology for giving me the opportunity. this. I sincerely thank the teachers.

Since this is the first time I have made a report, it is inevitable that there will be shortcomings, and I hope for the advice of the teachers of the Department of Information Technology in general and Ms. Ho Thi Thanh Tuyen and Mr. Pham Thai Ky Trung in particular that I can learn from it for the next time. Thank you.

**ĐỒ ÁN ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là sản phẩm đồ án của riêng tôi và được sự hướng dẫn của TS.Hồ Thị Thanh Tuyến ;. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình.** Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày 24 tháng 7 năm 2021*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Nguyễn Triều Dương*

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

**Phần xác nhận của GV hướng dẫn**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

**Phần đánh giá của GV chấm bài**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

TÓM TẮT

This is the final report of Object Oriented Programming. In part 1 we will see the very basic knowledge of programming with Java language such as naming rules for variables, primitive data types,... In part 2 is a small application of programming object-oriented through the design of a bookstore management system. We will meet again the very important knowledge of last semester such as Inheritance, Encapsulation Polymorphism and Abstraction and we will also see ArrayList again. The report is an opportunity to review as well as apply the knowledge learned in the last semester.

MỤC LỤC

[LỜI CẢM ƠN i](#_Toc78042128)

[PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN iii](#_Toc78042129)

[TÓM TẮT iv](#_Toc78042130)

[MỤC LỤC 1](#_Toc78042131)

[CHƯƠNG 1 – BÀI LÀM PHẦN 1 2](#_Toc78042132)

[1.1 Câu 1 2](#_Toc78042133)

[1.2 Câu 2 3](#_Toc78042134)

[CHƯƠNG 2 – BÀI LÀM PHẦN 2 6](#_Toc78042135)

[2.1 Câu 1 6](#_Toc78042136)

[2.2 Câu 2 10](#_Toc78042137)

[2.3 Câu 3 11](#_Toc78042139)

[2.4 Câu 4 19](#_Toc78042143)

[2.5 Câu 5 19](#_Toc78042144)

CHƯƠNG 1 – BÀI LÀM PHẦN 1

* 1. Câu 1

**a**. Rule of naming variable in Java:

- Variable names should started with a lowercase character.

- Variable names should not started with special character

- Variable names must not be same as Java’s keywords

- With variable name has many words, the first word should start with a lowercase character, all the following word started with uppercase character.

- If variable is a constant variable, variable name should uppercase all character of this name and separated word by \_ (underscore)

*Ex: book, name, bookName, …*

**b**. Describe 3 of all the primitive types in Java:

-int

-float

-boolean

**c**.

*import* *java.util.\**;

*//Define a Caculator class*

*public* *class* Calculator {

*//Define a method to check n is prime or not*

*public* *static* *boolean* checkPrime(*int* *n*) {

*boolean* result *=* true;

*if* ( n *<=* 1)

*return* false;

*for*( *int* i *=* 2; i *<=* n*/*2; i*++*)

*if* (n *%* i *==* 0) {

                result *=* false;

*break*;

            }

*return* result;

    }

*//Define a method to sum all prime number less than n*

*public* *static* *int* calSumPrime(*int* *n*) {

*int* sum *=* 0;

*for* ( *int* i*=*2; i *<* n; i*++*)

*if*(checkPrime(i)) sum *+=* i  ;

*return* sum;

    }

*//Main method to verify those methods*

*public* *static* *void* main(*String*[] *args*) {

*Scanner* sc *=* *new* Scanner(System.in);

        System.out.print("nhap vao mot so: ");

*int* n *=* sc.nextInt();

        System.out.println(checkPrime(n));

        System.out.println(calSumPrime(n));

    }

}

* 1. Câu 2

**a.**

*//Define a Student class*

*public* *class* Student {

*//3 properties*

*private* *String* name;

*private* *String* major;

*private* *int* age;

*//Constructor 1*

*public* Student(){};

*//Constructor 2*

*public* Student(*String* *name*, *String* *major*, *int* *age*){

*this*.name*=* name;

*this*.major *=* major;

*this*.age *=* age;

    };

*//getters and setters*

*public* *String* getName(){

*return* *this*.name;

