**TRƯỜNG ĐẠI HỌC GIA ĐỊNH**

**KHOA CÔNG NGHỆ THÔNG TIN**

.............................🙧🕮🙥…………………..

****

**BÀI BÁO CÁO CUỐI KÌ**

**MÔN**

**LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

**Đề tài: Phần mềm quản lý hóa đơn của công ty lắp ráp và kinh doanh laptop**

**TÊN GIẢNG VIÊN: Trần Hoài Thịnh**

**LỚP: 221402**

**NGÀNH: Kỹ Thuật Phần Mềm**

**THÀNH VIÊN VÀ MSSV:**

**Nguyễn Văn Khánh - 22150129**

**Lê Lâm Chiến Thắng - 22140345**

**Nguyễn Trần Hoàng Thịnh - 22140341**

**Nguyễn Văn Sang - 22140347**

**Phạm Nhật Quang - 22140462**

*TP. HỒ CHÍ MINH,THÁNG 07, NĂM 2023*

MỤC LỤC

[I. Phát biểu bài toán 3](#_Toc139417081)

[II. Xác định các lớp của bài toán 4](#_Toc139417082)

[1. Class Laptop 4](#_Toc139417083)

[2. Class CPU kế thừa từ class Laptop 4](#_Toc139417084)

[3. Class RAM kế thừa class Laptop 4](#_Toc139417085)

[4. Class Screen kế thừa class Laptop 5](#_Toc139417086)

[5. Class HardDisk kế thừa class Laptop 5](#_Toc139417087)

[6. Class Language 5](#_Toc139417088)

[7. Class Lib 5](#_Toc139417089)

[8. Class Quanly 7](#_Toc139417090)

[III. Mô tả thuật toán thao tác 10](#_Toc139417091)

[1. Thêm thông tin Laptop mới. 10](#_Toc139417092)

[2. Sửa thông tin Laptop. 10](#_Toc139417093)

[3. Xem thông tin Laptop. 10](#_Toc139417094)

[4. Xóa thông tin Laptop. 10](#_Toc139417095)

[IV. Cài đặt bài toán 11](#_Toc139417096)

[1. Class Laptop: 11](#_Toc139417098)

[2. Class CPU: 14](#_Toc139417243)

[3. Class RAM: 18](#_Toc139417399)

[4. Class Screen: 23](#_Toc139417565)

[5. Class HardDisk: 27](#_Toc139417566)

[6. Class Quanly: 31](#_Toc139417567)

[7. Class Lib 44](#_Toc139417568)

[8. Class Language 57](#_Toc139418065)

[9. Class main 59](#_Toc139418133)

[V. Kiểm thử lập bộ test 61](#_Toc139418197)

[1. Nhập danh sách Laptop 62](#_Toc139418198)

[2. Xem thông tin tất cả Laptop 66](#_Toc139418199)

[3. Sửa thông tin Laptop 67](#_Toc139418200)

[4. Xóa thông tin Laptop 68](#_Toc139418201)

[5. Xuất hóa đơn 69](#_Toc139418202)

[6. Xuất dữ liệu Laptop ra file 70](#_Toc139418204)

[7. Nhập dữ liệu Laptop ra file 71](#_Toc139418206)

# Phát biểu bài toán

Hiện nay xu hướng kinh doanh các mặt hàng thời trang ngày càng phát triển và đa dạng. Từ đó, Việc quản lý hóa đơn cho cửa hàng laptop là công việc quản lý và tổ chức các hoạt động của cửa hàng một cách hợp lý và hiệu quả nhất. Với nhu cầu Laptop ngày càng cao, việc quản lý và lên hóa đơn cho các sản phẩm tại cửa hàng là điều cần thiết để giúp cửa hàng tính toán số tiền, nhập thông tin được chính xác nhất và hiệu quả nhất. Nhằm đáp ứng nhu cầu trên. Nhóm chúng em đã chọn đề tài ‘Phần mềm quản lý hóa đơn của công ty lắp ráp và kinh doanh laptop”. Bài toán được phân tích xây dựng qua các nội dung sau đây.

* **Các đối tượng cần quản lý:**

**Các loại Laptop:** Butterfly, Fly, Bee

**Các loại CPU:** i3, i5, i7

**Các loại RAM:** DDR3, DDR4

**Các loại màn hình:** HD, FullHD

**Các loại ổ cứng:** HDD, SSD

# Xác định các lớp của bài toán

## Class Laptop

* + - Thuộc tính: String ma\_laptop, ten\_laptop, int dong\_laptop, String[] laptop\_arr, ArrayList<CPU>, <RAM>, <HardDisk>, <Screen>.
    - phương thức:
* Constructor: Hàm tạo có đối số và hàm tạo không đối số.
* Getter, Setter: Phương thức get, set cho các thuộc tính kiểu private.
* nhapTTLaptop(): Nhập thông tin cơ bản Laptop (Tên, mã, dòng).
* nhapMaLaptop(): Nhập vào mã Laptop.
* nhapTenLaptop(): Nhập vào tên Laptop.
* nhapDongLaptop(): Nhập vào dòng Laptop.
* reset\_dsCPU(): Khởi tạo lại danh sách CPU.
* reset\_dsRAM(): Khởi tạo lại danh sách RAM.
* reset\_dsScreen(): Khởi tạo lại danh sách màn hình.
* reset\_dsHardDisk(): Khởi tạo lại danh sách ổ cứng.
* reset\_dsALL(): Khởi tạo lại toàn bộ danh sách CPU, RAM, màn hình và ổ cứng.
* getTong\_gia(): Trả về tổng giá tiền của CPU, RAM, màn hình và ổ cứng.

## Class CPU kế thừa từ class Laptop

* + - Thuộc tính: String ma\_cpu, loai\_cpu, hang\_cpu; int don\_gia; String[] cpu\_arr; cpu\_price\_arr.
    - Phương thức:
* nhapTTCPU(): Nhập thông tin CPU.
* export\_config(): Xuất thông tin CPU dưới dạng Config.
* Constructor: Hàm tạo có đối số và hàm tạo không đối số.
* Getter, Setter: Phương thức get, set cho các thuộc tính kiểu private.
* toString(): Override toString() ở java.lang.Object, hiện thông tin CPU.

## Class RAM kế thừa class Laptop

* + - Thuộc tính: int so\_luong, dung\_luong; String[] ma\_ram, ram\_arr, ram\_price\_arr; String loai\_ram; long don\_gia.
    - Phương thức:
* nhapTTRam(): Nhập thông tin RAM.
* export\_config(): Xuất thông tin RAM dưới dạng Config.
* Constructor: Hàm tạo có đối số và hàm tạo không đối số.
* Getter, Setter: Phương thức get, set cho các thuộc tính kiểu private.
* toString(): Override toString() ở java.lang.Object, hiện thông tin RAM.

## Class Screen kế thừa class Laptop

* + - Thuộc tính: String ma\_manhinh, hang\_sanxuat, do\_phangiai; long don\_gia; int chong\_loa; String[] screen\_arr, screen\_fullhd\_arr, screen\_price\_arr.
    - Phương thức:
* nhapTTScreen(): Nhập thông tin màn hình.
* export\_config(): Xuất thông tin Màn hình dưới dạng Config.
* Constructor: Hàm tạo có đối số và hàm tạo không đối số.
* Getter, Setter: Phương thức get, set cho các thuộc tính kiểu private.
* toString(): Override toString() ở java.lang.Object, hiện thông tin Màn hình.

## Class HardDisk kế thừa class Laptop

* + - Thuộc tính: String kieu\_ocung, ma\_ocung; long don\_gia; int dung\_luong, so\_vong\_quay, toc\_do\_doc, toc\_do\_ghi; String[] harddisk\_arr, harddisk\_price
    - Phương thức:
* nhapTTHardDisk(): Nhập thông tin ổ cứng.
* export\_config(): Xuất thông tin ổ cứng dưới dạng Config.
* Constructor: Hàm tạo có đối số và hàm tạo không đối số.
* Getter, Setter: Phương thức get, set cho các thuộc tính kiểu private.
* toString(): Override toString() ở java.lang.Object, hiện thông tin ổ cứng.

## Class Language

* + - Thuộc tính: String lang\_path, current\_lang, current\_lang\_path; int count; String[] all\_language, all\_path\_lang.
    - Phương thức:
      * Constructor: Hàm tạo không đối số.
      * Getter: Phương thức get cho các thuộc tính kiểu private.
      * set\_language(): Chọn ngôn ngữ theo danh sách ngôn ngữ.
      * load\_language(): Khởi tạo danh sách ngôn ngữ từ các file.

## Class Lib

* + - Thuộc tính: Language lang; int MAX\_DATA, loaded; String[] is\_load\_path; String FILE\_CONFIG; ArrayList<Properties>; FileInputStream inputStream; Scanner keyboard.
    - Phương thức:
      * Constructor: Hàm tạo có đối số và hàm tạo không đối số.
      * clear\_console(): Dọn dẹp màn hình Terminal.
      * pause\_console(): Tạm dừng chờ người dùng nhấn Enter.
      * str\_input(): Trả về một chuỗi ký tự được nhập từ bàn phím.
      * yes\_or\_no(): Trả về true hoặc false theo câu hỏi Yes or No.
      * int\_input(): Trả về một số theo kiểu int, long được nhập từ bàn phím.
      * kt\_ma\_laptop(): Trả về vị trí của mã Laptop trong Array nếu nó tồn tại ở trong danh sách kèm 0 hoặc 1 (tìm thấy và không tìm thấy).
      * printAll(): In danh sách Laptop hiện có ra màn hình Terminal.
      * gettext\_cpu(): Trả về một chuỗi ký tự từ toString() hoặc export\_config() từ class CPU.
      * gettext\_ram(): Trả về một chuỗi ký tự từ toString() hoặc export\_config() từ class RAM.
      * gettext\_screen(): Trả về một chuỗi ký tự từ toString() hoặc export\_config() từ class Screen.
      * gettext\_harddisk(): Trả về một chuỗi ký tự từ toString() hoặc export\_config() từ class HardDisk.
      * gettext(): Trả về một chuỗi ký tự từ gettext\_cpu(), gettext\_ram(), gettext\_screen(), gettext\_harddisk() theo từng Laptop trong dsLaptop.
      * is\_load(): Trả về true nếu file properties đã được load và ngược lại.
      * load\_language(): khởi tạo đối tượng từ class Language vào Lib.lang;
      * getlang(): Trả về một chuỗi ký tự theo mã của ngôn ngữ đã chọn.
      * read\_config(): trả về một dữ liệu dạng chuỗi ký tự ở file config.
      * read\_data(): Trả về một dãy ký tự từ file do người dùng nhập.
      * rmBlankArr(): trả về một mảng các chuỗi ký tự đã được xóa các chuỗi trống.
      * rmSpaceArr(): Trả về một mảng các chuỗi ký tự đã được xóa khoảng trống ở đầu và cuối mảng.
      * getIndexArr(): Trả về vị trí của chuỗi ký tự trong mảng do người dùng nhập.
      * rmSpace(): Trả về một chuỗi ký tự đã được xóa khoảng trống ở đầu và cuối chuỗi.
      * readDataLaptop(): Trả về dữ liệu của dòng Laptop đó tại file data.properties.
      * arrToString(): Trả về một chuỗi ký tự được chuyển từ các phần tử trong mảng.
      * arrToListString(): Trả về một chuỗi ký tự được chuyển từ phần từ trong mảng nhưng có thêm dấu “(“ ở đầu và dấu “)” ở cuối chuỗi, các phần tử được cách nhau bằng dấu “, “.
      * lsdir(): Trả về một chuỗi ký tự là các file ở tại một thư mục nào đó.
      * fwrite(): Ghi dữ liệu vào một file.
      * isNotEmpty(): Trả về true nếu danh sách có độ dài lớn hơn 0 và ngược lại.

