

TRƯỜNG ĐẠI HỌC GIA ĐỊNH
KHOA CÔNG NGHỆ THÔNG TIN



GIA ĐỊNH
UNIVERSITY

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.....	4
II.	Xác định các lớp của bài toán.....	5
1.	Class Laptop.....	5
2.	Class CPU	5
3.	Class RAM	5
4.	Class Screen	6
5.	Class HardDisk	6
6.	Class Language	6
7.	Class Lib kế thừa Class Language.....	6
8.	Class Quanly	8
III.	Mô tả thuật toán thao tác.....	11
1.	Thêm thông tin Laptop mới.	11
2.	Sửa thông tin Laptop.....	11
3.	Xem thông tin Laptop.	11
4.	Xóa thông tin Laptop.	11
IV.	Cài đặt bài toán.....	12
1.	Class Laptop:.....	12
2.	Class CPU:.....	15
3.	Class RAM:	17
4.	Class Screen:.....	21
5.	Class HardDisk:	24
6.	Class Quanly:.....	28
7.	Class Lib	38
8.	Class Language	46
9.	Class main	47
10.	File config.properties	48
11.	File data.properties	48
12.	File ngôn ngữ - vi.lang.....	49
V.	Kiểm thử lập bộ test.....	51
1.	Nhập danh sách Laptop.....	52

2.	Xem thông tin tất cả Laptop	55
3.	Sửa thông tin Laptop	56
4.	Xóa thông tin Laptop	57
5.	Xuất hóa đơn.....	58
6.	Xuất dữ liệu Laptop ra file	59
7.	Nhập dữ liệu Laptop ra file	60

I. 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

II. Xác định các lớp của bài toán

1. 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.

2. Class CPU

- 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.

3. Class RAM

- 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.

4. Class Screen

- 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.

5. Class HardDisk

- Thuộc tính: String kieu_ocung, ma_ocung; long don_gia; int dung_luong, so_vong_quay, toc_do_doc, toc_do_ghi; String[] hddisk_arr, hddisk_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.

6. Class Language

- Thuộc tính: String lang_path, current_lang, current_lang_path; int count; String[] all_language, all_path_lang.
- Phương thức:
 - language(): gọi hàm private [load_language(), set_language()].
 - set_language(): Chọn ngôn ngữ theo danh sách ngôn ngữ.
 - load_language(): Khởi tạo danh sách ngôn ngữ từ các file.

7. Class Lib kế thừa Class Language

- Thuộc tính: 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 tử 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.
- `isEmpty()`: Trả về true nếu danh sách có độ dài lớn hơn 0 và ngược lại.

8. 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
<ul style="list-style-type: none"> - 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
<ul style="list-style-type: none"> - String ma_cpu, loai_cpu, hang_cpu - int don_gia - String[] cpu_arr, cpu_price_arr + Constructor() + Setter(), Getter() + nhapTTCPU() + export_config() + toString()

RAM
<ul style="list-style-type: none"> - 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()

Screen
<ul style="list-style-type: none"> - 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()

Lib	Language
<ul style="list-style-type: none"> - Language lang - int MAX_DATA, loaded - String[] is_load_path - ArrayList<Properties> properties - FileInputStream inputStream - Scanner keyboard + 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() + isEmpty() 	<ul style="list-style-type: none"> - String lang_path, current_lang, current_lang_path - int count - String[] all_language, all_path_lang + Language() + set_language() + load_language()
	HardDisk
	<ul style="list-style-type: none"> - 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
	<ul style="list-style-type: none"> - String conf_path - ArrayList<Laptop> dsLaptop + Constructor() + themLaptop() + menu_edit() + suaTTLaptop() + xemTTLaptop() + xuatTT2File() + nhapCONF() + xuatCONF() + menu()

III. Mô tả thuật toán thao tác

1. 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.

2. 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.

3. 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.

4. 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.

IV. Cài đặt bài toán

1. Class Laptop:

```
1  import java.util.ArrayList;
2  public class Laptop {
3      private String ma_laptop;
4      private String ten_laptop;
5      private int dong_laptop;
6      private final String[] laptop_arr = Lib.read_data("laptop_type");
7      protected ArrayList<CPU> dsCPU = null;
8      protected ArrayList<RAM> dsRAM = null;
9      protected ArrayList<HardDisk> dsHardDisk = null;
10     protected ArrayList<Screen> dsScreen = null;
11
12     public int getDong_laptop() {
13         return this.dong_laptop;
14     }
15
16     public void reset_dsCPU() {
17         this.dsCPU = new ArrayList<CPU>();
18     }
19
20     public void reset_dsRAM() {
21         this.dsRAM = new ArrayList<RAM>();
22     }
23
24     public void reset_dsScreen() {
25         this.dsScreen = new ArrayList<Screen>();
26     }
27
28     public void reset_dsHardDisk() {
29         this.dsHardDisk = new ArrayList<HardDisk>();
30     }
31
32     public void reset_dsALL(){
33         reset_dsCPU();
34         reset_dsRAM();
35         reset_dsScreen();
36         reset_dsHardDisk();
37     }
38
39     public String getnameDong_laptop() {
40         return this.laptop_arr[this.dong_laptop-1];
41     }
42
43     public boolean setDong_laptop(int dong_laptop) {
44         int count = 1;
45         for(String s : this.laptop_arr) {
46             if(dong_laptop == count) {
47                 this.dong_laptop = dong_laptop;
48                 return true;
49             }
50             count++;
51         }
52         return false;
53     }
54 }
```

```

54
55 public void nhapMaLaptop(ArrayList<Laptop> dsLaptop) {
56     String tmp;
57     while(true) {
58         System.out.print(Lib.getlang("LAPTOP_CODE") + ": ");
59         tmp = Lib.str_input();
60         int[] info = Lib.kt_ma_laptop(dsLaptop, tmp);
61         if(info[0]==1){
62             System.out.println("!! " + Lib.getlang("LAPTOP_CODE_EXIST"));
63             continue;
64         }
65         this.ma_laptop = tmp;
66         break;
67     }
68 }
69
70 public void nhapTenLaptop() {
71     System.out.print(Lib.getlang("LAPTOP_NAME") + ": ");
72     this.ten_laptop = Lib.str_input();
73 }
74
75 public void nhapDongLaptop() {
76     int count;
77     while(true) {
78         count = 1;
79         System.out.print(Lib.getlang("LAPTOP_SERIES") + " (");
80         for(String s : this.laptop_arr) {
81             if(count>1){
82                 System.out.print(", ");
83             }
84             System.out.print(String.valueOf(count)
85                 + " - "
86                 + Lib.rmSpace(s));
87             count++;
88         }
89         System.out.print("): ");
90         try {
91             if (this.setDong_laptop(
92                 Integer.valueOf(Lib.keyboard.next()
93                     + Lib.keyboard.nextLine())
94             )) {
95                 break;
96             }
97         } catch (Exception e) {
98             System.out.println("!! " + Lib.getlang("ENTER_NUM_PLZ"));
99         }
100     }
101 }
102
103 public void nhapTTLaptop(ArrayList<Laptop> dsLaptop) {
104     nhapDongLaptop();
105     nhapMaLaptop(dsLaptop);
106     nhapTenLaptop();
107 }
108
109 public Laptop(){
110     this.ma_laptop = "";
111     this.ten_laptop = "";
112     reset_dsALL();
113 }