    }

*public* *String* getMajor(){

*return*  *this*.major;

    }

*public* *int* getAge(){

*return*  *this*.age;

    }

*public* *void* setName(*String* *name*){

*this*.name *=* name;

    }

*public* *void* setMajor(*String* *major*){

*this*.major *=* major;

    }

*public* *void* setAge(*int* *age*){

*this*.age *=* age;

    }

}

**b.**

*//Define a Test class*

*public* *class* Test {

*//main method*

*public* *static* *void* main(*String*[] *args*) {

*Student* me *=* *new* Student("Nguyen Trieu Duong","Computer Science", 19);

*Student* someone *=* *new* Student();

        someone.setName("Anonymous");

        someone.setMajor("Unknown");

        someone.setAge(20);

        System.out.println(me.getName());

        System.out.println(me.getMajor());

        System.out.println(me.getAge());

        System.out.println("");

        System.out.println(someone.getName());

        System.out.println(someone.getMajor());

        System.out.println(someone.getAge());

    }

}

CHƯƠNG 2 – BÀI LÀM PHẦN 2

* 1. Câu 1

*My design has 3 class.*

**class Book:**

-Properties:

* float originalCost: the original price of book.
* String bookName: name of book.
* int numberSale: the number of books that were sold.

-Method:

* Constructor:
* public Book(): the constructor with no parameter.
* public Book(String bookName, float originalCost, int numberSale): the constructor that has some parameters is the name of book, its price how many books were sold.
* Processor:
* abstract float calSellingPrice(): an abstract method to calculate the book’s selling price.
* public float calBookIncome(): a method to calculate total money that earned by selling a book.( book’s income = numberSale \*(selling price – originalCost))
* public int calBookInStock(): a method to calculate how many book is in stock.

**class LiteraryBook:**

-Properties:

* String bookLanguage: the language that book written in.
* Boolean isNewBook(): is this book is a new product
* String author: author of book.
* int releaseYear: the year that book published.

-Method:

* Constructor:
* public LiteraryBook(): a constructor with no parameter.
* public LiteraryBook(String bookName, String author, int numberSale, int releaseYear, float originalCost, String bookLanguage, String bookGenre): the constructor that has some parameters is the name of book, its author, its price, when was this book published, how many books were sold and language the book written in and its status.
* Processor:
* float calSellingPrice(): method to calculate the book selling price ( book’s selling price = originalCost + orginalCost\*5% .
* float calPriceForForeignBook(): method to calculate price for book from other country.
* float calPriceByBookStatus(): method to calculate price for old book

**class Magazine:**

-Properties:

* String datePublished: the date that the book was published
* String publisher: the publisher of the book
* boolean isSpecial: is this magazine published in a special day.
* i-Method:
* Constructor:
* public Magazine(): the constructor with no parameter.
* public Magazine(String bookName, int numberSale, float originalCost, String publisher, String datePublished, String isSpecial): the constructor that has some parameters is the name of book, its price, how many books were sold, the publisher, the published date and the type of a magazine.
* Processor:
* float calSellingPrice(): method to calculate the book selling price ( book’s selling price = originalCost + orginalCost\*10%) .
* boolean isAvailable(): method to check if a magazine is still available or not
* float calPriceForSpecialMagazine: method to calculate price for magazine published in a special day.

*Inheritance:*

Two subclasses are LiteraryBook and Magazine has some properties and method of its super class is Book( classes LiteraryBook and Magazine extended from Book )

* Inherited properties: bookName, author, releaseYear, originalCost, numbersale.
* Inherited method: calSellingPrice

*Encapsulation:*

- Not only has inherited elements, each class has its own properties and method, some of them are private to prevent access from the outside, all of them are encapsulated in a class.

* LiteraryBook has some private properties are bookLanguage and isNewBook that mean other class can not use these properties. LiteraryBook also has it own method is calPriceForForeignBook and calPriceByBookStatus.
* Magazine has its own private properties are datePublished, publisher, isSpecial and its own method isAvailable and calPriceForSpecialMagazine.

*Polymorphism:*

- Overloading: the class Book has two constructors with the same name Book but different parameter, one with no parameter. Following that, two subclass have the same thing is LiteraryBook constructor and Magazine constructor.