## Class Quanly

* + - Thuộc tính:
      * String conf\_path

⮚ ArrayList<Laptop> dsLaptop;

* + - Phương thức:

⮚ Constructor: Hàm tạo không đối số.

⮚ themLaptop(): Nhập thông tin một hoặc nhiều Laptop mới.

⮚ menu\_edit(): In ra màn hình các lựa chọn để chỉnh sửa thông tin Laptop.

⮚ suaTTLaptop(): Sửa thông tin Laptop.

⮚ xoaTTLaptop(): Xóa theo mã Laptop hoặc xóa tất cả.

⮚ xemTTLaptop(): In ra màn hình thông tin của các Laptop trong danh sách.

⮚ xuatTT2File(): Xuất hóa đơn tất cả Laptop trong danh sách ra File.

⮚ nhapCONF(): Nhập dữ liệu các Laptop từ file .conf (không phải file hóa đơn).

⮚ xuatCONF(): Ghi dữ liệu ở dạng config từ danh sách Laptop vào File.

⮚ menu(): In ra màn hình các lựa chọn của menu chính.

|  |
| --- |
| Laptop |
| - String ma\_laptop, ten\_laptop  - int dong\_laptop  - String[] laptop\_arr  - ArrayList<CPU> dsCPU  - ArrayList<RAM> dsRAM  - ArrayList<HardDisk> dsHardDisk  - ArrayList<Screen> dsScreen  + Constructor()  + Setter(), Getter()  + nhapTTLaptop()  + nhapMaLaptop()  + nhapTenLaptop()  + nhapDongLaptop()  + reset\_dsCPU()  + reset\_dsRAM()  + reset\_dsScreen()  + reset\_dsHardDisk()  + reset\_dsALL()  + getTong\_gia |

|  |
| --- |
| CPU |
| - String ma\_cpu, loai\_cpu, hang\_cpu  - int don\_gia  - String[] cpu\_arr, cpu\_price\_arr + Constructor() + Setter(), Getter() + nhapTTCPU() + export\_config() + toString() |

|  |
| --- |
| Screen |
| - String ma\_manhinh, hang\_sanxuat, do\_phangiai  - long don\_gia  - int chong\_loa  - String[] screen\_arr, screen\_fullhd\_arr, screen\_price\_arr  + Constructor() + Setter(), Getter()  + nhapTTScreen()  + export\_config()  + toString() |

|  |
| --- |
| RAM |
| - int so\_luong. dung\_luong  - String[] ma\_ram, ram\_arr, ram\_price\_arr - String loai\_ram  - long don\_gia + Constructor() + Setter(), Getter() + nhapTTRam() + export\_config()  + toString() |

|  |
| --- |
| Lib |
| - Language lang  - int MAX\_DATA, loaded  - String[] is\_load\_path  - ArrayList<Properties> properties  - FileInputStream inputStream  - Scanner keyboard  + Constructor()  + clear\_console()  + pause\_console()  + str\_input()  + yes\_or\_no()  + int\_input()  + printAll()  + gettext\_cpu()  + gettext\_ram()  + gettext\_screen()  + gettext\_harddisk()  + gettext()  + is\_load()  + load\_language()  + getlang()  + read\_config()  + read\_data()  + rmBlankArr()  + rmSpaceArr()  + getIndexArr()  + rmSpace()  + readDataLaptop()  + arrToString()  + arrToListString()  + lsdir()  + fwrite()  + isNotEmpty() |

|  |
| --- |
| Language |
| - String lang\_path, current\_lang, current\_lang\_path  - int count  - String[] all\_language, all\_path\_lang  + Constructor()  + Getter()  + set\_language()  + load\_language() |

|  |
| --- |
| HardDisk |
| - String kieu\_ocung, ma\_ocung  - long don\_gia  - int dung\_luong, so\_vong\_quay, toc\_do\_doc, toc\_do\_ghi  - String[] harddisk\_arr, harddisk\_price + Constructor() + Setter(), Getter() + nhapTTHardDisk() + export\_config()  + toString() |

|  |
| --- |
| Quanly |
| - String conf\_path  - ArrayList<Laptop> dsLaptop + Constructor() + themLaptop() + menu\_edit()  + suaTTLaptop()  + xemTTLaptop()  + xuatTT2File()  + nhapCONF()  + xuatCONF()  + menu() |

# Mô tả thuật toán thao tác

## Thêm thông tin Laptop mới.

Thuật toán:

* + - Nhập vào Dòng Laptop, Mã Laptop, Tên Laptop.
    - Dùng vòng lặp và câu lệnh so sánh, nếu đã tồn tại Mã Laptop trong danh sách đã nhập thì sẽ yêu cầu nhập lại Mã Laptop.
    - Nhập vào thông tin CPU, RAM, Màn hình, Ổ cứng
    - Thêm: Nhập vào thông tin đối tượng mới.

## Sửa thông tin Laptop.

Thuật toán:

* + - Nhập mã Laptop muốn thay đổi.
    - Dùng vòng lặp và so sánh để kiểm tra nếu Mã Laptop đã tồn tại trong danh sách thì vào menu các lựa chọn thay đổi thông tin.
    - Sửa : Nhập lại thông tin bạn đã chọn.

## Xem thông tin Laptop.

Thuật toán:

* + - Dùng vòng lặp hiển thị ra màn hình từng Laptop trong danh sách.
    - Dùng vòng lặp hiển thị ra màn hình thông tin cụ thể của Laptop.

## Xóa thông tin Laptop.

Thuật toán:

* + - Nhập vào Mã Laptop muốn xóa.
    - Dùng vòng lặp tìm kiếm Mã Laptop đó trong danh sách, nếu tồn tại sẽ xóa nó.
    - Xóa: Tạo mới ArrayList và thêm các đối tượng ngoại trừ đối tượng muốn xóa, sau đó ghi đè lên ArrayList cũ.