- Overriding: all the three class, superclass Book and subclasses LiteraryBook and Magazine have toString method, but they have different body part, we can see easily they have different return value.

* class Book’s toString method has return value is

return "Book:["+ bookName +","+ originalCost +","+ numberSale +","+"]";

* class LiteraryBook’s toString method has return value is

return"LiteraryBook:["+bookName+","+author+","+releaseYear+","+bookLanguage+","+originalCost+","+numberSale+","+isNewBook+"]";

* class Magazine’s toString method has return value i return"Magazine:["+bookName+","+publisher+","+datePublished+","+numberSale+","+originalCost+","+isSpecial+"]";

This things also happen to the calSellingPrice method in 3 class.

* class Book’s calSellingPrice method has no return value
* class LiteraryBook’s calSellingPrice method has return value is

return (float)Math.ceil(this.originalCost + 0.05\*this.originalCost);

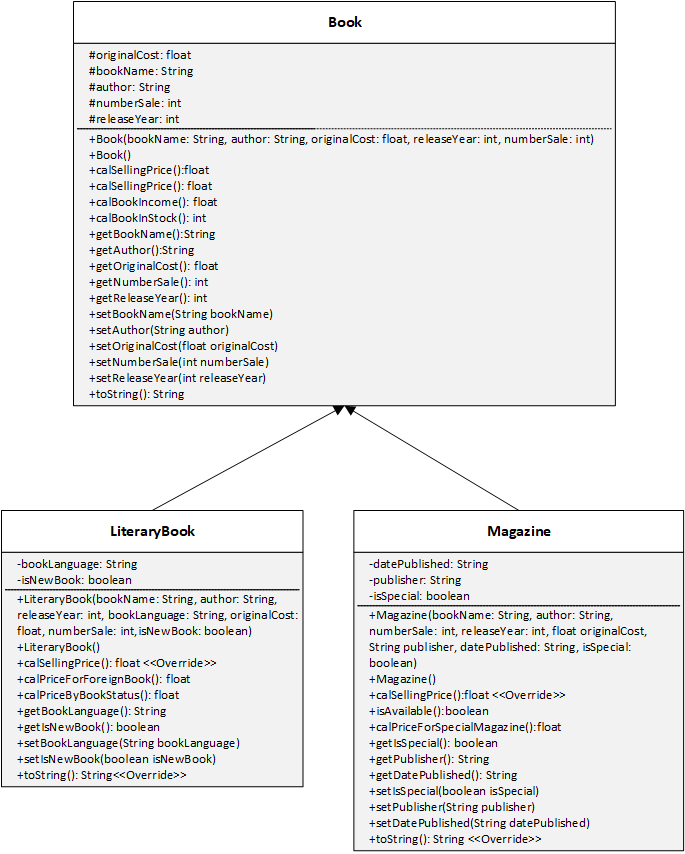
* While class Magazine’s calSellingPrice has return value is

return (float)(this.originalCost + 0.1\*this.originalCost);

*Abstraction:* In class Book calSellingPrice is an abstract method

* 1. Câu 2

UML Diagram



* 1. Câu 3

*class Book:*

*import* *java.util.\**;

*abstract* *class* Book {

*//Original cost property*

*protected* *float* originalCost;

*//some random properties*

*protected* *String* bookName;

*protected* *int* numberSale;

*//two random constructors*

*public* Book(*String* *bookName*, *float* *originalCost*, *int* *numberSale*) {

*this*.bookName *=* bookName;

*this*.originalCost *=* originalCost;

*this*.numberSale *=* numberSale;

    }

*public* Book() {};

*//abstract method*

*abstract* *float* calSellingPrice();

*//2 process method*

*public* *float* calBookIncome(){

*return* (calSellingPrice()*-*originalCost)*\**numberSale;

    }

*public* *int* calBookInStock() {

        System.out.print("How many book was imported? ");

*Scanner* sc *=* *new* Scanner(System.in);

*int* numberImport *=* sc.nextInt();

        sc.close();

*return* numberImport *-* *this*.numberSale;

    }

*//getters,setters and toString*

*public* *String* getBookName(){

*return* *this*.bookName;