# Cài đặt bài toán

# Lưu ý: Chương trình này cần folder data này để chạy được: https://tinyurl.com/BTL5-data-221402

## Class Laptop:

## import java.util.ArrayList;

## public class Laptop {

## private String ma\_laptop;

## private String ten\_laptop;

## private int dong\_laptop;

## private final String[] laptop\_arr = Lib.read\_data("laptop\_type");

## protected ArrayList<CPU> dsCPU = null;

## protected ArrayList<RAM> dsRAM = null;

## protected ArrayList<HardDisk> dsHardDisk = null;

## protected ArrayList<Screen> dsScreen = null;

## 

## public int getDong\_laptop() {

## return this.dong\_laptop;

## }

## 

## public void reset\_dsCPU() {

## this.dsCPU = new ArrayList<CPU>();

## }

## 

## public void reset\_dsRAM() {

## this.dsRAM = new ArrayList<RAM>();

## }

## 

## public void reset\_dsScreen() {

## this.dsScreen = new ArrayList<Screen>();

## }

## 

## public void reset\_dsHardDisk() {

## this.dsHardDisk = new ArrayList<HardDisk>();

## }

## 

## public void reset\_dsALL(){

## reset\_dsCPU();

## reset\_dsRAM();

## reset\_dsScreen();

## reset\_dsHardDisk();

## }

## 

## public String getnameDong\_laptop() {

## return this.laptop\_arr[this.dong\_laptop-1];

## }

## 

## public boolean setDong\_laptop(int dong\_laptop) {

## int count = 1;

## for(String s : this.laptop\_arr) {

## if(dong\_laptop == count) {

## this.dong\_laptop = dong\_laptop;

## return true;

## }

## count++;

## }

## return false;

## }

## 

## public void nhapMaLaptop(ArrayList<Laptop> dsLaptop) {

## String tmp;

## while(true) {

## System.out.print(Lib.getlang("LAPTOP\_CODE") + ": ");

## tmp = Lib.str\_input();

## int[] info = Lib.kt\_ma\_laptop(dsLaptop, tmp);

## if(info[0]==1){

## System.out.println("!! " + Lib.getlang("LAPTOP\_CODE\_EXIST"));

## continue;

## }

## this.ma\_laptop = tmp;

## break;

## }

## }

## 

## public void nhapTenLaptop() {

## System.out.print(Lib.getlang("LAPTOP\_NAME") +": ");

## this.ten\_laptop = Lib.str\_input();

## }

## 

## public void nhapDongLaptop() {

## int count;

## while(true) {

## count = 1;

## System.out.print(Lib.getlang("LAPTOP\_SERIES") +" (");

## for(String s : this.laptop\_arr) {

## if(count>1){

## System.out.print(", ");

## }

## System.out.print(String.valueOf(count)

## + " - "

## + Lib.rmSpace(s));

## count++;

## }

## System.out.print("): ");

## try {

## if (this.setDong\_laptop(Lib.keyboard.nextInt())) {

## break;

## }

## } catch (Exception e) {

## System.out.println("!! " + Lib.getlang("ENTER\_NUM\_PLZ"));

## }

## }

## }

## 

## public void nhapTTLaptop(ArrayList<Laptop> dsLaptop) {

## nhapDongLaptop();

## nhapMaLaptop(dsLaptop);

## nhapTenLaptop();

## }

## 

## public Laptop(){

## this.ma\_laptop = "";

## this.ten\_laptop = "";

## reset\_dsALL();

## }

## 

## public String getMa\_laptop() {

## return this.ma\_laptop;

## }

## 

## public void setMa\_laptop(String ma\_laptop) {

## this.ma\_laptop = ma\_laptop;

## }

## 

## public String getTen\_laptop() {

## return this.ten\_laptop;

## }

## 

## public void setTen\_laptop(String ten\_laptop) {

## this.ten\_laptop = ten\_laptop;

## }

## 

## public long getTong\_gia() {

## long tong\_gia = 0;

## for(CPU i : dsCPU) {

## tong\_gia += i.getDon\_gia();

## }

## for(RAM i : dsRAM) {

## tong\_gia += i.getDon\_gia();

## }

## for(Screen i : dsScreen) {

## tong\_gia += i.getDon\_gia();

## }

## for(HardDisk i : dsHardDisk) {

## tong\_gia += i.getDon\_gia();

## }

## return tong\_gia;

## }

## }

## Class CPU:

## import java.util.Arrays;

## 

## public class CPU extends Laptop {

## private String ma\_cpu; // CPU1, CPU2, CPU3

## private String loai\_cpu; // i3, i5, i7

## private long don\_gia; // 100, 200, 300

## private String hang\_cpu = Lib.arrToString(Lib.read\_data("cpu\_company"));

## private final String[] cpu\_arr = Lib.read\_data("cpu\_type");

## private final String[] cpu\_price\_arr = Lib.read\_data("cpu\_price");

## 

## public CPU(){

## this.ma\_cpu = "";

## this.loai\_cpu = "";

## this.don\_gia = 0;

## }

## 

## public CPU(String ma\_cpu, String loai\_cpu, String hang\_cpu, long don\_gia) {

## this.ma\_cpu = ma\_cpu;

## this.hang\_cpu = hang\_cpu;

## this.loai\_cpu = loai\_cpu;

## this.don\_gia = don\_gia;

## }

## 

## public void nhapTTCPU(){

## nhapTTCPU(0);

## }

## 

## public void nhapTTCPU(int type\_laptop){

## String[] tmpArr = {};

## System.out.print(">> CPU:\n");

## String ma\_cpu = "";

## System.out.println(" "

## + Lib.getlang("CPU\_COMPANY")

## + ": "

## + String.valueOf(this.hang\_cpu)

## );

## while(true){

## tmpArr = Lib.readDataLaptop(type\_laptop)[0].split("/");

## if(tmpArr[0].toLowerCase().equals("null")) {

## System.out.print(" "

## + Lib.getlang("INPUT")

## + " "

## + Lib.getlang("CPU\_CODE")

## + " "

## + String.valueOf(Lib.arrToListString(this.cpu\_arr, "CPU", true))

## + ": "

## );

## ma\_cpu = Lib.str\_input();

## } else {

## if(tmpArr.length < 2) {

## ma\_cpu = "CPU" + String.valueOf(Lib.getIndexArr(this.cpu\_arr,tmpArr[0])+1);

## System.out.println(" CPU: "

## + ma\_cpu

## + " - "

## + String.valueOf(tmpArr[0])

## );

## } else {

## System.out.print(" "

## + Lib.getlang("INPUT")

## + " "

## + Lib.getlang("CPU\_CODE")

## + " "

## + String.valueOf(Lib.arrToListString(tmpArr, "CPU", true))

## + ": "

## );

## ma\_cpu = Lib.str\_input();

## try {

## if(!Arrays.asList(this.cpu\_arr).contains(tmpArr[Integer.valueOf(ma\_cpu.toLowerCase().replace("cpu", ""))-1])){

## Integer.valueOf("KhanhNguyen9872");

## }

## } catch (Exception e) {

## System.out.println("!! " + Lib.getlang("INVALID\_CPU"));

## continue;

## }

## }

## }

## 

## try {

## this.loai\_cpu = cpu\_arr[Integer.valueOf(

## ma\_cpu.toLowerCase().replace("cpu", "")

## )-1];

## } catch (Exception e) {

## System.out.println("!! " + Lib.getlang("INVALID\_CPU\_CODE"));

## continue;

## }

## try {

## this.don\_gia = Long.valueOf(cpu\_price\_arr[Integer.valueOf(

## ma\_cpu.toLowerCase().replace("cpu", "")

## )-1]);

## } catch (Exception e) {

## this.don\_gia = Lib.int\_input(Lib.getlang("PRICE"), null, true, true, true);

## }

## 

## this.ma\_cpu = ma\_cpu;

## System.out.println(" " + Lib.getlang("PRICE") + ": " + String.valueOf(this.don\_gia) + " USD");

## break;

## }

## }

## 

## public String getHang\_cpu() {

## return hang\_cpu;

## }

## 

## public void setHang\_cpu(String hang\_cpu) {

## this.hang\_cpu = hang\_cpu;

## }

## 

## public String getMa\_cpu() {

## return ma\_cpu;

## }

## 

## public void setMa\_cpu(String ma\_cpu) {

## this.ma\_cpu = ma\_cpu;

## }

## 

## public String getLoai\_cpu() {

## return loai\_cpu;

## }

## 

## public void setLoai\_cpu(String loai\_cpu) {

## this.loai\_cpu = loai\_cpu;

## }

## 

## public long getDon\_gia() {

## return don\_gia;

## }

## 

## public void setDon\_gia(long don\_gia) {

## this.don\_gia = don\_gia;

## }

## 

## public String export\_config() {

## return String.valueOf(this.ma\_cpu)

## + "/"

## + String.valueOf(this.hang\_cpu)

## + "/"

## + String.valueOf(this.loai\_cpu)

## + "/"

## + String.valueOf(this.don\_gia) + ", ";

## }

## 

## @Override

## public String toString(){

## return String.valueOf(this.ma\_cpu)

## + " - "

## + String.valueOf(this.hang\_cpu)

## + " "

## + String.valueOf(this.loai\_cpu)

## + " ("

## + String.valueOf(this.don\_gia)

## + " "

## + Lib.getlang("MONEY")

## + ")\n";

## }

## }

## Class RAM:

## import java.util.Arrays;

## 

## public class RAM extends Laptop {

## private int so\_luong; // 1 thanh, 2 thanh

## private String[] ma\_ram; // ram1, ram2

## private String loai\_ram; // DDR3, DDR4

## private long don\_gia; // 1G DDR3 100, 1G DDR4 200

## private int dung\_luong; // 1G

## private final String[] ram\_arr = Lib.read\_data("ram\_type");

## private final String[] ram\_price\_arr = Lib.read\_data("ram\_price");

## 

## public RAM(){

## this.so\_luong = 0;

## this.ma\_ram = null;

## this.loai\_ram = "";

## this.dung\_luong = 0;

## this.don\_gia = 0;

## }

## 

## public RAM(int so\_luong, String[] ma\_ram, String loai\_ram, int dung\_luong, long don\_gia) {

## this.so\_luong = so\_luong;

## this.ma\_ram = ma\_ram;

## this.loai\_ram = loai\_ram;

## this.dung\_luong = dung\_luong;

## this.don\_gia = don\_gia;

## }

## 

## public void nhapTTRam(int type\_laptop) {

## String str;

## System.out.print(">> RAM:\n");

## int tmp = 0;

## String[] tmpArr = {};

## this.so\_luong = (int)Lib.int\_input(Lib.getlang("NUMBER\_OF\_RAM"), null, true, true,false);

## while(true){

## tmpArr = Lib.readDataLaptop(type\_laptop)[1].split("/");

## if(tmpArr[0].toLowerCase().equals("null")) {

## tmp = (int)Lib.int\_input(Lib.getlang("RAM\_TYPE"), Lib.arrToListString(this.ram\_arr, null, true), true, true,false);

## } else {

## if(tmpArr.length < 2) {

## tmp = Lib.getIndexArr(this.ram\_arr, tmpArr[0]);

## System.out.println(

## " "

## + Lib.getlang("RAM\_TYPE")

## + ": "

## + String.valueOf(tmpArr[0])

## );

## } else {

## tmp = (int)Lib.int\_input(Lib.getlang("RAM\_TYPE"), Lib.arrToListString(tmpArr, null, true), true, true,false);

## try {

## if(!Arrays.asList(this.ram\_arr).contains(tmpArr[tmp-1])){

## Integer.valueOf("KhanhNguyen9872");

## }

## } catch (Exception e) {

## System.out.println(

## "!! "

## + Lib.getlang("INVALID\_RAM")

## );

## continue;

## }

## }

## }

## try {

## this.loai\_ram = this.ram\_arr[tmp-1];

## } catch (Exception e) {

## System.out.println("!! " + Lib.getlang("INVALID\_RAM"));

## continue;

## }

## 

## try {

## this.don\_gia = Long.valueOf(this.ram\_price\_arr[tmp-1]);

## } catch (Exception e) {

## this.don\_gia = Lib.int\_input(Lib.getlang("PRICE"), null, true, true, true);

## }

## break;

## }

## this.dung\_luong = (int)Lib.int\_input(Lib.getlang("CAPACITY"), "(GB)", true, true,false);

## this.don\_gia = this.don\_gia \* this.dung\_luong;

## this.ma\_ram = new String[so\_luong];

## for(int i=0;i<so\_luong; i++) {

## System.out.print(" " + Lib.getlang("RAM\_CODE") + " " + String.valueOf(i+1) + ": ");

## this.ma\_ram[i] = Lib.str\_input();

## }

## System.out.println(" " + Lib.getlang("PRICE") + ": " + String.valueOf(this.don\_gia) + " " + Lib.getlang("MONEY"));

## System.out.println(" " + Lib.getlang("TOTAL\_PRICE") + ": " + String.valueOf(this.don\_gia \* this.so\_luong) + " " + Lib.getlang("MONEY"));

## }

## 

## public int getSo\_luong() {

## return so\_luong;

## }

## 

## public void setSo\_luong(int so\_luong) {

## this.so\_luong = so\_luong;

## }

## 

## public String[] getMa\_ram() {

## return ma\_ram;

## }

## 

## public void setMa\_ram(String[] ma\_ram) {

## this.ma\_ram = ma\_ram;

## }

## 

## public String getLoai\_ram() {

## return loai\_ram;

## }

## 

## public void setLoai\_ram(String loai\_ram) {

## this.loai\_ram = loai\_ram;

## }

## 

## public long getDon\_gia() {

## return don\_gia;

## }

## 

## public void setDon\_gia(long don\_gia) {

## this.don\_gia = don\_gia;

## }

## 

## public int getDung\_luong() {

## return dung\_luong;

## }

## 

## public void setDung\_luong(int dung\_luong) {

## this.dung\_luong = dung\_luong;

## }

## 

## public String export\_config() {

## String str = "";

## for(String z : this.getMa\_ram()) {

## str += String.valueOf(z)

## + "/"

## + String.valueOf(this.loai\_ram)

## + "/"

## + String.valueOf(this.dung\_luong)

## + "/"

## + String.valueOf(this.don\_gia)

## + ", ";

## }

## return str;

## }

## 

## @Override

## public String toString(){

## int count = 1;

## String str = "";

## for(String z : this.getMa\_ram()) {

## str += " "

## + String.valueOf(count)

## + ". "

## + String.valueOf(z)

## + " - "

## + String.valueOf(this.loai\_ram)

## + " - "

## + String.valueOf(this.dung\_luong)

## + " GB ("

## + String.valueOf(this.don\_gia)

## + " "

## + Lib.getlang("MONEY")

## + ")";

## str += "\n";

## count++;

## }

## return str;

## }

## }

## Class Screen:

* import java.util.Arrays;
* public class Screen extends Laptop {
* private String ma\_manhinh;
* private String hang\_sanxuat;
* private String do\_phangiai;
* private long don\_gia;
* private int chong\_loa;
* private final String[] screen\_arr = Lib.read\_data("screen\_type");
* private final String[] screen\_fullhd\_arr = Lib.read\_data("screen\_FullHD");
* private final String[] screen\_price\_arr = Lib.read\_data("screen\_price");
* public Screen() {
* this.ma\_manhinh = "";
* this.hang\_sanxuat = "";
* this.do\_phangiai = "";
* this.don\_gia = 0;
* this.chong\_loa = 0;
* }
* public Screen(String ma\_manhinh, String hang\_sanxuat, String do\_phangiai, long don\_gia, int chong\_loa) {
* this.ma\_manhinh = ma\_manhinh;
* this.hang\_sanxuat = hang\_sanxuat;
* this.do\_phangiai = do\_phangiai;
* this.don\_gia = don\_gia;
* this.chong\_loa = chong\_loa;
* }
* public void nhapTTScreen(int type\_laptop){
* int tmp;
* String[] tmp2;
* String[] tmpArr = {};
* System.out.print(">> " + Lib.getlang("SCREEN") + ": \n");
* System.out.print(" " + Lib.getlang("SCREEN\_CODE") + ": ");
* this.ma\_manhinh = Lib.str\_input();
* System.out.print(" " + Lib.getlang("MANUFACTURER") + ": ");
* this.hang\_sanxuat = Lib.str\_input();
* while(true) {
* tmpArr = Lib.readDataLaptop(type\_laptop)[2].split("/");
* System.out.print(" " + Lib.getlang("SCREEN\_TYPE"));
* if(tmpArr[0].toLowerCase().equals("null")) {
* System.out.print(" " + String.valueOf(Lib.arrToListString(this.screen\_arr, null, false)) + ": ");
* this.do\_phangiai = Lib.str\_input();
* } else {
* if(tmpArr.length < 2) {
* this.do\_phangiai = String.valueOf(tmpArr[0]);
* System.out.println(": " + this.do\_phangiai);
* } else {
* System.out.print(" " + String.valueOf(Lib.arrToListString(tmpArr, null, false) + ": "));
* this.do\_phangiai = Lib.str\_input();
* try {
* if(!Arrays.asList(this.screen\_arr).contains(tmpArr[Lib.getIndexArr(tmpArr, this.do\_phangiai)])){
* Integer.valueOf("KhanhNguyen9872");
* }
* } catch (Exception e) {
* System.out.println("!! " + Lib.getlang("SCREEN") + " " + Lib.getlang("IS\_NOT\_VALID"));
* continue;
* }
* }
* }
* tmp2 = Lib.read\_data("screen\_" + this.do\_phangiai);
* if(tmp2[0].toLowerCase().equals("null")) {
* this.don\_gia = Long.valueOf(this.screen\_price\_arr[Lib.getIndexArr(this.screen\_arr, this.do\_phangiai)]);
* } else {
* while(true) {
* tmp = (int)Lib.int\_input(Lib.getlang("SCREEN\_STD"), Lib.arrToListString(tmp2, null, false), true, true, false);
* try {
* this.don\_gia = Long.valueOf(this.screen\_price\_arr[Lib.getIndexArr(this.screen\_arr, this.do\_phangiai)].split("/")[tmp-1]);
* } catch (Exception e) {
* System.out.println("!! " + Lib.getlang("INVALID\_SCREEN\_STD"));
* continue;
* }
* this.chong\_loa = tmp;
* break;
* }
* }
* System.out.println(" " + Lib.getlang("PRICE") + ": " + String.valueOf(this.don\_gia) + " " + Lib.getlang("MONEY"));
* break;
* }
* }
* public String getMa\_manhinh() {
* return ma\_manhinh;
* }
* public void setMa\_manhinh(String ma\_manhinh) {
* this.ma\_manhinh = ma\_manhinh;
* }
* public String getHang\_sanxuat() {
* return hang\_sanxuat;
* }
* public void setHang\_sanxuat(String hang\_sanxuat) {
* this.hang\_sanxuat = hang\_sanxuat;
* }
* public String getDo\_phangiai() {
* return do\_phangiai;
* }
* public void setDo\_phangiai(String do\_phangiai) {
* this.do\_phangiai = do\_phangiai;
* }
* public long getDon\_gia() {
* return don\_gia;
* }
* public void setDon\_gia(long don\_gia) {
* this.don\_gia = don\_gia;
* }
* public int getChong\_loa() {
* return chong\_loa;
* }
* public void setChong\_loa(int chong\_loa) {
* this.chong\_loa = chong\_loa;
* }
* public String export\_config() {
* return String.valueOf(this.ma\_manhinh)
* + "/" + String.valueOf(this.hang\_sanxuat)
* + "/" + String.valueOf(this.do\_phangiai)
* + "/" + String.valueOf(this.chong\_loa)
* + "/" + String.valueOf(this.don\_gia);
* }
* @Override
* public String toString(){
* String str = "";
* str += String.valueOf(this.ma\_manhinh)
* + " - "
* + String.valueOf(this.hang\_sanxuat)
* + " - "
* + String.valueOf(this.do\_phangiai)
* + " - "
* + String.valueOf(this.don\_gia)
* ;
* if(this.chong\_loa != 0) {
* str += " - " + String.valueOf(this.chong\_loa);
* }
* str += " ("
* + String.valueOf(this.don\_gia)
* + " "
* + Lib.getlang("MONEY")
* + ")\n";
* return str;
* }
* }

## Class HardDisk:

* import java.util.Arrays;
* public class HardDisk extends Laptop {
* private String kieu\_ocung = null;
* private String ma\_ocung;
* private long don\_gia;
* private int dung\_luong;
* private int so\_vong\_quay;
* private int toc\_do\_doc;
* private int toc\_do\_ghi;
* private final String[] harddisk\_arr = Lib.read\_data("harddisk\_type");
* private final String[] harddisk\_price = Lib.read\_data("harddisk\_price");
* public HardDisk() {
* this.ma\_ocung = "";
* this.don\_gia = 0;
* this.dung\_luong = 0;
* this.so\_vong\_quay = 0;
* this.toc\_do\_doc = 0;
* this.toc\_do\_ghi = 0;
* }
* public HardDisk(String ma\_ocung, String kieu\_ocung, long don\_gia, int dung\_luong, int so\_vong\_quay, int toc\_do\_doc, int toc\_do\_ghi) {
* this.ma\_ocung = ma\_ocung;
* this.kieu\_ocung = kieu\_ocung;
* this.don\_gia = don\_gia;
* this.dung\_luong = dung\_luong;
* this.so\_vong\_quay = so\_vong\_quay;
* this.toc\_do\_doc = toc\_do\_doc;
* this.toc\_do\_ghi = toc\_do\_ghi;
* }
* public void set\_type\_harddisk(int type\_laptop){
* String[] tmpArr;
* String tmp;
* tmpArr = Lib.readDataLaptop(type\_laptop)[3].split("/");
* if(tmpArr[0].toLowerCase().equals("null")) {
* System.out.print(" " + Lib.getlang("HARDDISK\_TYPE") + " " + String.valueOf(Lib.arrToListString(this.harddisk\_arr, null, false)) + ": ");
* this.kieu\_ocung = Lib.str\_input();
* } else {
* while(true) {
* System.out.print(" " + Lib.getlang("HARDDISK\_TYPE"));
* if(tmpArr.length < 2) {
* this.kieu\_ocung = tmpArr[0];
* System.out.println(": " + this.kieu\_ocung);
* } else {
* System.out.print(" " + String.valueOf(Lib.arrToListString(tmpArr, null, false) + ": "));
* this.kieu\_ocung = Lib.str\_input();
* try {
* if(!Arrays.asList(this.harddisk\_arr).contains(tmpArr[Lib.getIndexArr(tmpArr, this.kieu\_ocung)])){
* Integer.valueOf("KhanhNguyen9872");
* }
* tmp = this.harddisk\_price[Lib.getIndexArr(this.harddisk\_arr, this.kieu\_ocung)];
* } catch (Exception e) {
* System.out.println("!! " + Lib.getlang("INVALID\_HARDDISK\_TYPE"));
* continue;
* }
* }
* break;
* }
* }
* }
* public void nhapTTHardDisk(int type\_laptop) {
* if(this.kieu\_ocung == null) {
* set\_type\_harddisk(type\_laptop);
* }
* String tmp = "";
* tmp = this.harddisk\_price[Lib.getIndexArr(this.harddisk\_arr, this.kieu\_ocung)];
* System.out.print(" " + Lib.getlang("HARDDISK\_CODE") + ": ");
* this.ma\_ocung = Lib.str\_input();
* this.dung\_luong = (int)Lib.int\_input(Lib.getlang("HARDDISK\_CAPACITY"), "(GB)", true, true,false);
* if(tmp.toLowerCase().equals("null")) {
* this.don\_gia = Lib.int\_input(Lib.getlang("PRICE"), "(USD)", true, true, true);
* } else {
* this.don\_gia = Long.valueOf(tmp);
* }
* if(this.kieu\_ocung.equals("HDD")) {
* this.so\_vong\_quay = (int)Lib.int\_input(Lib.getlang("NUMBER\_ROTATIONS"), "(rpm)", true, true,false);
* } else if(this.kieu\_ocung.equals("SSD")) {
* this.toc\_do\_doc = (int)Lib.int\_input(Lib.getlang("READ\_SPEED"), "(mb/s)", true, true,false);
* this.toc\_do\_ghi = (int)Lib.int\_input(Lib.getlang("WRITE\_SPEED"), "(mb/s)", true, true,false);
* }
* }
* public String getKieu\_ocung() {
* return kieu\_ocung;
* }
* public void setKieu\_ocung(String kieu\_ocung) {
* this.kieu\_ocung = kieu\_ocung;
* }
* public String getMa\_ocung() {
* return ma\_ocung;
* }
* public void setMa\_ocung(String ma\_ocung) {
* this.ma\_ocung = ma\_ocung;
* }
* public long getDon\_gia() {
* return don\_gia;
* }
* public void setDon\_gia(long don\_gia) {
* this.don\_gia = don\_gia;
* }
* public int getDung\_luong() {
* return dung\_luong;
* }
* public void setDung\_luong(int dung\_luong) {
* this.dung\_luong = dung\_luong;
* }
* public int getSo\_vong\_quay() {
* return so\_vong\_quay;
* }
* public void setSo\_vong\_quay(int so\_vong\_quay) {
* this.so\_vong\_quay = so\_vong\_quay;
* }
* public int getToc\_do\_doc() {
* return toc\_do\_doc;
* }
* public void setToc\_do\_doc(int toc\_do\_doc) {
* this.toc\_do\_doc = toc\_do\_doc;
* }
* public int getToc\_do\_ghi() {
* return toc\_do\_ghi;
* }
* public void setToc\_do\_ghi(int toc\_do\_ghi) {
* this.toc\_do\_ghi = toc\_do\_ghi;
* }
* public String export\_config() {
* return String.valueOf(this.ma\_ocung)
* + "/" + String.valueOf(this.kieu\_ocung)
* + "/" + String.valueOf(this.dung\_luong)
* + "/" + String.valueOf(this.toc\_do\_doc)
* + "/" + String.valueOf(this.toc\_do\_ghi)
* + "/" + String.valueOf(this.so\_vong\_quay)
* + "/" + String.valueOf(this.don\_gia) + ", ";
* }
* @Override
* public String toString(){
* String str = "";
* str += String.valueOf(this.ma\_ocung)
* + " - " + String.valueOf(this.kieu\_ocung)
* + " ("
* + String.valueOf(this.dung\_luong)
* + " GB)";
* if(this.kieu\_ocung.equals("HDD")) {
* str += " - ["
* + String.valueOf(this.so\_vong\_quay)
* + "rpm]";
* } else if(this.kieu\_ocung.equals("SSD")) {
* str += " - [R: "
* + String.valueOf(this.toc\_do\_doc)
* + " MB/s - W: "
* + String.valueOf(this.toc\_do\_ghi) + " MB/s]";
* }
* str += " ("
* + String.valueOf(this.don\_gia)
* + " "
* + Lib.getlang("MONEY")
* + ")\n";
* return str;
* }
* }