    }

*public* *float* getOriginalCost(){

*return* *this*.originalCost;

    }

*public* *int* getNumberSale(){

*return* *this*.numberSale;

    }

*public* *void* setBookName(*String* *bookName*){

*this*.bookName *=* bookName;

    }

*public* *void* setOriginalCost(*float* *originalCost*){

*this*.originalCost *=* originalCost;

    }

*public* *void* setNumberSale(*int* *numberSale*){

*this*.numberSale *=* numberSale;

    }

*public* *String* toString(){

*return* "Book:["*+* bookName *+*","*+* originalCost *+*","*+* numberSale *+*","*+*"]";

    }

}

*class LiteraryBook:*

*public* *class* LiteraryBook *extends* Book {

*private* *String* bookLanguage;

*private* *boolean* isNewBook;

*private* *String* author;

*private* *int* releaseYear;

*//2 constructors*

*public* LiteraryBook(*String* *bookName*, *String* *author*, *int* *releaseYear*, *String* *bookLanguage*, *float* *originalCost*, *int* *numberSale*, *boolean* *isNewBook*){

*super*(bookName, originalCost, numberSale);

*this*.bookLanguage *=* bookLanguage;

*this*.isNewBook *=* isNewBook;

*this*.author *=* author;

*this*.releaseYear *=* releaseYear;

    }

*public* LiteraryBook() {};

    @*Override*

*float* calSellingPrice() {

*return* (*float*)Math.ceil(*this*.originalCost *+* 0.05*\*this*.originalCost);

    }

*public* *float* calPriceForForeignBook(){

*double* importTax*=*0;

*if*(*this*.bookLanguage *!=* "vietnam") importTax *=* 0.1;

*return* (*float*)(calSellingPrice() *+* importTax*\**calSellingPrice());

    }

*public* *float* calPriceByBookStatus() {

*if*(isNewBook *==* true) *return* calSellingPrice();

*else* *return* (*float*)(0.5*\**calSellingPrice());

    }

*//getters, setters and toString*

*public* *String* getBookLanguage(){

*return* *this*.bookLanguage;

    }

*public* *boolean* getBookStatus(){

*return* *this*.isNewBook;

    }

*public* *String* getAuthor(){

*return* *this*.author;

    }

*public* *int* getReleaseYear(){

*return* *this*.releaseYear;

    }

*public* *void* setBookLanguage(*String* *bookLanguage*){

*this*.bookLanguage *=* bookLanguage;

    }

*public* *void* setBookStatus(*boolean* *isNewBook*){

*this*.isNewBook *=* isNewBook;

    }

*public* *void* setAuthor(*String* *author*){

*this*.author *=* author;

    }

*public* *void* setReleaseYear(*int* *releaseYear*){

*this*.releaseYear *=* releaseYear;

    }

    @*Override*

*public* *String* toString() {

*return* "LiteraryBook:["*+*bookName*+*","*+*author*+*","*+*releaseYear*+*","*+*bookLanguage*+*","*+*originalCost*+*","*+*numberSale*+*","*+*isNewBook*+*"]";

    }

}

*class Magazine:*

*public* *class* Magazine *extends* Book {

*private* *String* datePublished;

*private* *String* publisher;

*private* *boolean* isSpecial;

*//2 constructors*

*public* Magazine() {};

*public* Magazine(*String* *bookName*, *String* *publisher*, *String* *datePublished*, *int* *numberSale*, *float* *originalCost*, *boolean* *isSpecial*) {

*super*(bookName, originalCost, numberSale);

*this*.publisher *=* publisher;

*this*.datePublished *=* datePublished;

*this*.isSpecial *=* isSpecial;

    }

    @*Override*

*float* calSellingPrice(){

*return* (*float*)(*this*.originalCost *+* 0.1*\*this*.originalCost);

    }

*//2 process constructor*

*public* *boolean* isAvailable(){

*if* (calBookInStock() *>* 0) *return* true;

*else* *return* false;

    }

*public* *float* calPriceForSpecialMagazine() {

*double* tax;

*if* ( *this*.isSpecial *==* true)

            tax *=* 0.2;

*else* tax *=* 0;

*return* (*float*)(calSellingPrice() *+* tax*\*this*.originalCost);