## Class Quanly:

* import java.io.IOException;
* import java.util.ArrayList;
* public class Quanly {
* private final String conf\_path = Lib.read\_config("conf\_path");
* private ArrayList<Laptop> dsLaptop;
* public Quanly(){
* this.dsLaptop = new ArrayList<Laptop>();
* }
* public void themLaptop(int so\_luong){
* Laptop newlaptop = null;
* for(int i=0; i<so\_luong; i++) {
* System.out.println("------------------");
* newlaptop = new Laptop();
* // Laptop
* System.out.print(">> " + String.valueOf(Lib.getlang("ENTER\_PARAMETERS\_OF\_LAPTOP")) + " " + String.valueOf(i+1) + ":\n");
* newlaptop.nhapTTLaptop(this.dsLaptop);
* // CPU
* CPU newCPU = new CPU();
* newCPU.nhapTTCPU(newlaptop.getDong\_laptop());
* newlaptop.dsCPU.add(newCPU);
* // RAM
* RAM newRAM = new RAM();
* newRAM.nhapTTRam(newlaptop.getDong\_laptop());
* newlaptop.dsRAM.add(newRAM);
* // Screen
* Screen newScreen = new Screen();
* newScreen.nhapTTScreen(newlaptop.getDong\_laptop());
* newlaptop.dsScreen.add(newScreen);
* // HardDisk
* int n = (int)Lib.int\_input(">> " + Lib.getlang("NUMBER\_OF\_HARDDISK"),null,false,true,false);
* for(int ii = 0; ii<n; ii++){
* System.out.println(">> " + Lib.getlang("HARDDISK") + " " + String.valueOf(ii+1) + ":");
* HardDisk newHardDisk = new HardDisk();
* newHardDisk.nhapTTHardDisk(newlaptop.getDong\_laptop());
* newlaptop.dsHardDisk.add(newHardDisk);
* }
* //
* if(Lib.yes\_or\_no(Lib.getlang("SAVE").toLowerCase())){
* this.dsLaptop.add(newlaptop);
* System.out.print(">> " + Lib.getlang("SAVE\_COMPLETED") + "\n");
* }
* }
* }
* private void menu\_edit(Laptop i) {
* System.out.println(
* " >> Laptop: "
* + String.valueOf(i.getMa\_laptop())
* + " - "
* + String.valueOf(i.getTen\_laptop())
* );
* System.out.println(" " + Lib.getlang("SEL\_ITEM\_EDIT"));
* System.out.println("1. " + Lib.getlang("LAPTOP\_SERIES") + " (" + String.valueOf(i.getnameDong\_laptop()) + ")");
* System.out.println("2. " + Lib.getlang("LAPTOP\_CODE") + " (" + String.valueOf(i.getMa\_laptop()) + ")");
* System.out.println("3. " + Lib.getlang("LAPTOP\_NAME") + " (" + String.valueOf(i.getTen\_laptop()) + ")");
* System.out.print("4. " + Lib.getlang("CPU"));
* if(i.dsCPU.isEmpty()) System.out.print(" (NO DATA)");
* System.out.print("\n5. " + Lib.getlang("RAM"));
* if(i.dsRAM.isEmpty()) System.out.print(" (NO DATA)");
* System.out.print("\n6. " + Lib.getlang("SCREEN"));
* if(i.dsScreen.isEmpty()) System.out.print(" (NO DATA)");
* System.out.print("\n7. " + Lib.getlang("HARDDISK"));
* if(i.dsHardDisk.isEmpty()) System.out.print(" (NO DATA)");
* System.out.println("\n0. " + Lib.getlang("BACK"));
* }
* public void suaTTLaptop() {
* if(Lib.isNotEmpty(this.dsLaptop)) {
* Lib.printAll(this.dsLaptop);
* System.out.println("------------------");
* System.out.print(Lib.getlang("INPUT") + " " + Lib.getlang("LAPTOP\_CODE") + ": ");
* int[] info = Lib.kt\_ma\_laptop(this.dsLaptop, Lib.str\_input());
* if(info[0] == 0){
* System.out.print("!! " + Lib.getlang("LAPTOP\_CODE\_NOT\_EXIST") + "\n");
* Lib.pause\_console();
* return;
* }
* Laptop oldlaptop = this.dsLaptop.get(info[1]);
* Laptop newlaptop = null;
* int n;
* while(true) {
* newlaptop = new Laptop();
* Lib.clear\_console();
* menu\_edit(oldlaptop);
* n = (int)Lib.int\_input(Lib.getlang("CHOOSE"),null,false,false,false);
* Lib.clear\_console();
* switch(n) {
* case 1:
* System.out.println("!! " + Lib.getlang("WARNING") + ": " + Lib.getlang("WARNING001"));
* System.out.println(">> " + Lib.getlang("LAPTOP\_SERIES") + " " + Lib.getlang("CURRENT").toLowerCase() + ": " + String.valueOf(oldlaptop.getnameDong\_laptop()));
* newlaptop.nhapDongLaptop();
* if(Lib.yes\_or\_no(Lib.getlang("SAVE").toLowerCase())){
* if(oldlaptop.getDong\_laptop() != newlaptop.getDong\_laptop()) {
* oldlaptop.setDong\_laptop(newlaptop.getDong\_laptop());
* oldlaptop.reset\_dsALL();
* System.out.print(" !! " + Lib.getlang("WARNING") + ": " + Lib.getlang("RESET\_INFORMATION\_COMPLETED") + "!\n");
* }
* System.out.print(">> " + Lib.getlang("SAVE\_COMPLETED") + "\n");
* Lib.pause\_console();
* }
* break;
* case 2:
* System.out.println(">> " + Lib.getlang("LAPTOP\_CODE") + " " + Lib.getlang("CURRENT").toLowerCase() + ": " + String.valueOf(oldlaptop.getMa\_laptop()));
* newlaptop.nhapMaLaptop(this.dsLaptop);
* if(Lib.yes\_or\_no(Lib.getlang("SAVE").toLowerCase())){
* oldlaptop.setMa\_laptop(newlaptop.getMa\_laptop());
* System.out.print(">> " + Lib.getlang("SAVE\_COMPLETED") + "\n");
* Lib.pause\_console();
* }
* break;
* case 3:
* System.out.println(">> " + Lib.getlang("LAPTOP\_NAME") + " " + Lib.getlang("CURRENT").toLowerCase() + ": " + String.valueOf(oldlaptop.getTen\_laptop()));
* newlaptop.nhapTenLaptop();
* if(Lib.yes\_or\_no(Lib.getlang("SAVE").toLowerCase())){
* oldlaptop.setTen\_laptop(newlaptop.getTen\_laptop());
* System.out.print(">> " + Lib.getlang("SAVE\_COMPLETED") + "\n");
* Lib.pause\_console();
* }
* break;
* case 4:
* System.out.print(">> " + Lib.getlang("CPU") + " " + Lib.getlang("CURRENT").toLowerCase() + ": " + String.valueOf(Lib.gettext\_cpu(oldlaptop, false)));
* CPU newCPU = new CPU();
* newCPU.nhapTTCPU(oldlaptop.getDong\_laptop());
* newlaptop.dsCPU.add(newCPU);
* if(Lib.yes\_or\_no(Lib.getlang("SAVE").toLowerCase())){
* oldlaptop.dsCPU = newlaptop.dsCPU;
* System.out.print(">> " + Lib.getlang("SAVE\_COMPLETED") + "\n");
* Lib.pause\_console();
* }
* break;
* case 5:
* System.out.print(">> " + Lib.getlang("RAM") + " " + Lib.getlang("CURRENT").toLowerCase() + ": " + String.valueOf(Lib.gettext\_ram(oldlaptop, false)));
* RAM newRAM = new RAM();
* newRAM.nhapTTRam(oldlaptop.getDong\_laptop());
* newlaptop.dsRAM.add(newRAM);
* if(Lib.yes\_or\_no(Lib.getlang("SAVE").toLowerCase())){
* oldlaptop.dsRAM = newlaptop.dsRAM;
* System.out.print(">> " + Lib.getlang("SAVE\_COMPLETED") + "\n");
* Lib.pause\_console();
* }
* break;
* case 6:
* System.out.print(">> " + Lib.getlang("SCREEN") + " " + Lib.getlang("CURRENT").toLowerCase() + ": " + String.valueOf(Lib.gettext\_screen(oldlaptop, false)));
* Screen newScreen = new Screen();
* newScreen.nhapTTScreen(oldlaptop.getDong\_laptop());
* newlaptop.dsScreen.add(newScreen);
* if(Lib.yes\_or\_no(Lib.getlang("SAVE").toLowerCase())){
* oldlaptop.dsScreen = newlaptop.dsScreen;
* System.out.print(">> " + Lib.getlang("SAVE\_COMPLETED") + "\n");
* Lib.pause\_console();
* }
* break;
* case 7:
* System.out.print(">> " + Lib.getlang("HARDDISK") + " " + Lib.getlang("CURRENT").toLowerCase() + ": " + String.valueOf(Lib.gettext\_cpu(oldlaptop, false)));
* n = (int)Lib.int\_input(">> Số lượng ổ cứng",null,false,true,false);
* for(int ii = 0; ii<n; ii++){
* System.out.println(">> Ổ cứng " + String.valueOf(ii+1) + ":");
* HardDisk newHardDisk = new HardDisk();
* newHardDisk.nhapTTHardDisk(oldlaptop.getDong\_laptop());
* newlaptop.dsHardDisk.add(newHardDisk);
* }
* if(Lib.yes\_or\_no(Lib.getlang("SAVE").toLowerCase())){
* oldlaptop.dsHardDisk = newlaptop.dsHardDisk;
* System.out.print(">> " + Lib.getlang("SAVE\_COMPLETED") + "\n");
* Lib.pause\_console();
* }
* break;
* case 0:
* return;
* default:
* continue;
* }
* }
* } else {
* System.out.println("!! " + Lib.getlang("LIST\_EMPTY"));
* }
* }
* public void xoaTTLaptop() {
* if(Lib.isNotEmpty(this.dsLaptop)) {
* while(true) {
* Lib.clear\_console();
* System.out.print("1. " + Lib.getlang("DEL\_WITH\_LAPTOP\_CODE") + "\n2. " + Lib.getlang("DEL\_ALL\_LAPTOP") + "\n0. " + Lib.getlang("BACK") + "\n");
* switch((int)Lib.int\_input(Lib.getlang("CHOOSE"), null, false, false, false)) {
* case 1:
* Lib.clear\_console();
* Lib.printAll(this.dsLaptop);
* System.out.print(Lib.getlang("INPUT") + " " + Lib.getlang("LAPTOP\_CODE") + ": ");
* int[] info = Lib.kt\_ma\_laptop(this.dsLaptop, Lib.str\_input());
* if(info[0] == 0){
* System.out.print("!! " + Lib.getlang("INVALID\_LAPTOP\_CODE") + "\n");
* return;
* }
* System.out.println("------------------");
* System.out.print(Lib.getlang("LAPTOP\_NAME") + ": " + String.valueOf(this.dsLaptop.get(info[1]).getTen\_laptop()) + "\n");
* if(Lib.yes\_or\_no(Lib.getlang("DEL\_LAPTOP"))){
* ArrayList<Laptop> dsTam = new ArrayList<Laptop>();
* int count = 0;
* for(Laptop i : this.dsLaptop) {
* if(count == info[1]){
* count++;
* continue;
* }
* dsTam.add(i);
* count++;
* }
* this.dsLaptop = dsTam;
* System.out.print(">> " + Lib.getlang("DEL\_COMPLETED") + "\n");
* }
* break;
* case 2:
* Lib.clear\_console();
* Lib.printAll(this.dsLaptop);
* if(Lib.yes\_or\_no(Lib.getlang("DEL\_ALL\_LAPTOP"))){
* this.dsLaptop = new ArrayList<Laptop>();
* System.out.print(">> " + Lib.getlang("DEL\_ALL\_COMPLETED") + "\n");
* }
* break;
* case 0:
* return;
* default:
* continue;
* }
* }
* } else {
* System.out.println("!! " + Lib.getlang("LIST\_EMPTY"));
* }
* }
* public void xemTTLaptop(){
* String full\_str = Lib.gettext(this.dsLaptop, false);
* if(full\_str == null) {
* return;
* } else {
* System.out.print(full\_str);
* }
* }
* public void xuatTT2File() throws IOException{
* if(Lib.isNotEmpty(this.dsLaptop)) {
* Lib.fwrite("ds\_laptop.txt", Lib.gettext(this.dsLaptop, false), false);
* System.out.print(">> " + Lib.getlang("EXPORT\_COMPLETED") + " [ds\_laptop.txt]\n");
* } else {
* System.out.println("!! " + Lib.getlang("LIST\_EMPTY"));
* }
* }
* public void nhapCONF() {
* int count = 1;
* int count2;
* Laptop newLaptop = null;
* String name = "";
* String tmpStr = "";
* String[] tmp = {};
* String[] tmp2 = {};
* String[] list\_target = {"series", "code", "name", "cpu", "ram", "screen", "harddisk"};
* System.out.println(">> Các file data tại [" + this.conf\_path + "]");
* for(String s : Lib.lsdir(this.conf\_path)) {
* if(s.contains(".conf")) {
* System.out.println("=> " + s.substring(0, s.length() - 5));
* }
* }
* System.out.println("------------------");
* // input file name conflaptopbut
* String str = this.conf\_path + Lib.str\_input(Lib.getlang("INPUT\_FILE\_NAME") + ": ") + ".conf";
* tmp = Lib.read\_data(str, "CONFIG", false, false);
* if(tmp == null) {
* System.out.println("!! " + Lib.getlang("DATA\_ERROR"));
* return;
* } else if(!tmp[0].equals("Hoa\_Don\_Laptop")) {
* System.out.println("!! " + Lib.getlang("DATA\_ERROR"));
* return;
* }
* System.out.println(">> " + Lib.getlang("PROCESS\_DATA") + "....");
* while(true) {
* newLaptop = new Laptop();
* for(String target : list\_target) {
* name = target + String.valueOf(count);
* tmp = Lib.read\_data(str, name, false, true);
* if(tmp == null) {
* System.out.println("\n>>" + Lib.getlang("IMPORT\_CONFIG\_COMPLETED"));
* return;
* }
* if(tmp.length <= 0) continue;
* // series
* if(target.equals(list\_target[0])) {
* newLaptop.setDong\_laptop(Lib.getIndexArr(Lib.read\_data("laptop\_type"), tmp[0]) + 1);
* }
* // code
* if(target.equals(list\_target[1])) {
* tmpStr = Lib.arrToString(tmp);
* int[] info = Lib.kt\_ma\_laptop(this.dsLaptop, tmpStr);
* if(info[0]==1){
* System.out.println("!! " + Lib.getlang("LAPTOP\_CODE\_EXIST") + " [" + Lib.getlang("LAPTOP\_CODE") + ": " + tmpStr + "]");
* if(Lib.yes\_or\_no(Lib.getlang("CHANGE\_LAPTOP\_CODE").toLowerCase())) {
* newLaptop.nhapMaLaptop(this.dsLaptop);
* continue;
* } else {
* return;
* }
* } else {
* System.out.println(" - " + Lib.getlang("LAPTOP\_CODE") + ": " + tmpStr);
* }
* newLaptop.setMa\_laptop(tmpStr);
* }
* // name
* if(target.equals(list\_target[2])) {
* newLaptop.setTen\_laptop(Lib.arrToString(tmp));
* }
* // cpu
* if(target.equals(list\_target[3])) {
* for(String cpu : tmp) {
* if(cpu.isBlank()) continue;
* tmp2 = cpu.split("/");
* CPU newCPU = new CPU(
* tmp2[0],
* tmp2[1],
* tmp2[2],
* Long.valueOf(tmp2[3])
* );
* newLaptop.dsCPU.add(newCPU);
* }
* }
* // ram
* if(target.equals(list\_target[4])) {
* int len\_ram = 0;
* for(String ram : tmp) {
* if(ram.isBlank()) continue;
* len\_ram++;
* }
* String[] ma\_ram = new String[len\_ram];
* count2 = 0;
* for(String ram : tmp) {
* if(ram.isBlank()) continue;
* tmp2 = ram.split("/");
* ma\_ram[count2] = tmp2[0];
* count2++;
* }
* RAM newRAM = new RAM(
* len\_ram,
* ma\_ram,
* tmp2[1],
* Integer.valueOf(tmp2[2]),
* Long.valueOf(tmp2[3])
* );
* newLaptop.dsRAM.add(newRAM);
* }
* // Screen
* if(target.equals(list\_target[5])) {
* for(String screen : tmp) {
* if(screen.isBlank()) continue;
* tmp2 = screen.split("/");
* Screen newScreen = new Screen(
* tmp2[0],
* tmp2[1],
* tmp2[2],
* Long.valueOf(tmp2[4]),
* Integer.valueOf(tmp2[3])
* );
* newLaptop.dsScreen.add(newScreen);
* }
* }
* // HardDisk
* if(target.equals(list\_target[6])) {
* for(String harddisk : tmp) {
* if(harddisk.isBlank()) continue;
* tmp2 = harddisk.split("/");
* HardDisk newHardDisk = new HardDisk(
* tmp2[0],
* tmp2[1],
* Long.valueOf(tmp2[6]),
* Integer.valueOf(tmp2[2]),
* Integer.valueOf(tmp2[5]),
* Integer.valueOf(tmp2[3]),
* Integer.valueOf(tmp2[4])
* );
* newLaptop.dsHardDisk.add(newHardDisk);
* }
* }
* }
* this.dsLaptop.add(newLaptop);
* count++;
* }
* }
* public void xuatCONF() throws IOException {
* if(Lib.isNotEmpty(this.dsLaptop)) {
* String str = this.conf\_path + Lib.str\_input(Lib.getlang("INPUT\_FILE\_NAME") + ": ") + ".conf";
* Lib.fwrite(str,
* "CONFIG = Hoa\_Don\_Laptop"
* + "\n"
* + Lib.gettext(this.dsLaptop, true),
* false
* );
* System.out.print(">> " + Lib.getlang("EXPORT\_CONFIG\_COMPLETED") + " [" + str + "]\n");
* } else {
* System.out.println("!! " + Lib.getlang("LIST\_EMPTY"));
* }
* }
* public static void menu(){
* System.out.print(" >> " + Lib.getlang("MAIN\_PROGRAM") + " <<\n\n");
* System.out.print("1. " + Lib.getlang("ADD\_LAPTOP\_INFO") + "\n");
* System.out.print("2. " + Lib.getlang("EDIT\_LAPTOP\_INFO") + "\n");
* System.out.print("3. " + Lib.getlang("VIEW\_LAPTOP\_INFO") + "\n");
* System.out.print("4. " + Lib.getlang("DEL\_LAPTOP\_INFO") + "\n");
* System.out.print("5. " + Lib.getlang("EXPORT\_LAPTOP\_INFO") + "\n");
* System.out.print("6. " + Lib.getlang("IMPORT\_LAPTOP\_CONFIG") + "\n");
* System.out.print("7. " + Lib.getlang("EXPORT\_LAPTOP\_CONFIG") + "\n");
* System.out.print("8. " + Lib.getlang("EXIT\_PROGRAM") + "\n\n");
* }
* }

## Class Lib

## import java.io.BufferedWriter;

## import java.io.File;

## import java.util.ArrayList;

## import java.util.Scanner;

## import java.io.FileInputStream;

## import java.io.FileWriter;

## import java.io.IOException;

## import java.io.InputStreamReader;

## import java.io.Writer;

## import java.nio.charset.Charset;

## import java.util.Properties;

## 

## public class Lib {

## private static Language lang = null;

## private static int MAX\_DATA = 20;

## private static String[] is\_load\_path = new String[Lib.MAX\_DATA];

## private static int[] properties\_index = new int[Lib.MAX\_DATA];

## private static int loaded = 0;

## private static ArrayList<Properties> properties = new ArrayList<Properties>();

## private static FileInputStream inputStream = null;

## private static final String FILE\_CONFIG = "data/config.properties";

## protected static Scanner keyboard = new Scanner(System.in, "UTF-8");

## 

## protected static void clear\_console(){

## System.out.print("\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n");

## }

## 

## protected static String str\_input(){

## return Lib.keyboard.next() + Lib.keyboard.nextLine();

## }

## 

## protected static String str\_input(String s){

## System.out.print(s);

## return Lib.keyboard.next() + Lib.keyboard.nextLine();

## }

## 

## protected static void pause\_console(){

## System.out.print("\n>> " + Lib.getlang("PRESS\_ENTER\_CONTINUE") + " ");

## Lib.keyboard.nextLine();

## Lib.keyboard.nextLine();

## }

## 

## protected static boolean yes\_or\_no(String name) {

## String user\_inp;

## while(true) {

## System.out.print(">> " + Lib.getlang("DO\_YOU\_WANT") + " " + name + "? [Y/n]: ");

## user\_inp = Lib.str\_input();

## if(user\_inp.equalsIgnoreCase("y")){

## return true;

## } else if(user\_inp.equalsIgnoreCase("n")){

## return false;

## }

## }

## }

## 

## protected static long int\_input(String name, String name2, boolean is\_tab, boolean greater\_than\_zero, boolean is\_long) {