    }

*//getters, setters and toString*

*public* *boolean* getIsSpecial(){

*return* *this*.isSpecial;

    }

*public* *String* getPublisher(){

*return* *this*.publisher;

    }

*public* *String* getDatePublished(){

*return* *this*.datePublished;

    }

*public* *void* setIsSpecial(*boolean* *isSpecial*){

*this*.isSpecial *=* isSpecial;

    }

*public* *void* setPublisher(*String* *publisher*){

*this*.publisher *=* publisher;

    }

*public* *void* setDatePublished(*String* *datePublished*){

*this*.datePublished *=* datePublished;

    }

    @*Override*

*public* *String* toString(){

*return* "Magazine:["*+*bookName*+*","*+*publisher*+*","*+*datePublished*+*","*+*numberSale*+*","*+*originalCost*+*","*+*isSpecial*+*"]";

    }

}

* 1. Câu 4

import java.util.\*;

public class ManageBook {

//List of books

ArrayList<Book> bookList = new ArrayList<Book>();

//constructor with a list of book as its argument

ManageBook(ArrayList<Book> bookList){

this.bookList = bookList;

}

//choose 1 of 3 method

public Book findMaxSellingPrice(){

int index = 0;

float max = bookList.get(0).calSellingPrice();

for(int i=0; i<bookList.size(); i++) {

if (bookList.get(i).calSellingPrice() > max){

max = bookList.get(i).calSellingPrice();

index = i;

}

}

return bookList.get(index);

}

}

* 1. Câu 5

import java.util.\*;

public class BookstoreManagement {

public static void main(String[] args) {

//construct a literary book and a magazine, call all methods for each

LiteraryBook selfhelp = new LiteraryBook("How to win friend and influence people","Dale Carnegie",1936,"english",155000,1000,false);

LiteraryBook matbiec = new LiteraryBook("Mat Biec","Nguyen Nhat Anh",2000,"vietnam",50000,2000,false);

LiteraryBook minimal = new LiteraryBook("Nghi don gian song don thuan","ABC",2019,"vietnam",70000,1000,true);

Magazine glamour = new Magazine("Glamour","XYZ","31/2",500, 200000, false);

Magazine IT = new Magazine("IT la vua cua cac nghe","thang em 2k2 KHMT-TDT","14/03",2021,500000,true);

Magazine phunu = new Magazine("Phu Nu","NXB Phu nu","23/7",1000,75000,false);

System.out.println(selfhelp.toString());

System.out.println(selfhelp.getBookName());

System.out.println(selfhelp.getAuthor());

System.out.println(selfhelp.getReleaseYear());

System.out.println(selfhelp.getBookLanguage());

System.out.println(selfhelp.getOriginalCost());

System.out.println(selfhelp.getNumberSale());

System.out.println(selfhelp.getBookStatus());

System.out.println("selling price start at: "+selfhelp.calSellingPrice());

System.out.println("if book from foreign country then price: "+selfhelp.calPriceForForeignBook());

System.out.println("price by book status, if old sale 50%: "+selfhelp.calPriceByBookStatus());

System.out.println("total income of this book: "+selfhelp.calBookIncome());

System.out.println(phunu.toString());

System.out.println(phunu.getBookName());

System.out.println(phunu.getPublisher());

System.out.println(phunu.getDatePublished());

System.out.println(phunu.getNumberSale());

System.out.println(phunu.getOriginalCost());

System.out.println(phunu.getIsSpecial());

System.out.println("selling price start at: "+phunu.calSellingPrice());

System.out.println("Does this book still available: "+phunu.isAvailable());

System.out.println("price if this is a special magazine, if not price will remain: "+phunu.calPriceForSpecialMagazine());

System.out.println("total income of this book: "+phunu.calBookIncome());

//construct book management object, pass 6 book, call method

ArrayList<Book> bookLists = new ArrayList<Book>();

bookLists.add(selfhelp);

bookLists.add(matbiec);

bookLists.add(minimal);

bookLists.add(phunu);

bookLists.add(glamour);

bookLists.add(IT);

ManageBook list = new ManageBook(bookLists);

System.out.print("Book has highest price is: "+list.findMaxSellingPrice());

}

}