## long var = 0;

## while(true) {

## if(is\_tab) {

## System.out.print(" ");

## }

## System.out.print(name);

## if(name2 != null) {

## System.out.print(" " + name2);

## }

## System.out.print(": ");

## try {

## if(is\_long){

## var = Lib.keyboard.nextLong();

## } else {

## var = Lib.keyboard.nextInt();

## }

## } catch (Exception e) {

## System.out.println("!! " + Lib.getlang("ENTER\_NUM\_PLZ"));

## Lib.keyboard.next();

## continue;

## }

## if(var<=0 && greater\_than\_zero) {

## System.out.println("!! " + name + " " + Lib.getlang("IS\_NOT\_VALID"));

## continue;

## }

## break;

## }

## return var;

## }

## 

## protected static int[] kt\_ma\_laptop(ArrayList<Laptop> dsLaptop, String ma\_laptop) {

## int count = 0;

## int[] info = {0, 0};

## for (Laptop i: dsLaptop){

## if(i.getMa\_laptop().equals(ma\_laptop)){

## info[0] = 1;

## info[1] = count;

## }

## count++;

## }

## return info;

## }

## 

## protected static void printAll(ArrayList<Laptop> dsLaptop){

## int count = 0;

## System.out.println(">> " + Lib.getlang("LIST\_LAPTOP") + ": ");

## for(Laptop i : dsLaptop) {

## System.out.println(" "

## + String.valueOf(count+1)

## + ". "

## + String.valueOf(i.getMa\_laptop())

## );

## count++;

## }

## }

## 

## protected static String gettext\_cpu(Laptop i, boolean is\_conf) {

## if (i.dsCPU.size() == 0) {

## if (is\_conf) {

## return "";

## }

## return Lib.getlang("NO\_INFORMATION") + " " + Lib.getlang("CPU") + "\n";

## }

## String str = "";

## for(CPU j : i.dsCPU) {

## if(is\_conf) {

## str += j.export\_config();

## } else {

## str += j;

## }

## }

## return str;

## }

## 

## protected static String gettext\_ram(Laptop i, boolean is\_conf) {

## if (i.dsRAM.size() == 0) {

## if (is\_conf) {

## return "";

## }

## return Lib.getlang("NO\_INFORMATION") + " " + Lib.getlang("RAM") + "\n";

## }

## String str;

## if(is\_conf) str = "";

## else str = "\n";

## for(RAM j : i.dsRAM) {

## if(is\_conf) {

## str += j.export\_config();

## } else {

## str += j;

## }

## }

## return str;

## }

## 

## protected static String gettext\_screen(Laptop i, boolean is\_conf){

## if (i.dsScreen.size() == 0) {

## if (is\_conf) {

## return "";

## }

## return Lib.getlang("NO\_INFORMATION") + " " + Lib.getlang("SCREEN") + "\n";

## }

## String str = "";

## for(Screen j : i.dsScreen) {

## if(is\_conf) {

## str += j.export\_config();

## } else {

## str += j;

## }

## }

## return str;

## }

## 

## protected static String gettext\_harddisk(Laptop i, boolean is\_conf) {

## if (i.dsHardDisk.size() == 0) {

## if (is\_conf) {

## return "";

## }

## return Lib.getlang("NO\_INFORMATION") + " " + Lib.getlang("HARDDISK") + "\n";

## }

## int count = 1;

## String str;

## if(is\_conf) str = "";

## else str = "\n";

## for(HardDisk j : i.dsHardDisk) {

## if(is\_conf) {

## str += j.export\_config();

## } else {

## str += " " + String.valueOf(count) + ". " + j;

## }

## count++;

## }

## return str;

## }

## 

## protected static String gettext(ArrayList<Laptop> dsLaptop, boolean is\_conf){

## if(isNotEmpty(dsLaptop)) {

## String full\_str = "";

## int count = 1;

## for (Laptop i: dsLaptop) {

## if(!is\_conf) {

## full\_str += "-------------------\n";

## } else {

## full\_str += "# Config by KhanhNguyen9872\n\n";

## }

## 

## // number laptop

## if(!is\_conf) {

## full\_str += " >> " + Lib.getlang("NUM\_OF\_LAPTOP") + " " + String.valueOf(count) + "\n";

## }

## 

## // laptop series

## if(!is\_conf) {

## full\_str += "- " + Lib.getlang("LAPTOP\_SERIES") + ": ";

## } else {

## full\_str += "series" + String.valueOf(count) + " = ";

## }

## full\_str += String.valueOf(i.getnameDong\_laptop()) + "\n";

## 

## // laptop code

## if(!is\_conf) {

## full\_str += "- " + Lib.getlang("LAPTOP\_CODE") + ": ";

## } else {

## full\_str += "code" + String.valueOf(count) + " = ";

## }

## full\_str += String.valueOf(i.getMa\_laptop()) + "\n";

## 

## // laptop name

## if(!is\_conf) {

## full\_str += "- " + Lib.getlang("LAPTOP\_NAME") + ": ";

## } else {

## full\_str += "name" + String.valueOf(count) + " = ";

## }

## full\_str += String.valueOf(i.getTen\_laptop()) + "\n";

## 

## // CPU

## if(!is\_conf) {

## full\_str += "- " + Lib.getlang("CPU") + ": ";

## } else {

## full\_str += "cpu" + String.valueOf(count) + " = ";

## }

## full\_str += gettext\_cpu(i, is\_conf);

## 

## // RAM

## if(!is\_conf) {

## full\_str += "- " + Lib.getlang("RAM") + ": ";

## } else {

## full\_str += "\nram" + String.valueOf(count) + " = ";

## }

## full\_str += gettext\_ram(i, is\_conf);

## 

## // SCREEN

## if(!is\_conf) {

## full\_str += "- " + Lib.getlang("SCREEN") + ": ";

## } else {

## full\_str += "\nscreen" + String.valueOf(count) + " = ";

## }

## full\_str += gettext\_screen(i, is\_conf);

## 

## // HARDDISK

## if(!is\_conf) {

## full\_str += "- " + Lib.getlang("HARDDISK") + ": ";

## } else {

## full\_str += "\nharddisk" + String.valueOf(count) + " = ";

## }

## full\_str += gettext\_harddisk(i, is\_conf);

## 

## full\_str += "\n";

## if(!is\_conf) {

## full\_str += ">> " + Lib.getlang("TOTAL\_PRICE") + ": " + String.valueOf(i.getTong\_gia()) + " " + Lib.getlang("MONEY") + "\n";

## } else {

## full\_str += "\n";

## }

## count++;

## }

## if(!is\_conf) {

## full\_str += "-------------------\n";

## }

## return full\_str;

## } else {

## System.out.println("!! " + Lib.getlang("LIST\_EMPTY"));

## }

## return null;

## }

## 

## private static boolean is\_load(String file) {

## for(String s : Lib.is\_load\_path) {

## try {

## if(s.equals(file)) {

## return true;

## }

## } catch (Exception e) {

## return false;

## }

## }

## return false;

## }

## 

## protected static void load\_language() {

## Lib.lang = new Language();

## }

## 

## protected static String getlang(String code) {

## if(Lib.lang == null) {

## load\_language();

## }

## boolean a = false;

## String str = "";

## for(String s : Lib.read\_data(Lib.lang.Path(), code, true, false)) {

## if(a) {

## str += ",";

## }

## str += s;

## a = true;

## }

## return str;

## }

## 

## protected static String read\_config(String target) {

## return Lib.arrToString(read\_data(FILE\_CONFIG, target, true, false));

## }

## 

## protected static String[] read\_data(String target) {

## return read\_data(Lib.read\_config("data\_path"), target, true, false);

## }

## 

## protected static String[] read\_data(String file, String target, boolean is\_exit, boolean is\_err\_null) {

## Properties tmp\_properties = new Properties();

## if(!is\_load(file)) {

## if(Lib.loaded >= Lib.MAX\_DATA) {

## System.out.println("\nCannot load data! Program limited " + Lib.MAX\_DATA + " data!\n");

## Lib.pause\_console();

## return null;

## }

## try {

## Lib.inputStream = new FileInputStream(new File(file));

## tmp\_properties.load(new InputStreamReader(Lib.inputStream, Charset.forName("UTF-8")));

## } catch (Exception e) {

## if(!is\_err\_null) System.out.println("!! " + Lib.getlang("READ\_FAILED") + ": " + String.valueOf(file));

## if(is\_exit) {

## System.exit(1);

## } else {

## return null;

## }

## }

## Lib.properties.add(tmp\_properties);

## Lib.is\_load\_path[Lib.loaded] = file;

## Lib.loaded++;

## }

## 

## try {

## if(target == null) {

## return null;

## } else {

## return rmSpaceArr(

## rmBlankArr(Lib.properties.get(getIndexArr(Lib.is\_load\_path, file)).getProperty(target).split(","))

## );

## }

## } catch (Exception e) {

## if(!is\_err\_null) System.out.println("\n!! Missing data [" + String.valueOf(target) + "] from [" + String.valueOf(file) + "] !!\n");

## if(is\_exit) {

## System.exit(1);

## } else {

## return null;

## }

## }

## return null;

## }

## 

## protected static String[] rmBlankArr(String[] arr) {

## int len = arr.length, count = 0;

## String[] newArr = null;

## for(String s : arr) {

## if(s.isBlank()) {

## len -= 1;

## }

## }

## newArr = new String[len];

## for(String s : arr) {

## if(!s.isBlank()) {

## newArr[count] = s;

## count++;

## }

## }

## return newArr;

## }

## 

## protected static String[] rmSpaceArr(String[] arr) {

## int count = 0;

## for(String s : arr) {

## arr[count] = rmSpace(s);

## count++;

## }

## return arr;

## }

## 

## protected static int getIndexArr(String[] str, String target) {

## int count = 0;

## for(String s : str) {

## if(s.equals(target)) {

## return count;

## }

## count++;

## }

## return -1;

## }

## 

## protected static String rmSpace(String s) {

## if (s.charAt(0) == ' ') {

## s = s.substring(1, s.length());

## }

## if (s.charAt(s.length()-1) == ' ') {

## s = s.substring(0, s.length()-1);

## }

## return s;

## }

## 

## protected static String[] readDataLaptop(int type\_laptop){

## String[] Empty = {};

## int count = 1;

## for(String s : read\_data("laptop\_type")) {

## if(count == type\_laptop) {

## return read\_data(s + "\_data");

## }

## count++;

## }

## return Empty;

## }

## 

## protected static String arrToString(String[] arr) {

## return arrToString(arr, ",");

## }

## 

## protected static String arrToString(String[] arr, String var) {

## if(arr == null) return null;

## String tmpStr = "";

## for(String ss : arr) {

## if(!tmpStr.isBlank()) tmpStr += var;

## tmpStr += ss;

## }

## return tmpStr;

## }

## 

## protected static String arrToListString(String[] arr, String name, boolean is\_id) {

## if(name == null) {

## name = "";

## }

## int count = 1;

## String str = "(";

## for(String s : arr) {

## if(count > 1) {

## str += ", ";

## }

## if(is\_id) {

## str += name

## + String.valueOf(count)

## + " - ";

## }

## str += String.valueOf(s);

## count++;

## }

## str += ")";

## return str;

## }

## 

## public static String[] lsdir(String path) {

## String[] lstFile = null;

## File folder = new File(path);

## File[] lstOF = folder.listFiles();

## int len = lstOF.length;

## 

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

## if (!lstOF[i].isFile()) {

## len -= 1;

## }

## }

## 

## lstFile = new String[len];

## int count = 0;

## 

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

## if (lstOF[i].isFile()) {

## lstFile[count] = lstOF[i].getName();

## count++;

## }

## }

## 

## return lstFile;

## }

## 

## protected static void fwrite(String filename, String s, boolean is\_append) throws IOException {

## Writer output;

## output = new BufferedWriter(new FileWriter(filename, is\_append));

## output.append(s);

## output.close();

## }

## 

## protected static boolean isNotEmpty(ArrayList ds) {

## return ds.size() > 0;

## }

## }

## Class Language

## public class Language {

## private final String lang\_path = Lib.read\_config("lang\_path");

## private String[] all\_language = new String[100];

## private String[] all\_path\_lang = new String[100];

## private int count = 0;

## public String current\_lang = "";

## public String current\_path\_lang = "";

## 

## public Language() {

## load\_language();

## set\_language();

## }

## 

## public String Lang() {

## return current\_lang;

## }

## 

## public String Path() {

## return current\_path\_lang;

## }

## 

## public void set\_language() {

## int count;

## int choose;

## while(true) {

## count = 1;

## Lib.clear\_console();

## System.out.println(">> SET LANGUAGE <<");

## for(String s : this.all\_language) {

## if(s == null) {

## continue;

## }

## System.out.println(

## String.valueOf(count)

## + ". "

## + String.valueOf(s)

## + " ("

## + String.valueOf(this.all\_path\_lang[count-1].replace(this.lang\_path, ""))

## + ")"

## );

## count++;

## }

## System.out.println();

## choose = (int)Lib.int\_input(" - Choose", null, false, true, false)-1;

## try {

## if(this.all\_language[choose] != null) {

## this.current\_lang = this.all\_language[choose];

## this.current\_path\_lang = this.all\_path\_lang[choose];

## break;

## }

## } catch (Exception e) {

## continue;

## }

## }

## }

## 

## public void load\_language() {

## for(String s : Lib.lsdir(this.lang\_path)) {

## if(s.contains(".lang")) {

## this.all\_language[this.count] = Lib.read\_data(this.lang\_path + s, "name", true, false)[0];

## this.all\_path\_lang[this.count] = this.lang\_path + s;

## this.count++;

## }

## }

## }

## 

## }

## Class main

## import java.io.IOException;

## import java.io.PrintStream;

## 

## public class Main {

## public static void main(String args[]) throws IOException{

## System.setOut(new PrintStream(System.out, true, "UTF8"));

## Lib.read\_config(null);

## Lib.read\_data(null);

## Lib.load\_language();

## Quanly quanLy = new Quanly();

## int choose = 0;

## while(true){

## Lib.clear\_console();

## Quanly.menu();

## System.out.print(Lib.getlang("CHOOSE") + ": ");

## try {

## choose = Lib.keyboard.nextInt();

## } catch (Exception e) {

## Lib.keyboard.next();

## continue;

## }

## Lib.clear\_console();

## switch(choose) {

## case 1:

## System.out.println(Lib.getlang("PRESS\_0\_TO\_BACK"));

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

## quanLy.themLaptop((int)Lib.int\_input(Lib.getlang("NUMBER\_OF\_LAPTOP"), null, false, false,false));

## Lib.pause\_console();

## break;

## case 2:

## quanLy.suaTTLaptop();

## break;

## case 3:

## quanLy.xemTTLaptop();

## Lib.pause\_console();

## break;

## case 4:

## quanLy.xoaTTLaptop();

## Lib.pause\_console();

## break;

## case 5:

## quanLy.xuatTT2File();

## Lib.pause\_console();

## break;

## case 6:

## quanLy.nhapCONF();

## Lib.pause\_console();

## break;

## case 7:

## quanLy.xuatCONF();

## Lib.pause\_console();

## break;

## case 8:

## Lib.clear\_console();

## if(Lib.yes\_or\_no(Lib.getlang("EXIT").toLowerCase())) {

## System.exit(0);

## }

## default:

## continue;

## }

## }

## }

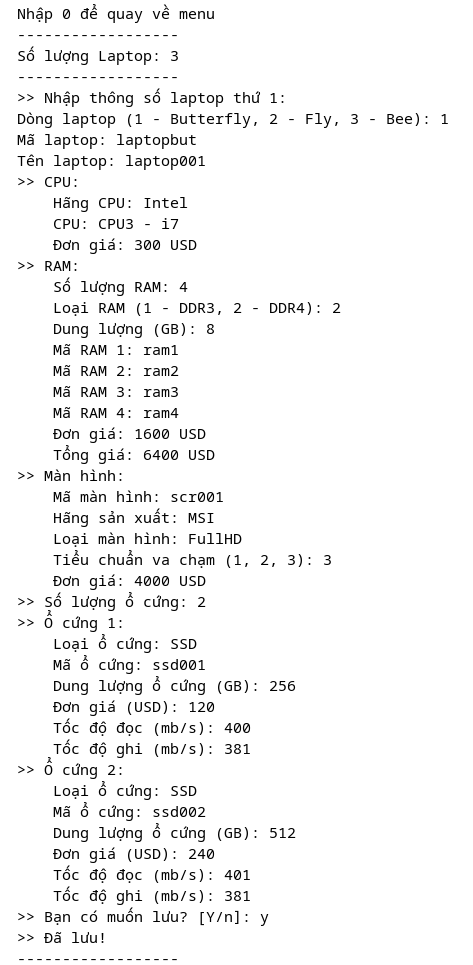
## }

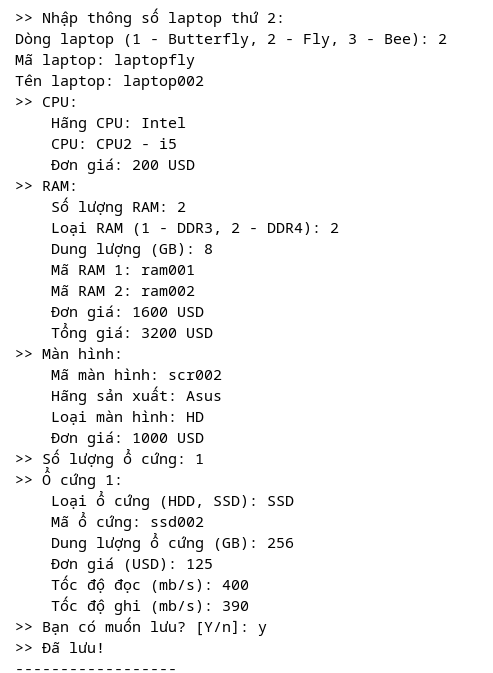
# Kiểm thử lập bộ test

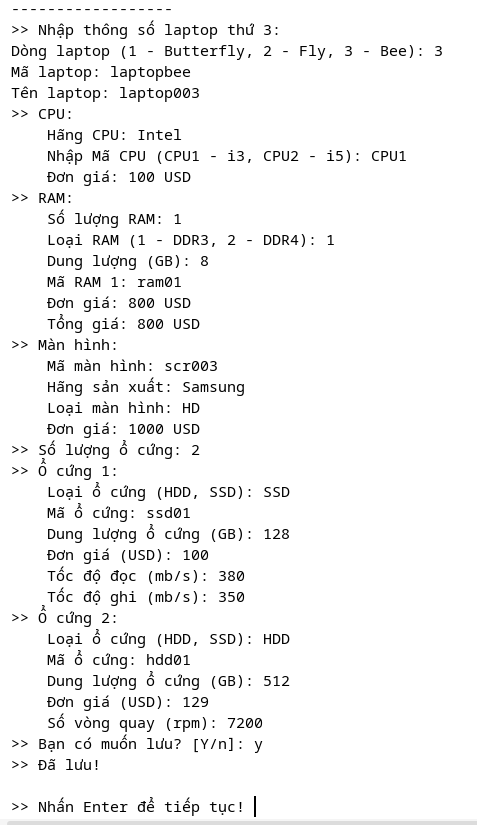
Dữ liệu laptop

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Mã Laptop | Dòng Laptop | Tên Laptop | CPU | RAM | Màn hình | Ổ cứng |
| laptopbut | Butterfly | laptop001 | CPU3 – i7 | ram1 – DDR4 – 8GB,  ram2 – DDR4 – 8GB,  ram3 – DDR4 – 8GB,  ram4 – DDR4 – 8GB | scr001 – MSI – FullHD – 3 | ssd001 – SSD – 256GB –  120 USD - 400mb/s – 381mb/s,  ssd002 – SSD – 512GB - 240 USD – 401mb/s – 381mb/s |
| laptopfly | Fly | laptop002 | CPU2 – i5 | ram001 – DDR4 – 8GB,  ram002 – DDR4 – 8GB | scr002 – Asus – HD | ssd002 – SSD – 256GB –  125 USD - 400mb/s – 390mb/s |
| laptopbee | Bee | laptop003 | CPU1 – i3 | ram01 – DDR3 – 8GB | scr003 – Samsung – HD | ssd01 – SSD – 128GB - 100 USD – 380mb/s – 350mb/s  hdd01 – HDD –512GB –129 USD – 7200rpm |

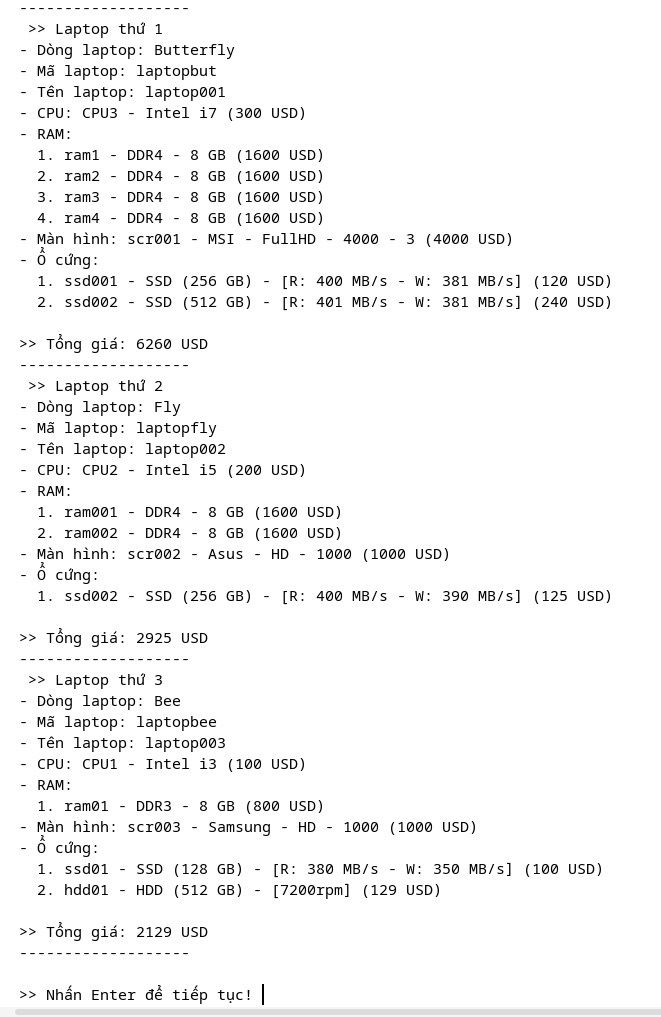
## Nhập danh sách Laptop







## 2. Xem thông tin tất cả Laptop

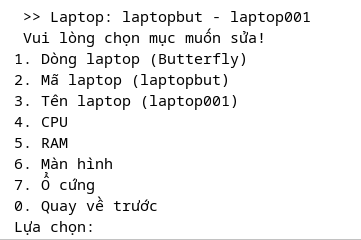


## 3. Sửa thông tin Laptop

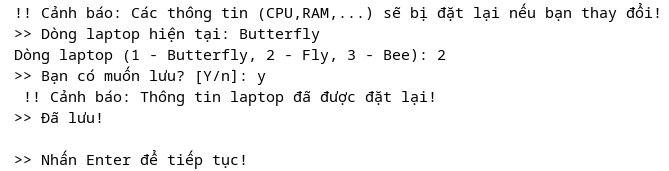
A picture containing text, font, receipt, screenshot

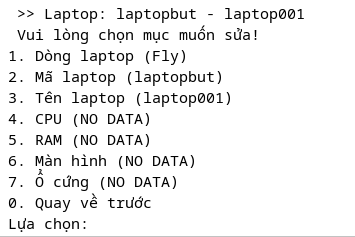
Description automatically generated

**laptopbut**

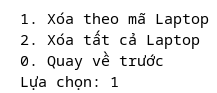


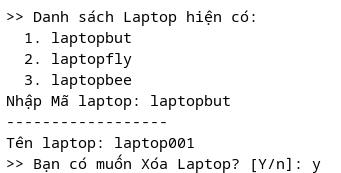
**1**

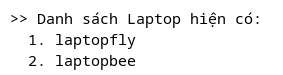




## 4. Xóa thông tin Laptop

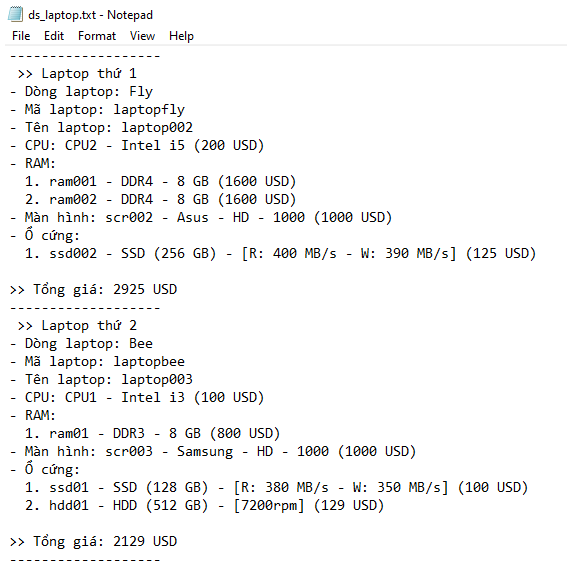






## 5. Xuất hóa đơn

## 



## 6. Xuất dữ liệu Laptop ra file

## 



## 7. Nhập dữ liệu Laptop ra file