```

```

114
115     public String getMa_laptop() {
116         return this.ma_laptop;
117     }
118
119     public void setMa_laptop(String ma_laptop) {
120         this.ma_laptop = ma_laptop;
121     }
122
123     public String getTen_laptop() {
124         return this.ten_laptop;
125     }
126
127     public void setTen_laptop(String ten_laptop) {
128         this.ten_laptop = ten_laptop;
129     }
130
131     public long getTong_gia() {
132         long tong_gia = 0;
133         for(CPU i : dsCPU) {
134             tong_gia += i.getDon_gia();
135         }
136         for(RAM i : dsRAM) {
137             tong_gia += i.getDon_gia();
138         }
139         for(Screen i : dsScreen) {
140             tong_gia += i.getDon_gia();
141         }
142         for(HardDisk i : dsHardDisk) {
143             tong_gia += i.getDon_gia();
144         }
145         return tong_gia;
146     }
147 }
148

```

2. Class CPU:

```
1 import java.util.Arrays;
2
3 public class CPU {
4     private String ma_cpu; // CPU1, CPU2, CPU3
5     private String loai_cpu; // i3, i5, i7
6     private long don_gia; // 100, 200, 300
7     private String hang_cpu = Lib.arrToString(Lib.read_data("cpu_company"));
8     private final String[] cpu_arr = Lib.read_data("cpu_type");
9     private final String[] cpu_price_arr = Lib.read_data("cpu_price");
10
11     public CPU(){
12         this.ma_cpu = "";
13         this.loai_cpu = "";
14         this.don_gia = 0;
15     }
16
17     public CPU(String ma_cpu, String loai_cpu, String hang_cpu, long don_gia) {
18         this.ma_cpu = ma_cpu;
19         this.hang_cpu = hang_cpu;
20         this.loai_cpu = loai_cpu;
21         this.don_gia = don_gia;
22     }
23
24     public void nhapTTCPU(){
25         nhapTTCPU(0);
26     }
27
28     public void nhapTTCPU(int type_laptop){
29         String[] tmpArr = {};
30         System.out.print(">> CPU:\n");
31         String ma_cpu = "";
32         System.out.println("    "
33             + Lib.getlang("CPU_COMPANY")
34             + ": "
35             + String.valueOf(this.hang_cpu)
36         );
37         while(true){
38             tmpArr = Lib.readDataLaptop(type_laptop)[0].split("/");
39             if(tmpArr[0].toLowerCase().equals("null")) {
40                 System.out.print("    "
41                     + Lib.getlang("INPUT")
42                     + " "
43                     + Lib.getlang("CPU_CODE")
44                     + " "
45                     + String.valueOf(Lib.arrToListString(this.cpu_arr, "CPU", true))
46                     + ": "
47                 );
48                 ma_cpu = Lib.str_input();
49             } else {
50                 if(tmpArr.length < 2) {
51                     ma_cpu = "CPU" + String.valueOf(Lib.getIndexArr(this.cpu_arr,tmpArr[0])+1);
52                     System.out.println("    CPU: "
53                         + ma_cpu
54                         + " - "
55                         + String.valueOf(tmpArr[0])
56                     );
57                 } else {
58                     System.out.print("    "
59                         + Lib.getlang("INPUT")
60                         + " "
61                         + Lib.getlang("CPU_CODE")
62                         + " "
63                         + String.valueOf(Lib.arrToListString(tmpArr, "CPU", true))
64                         + ": "
65                     );
66                 }
67             }
68         }
69     }
70 }
```

```

66         ma_cpu = Lib.str_input();
67         try {
68             if (!Arrays.asList(this.cpu_arr).contains(tmpArr[
69                 Integer.valueOf(ma_cpu.toLowerCase().replace("cpu", ""))
70                 - 1
71             ])) {
72                 Integer.valueOf("KhanhNguyen9872");
73             }
74         } catch (Exception e) {
75             System.out.println("!! " + Lib.getlang("INVALID_CPU"));
76             continue;
77         }
78     }
79 }
80
81     try {
82         this.loai_cpu = cpu_arr[Integer.valueOf(
83             ma_cpu.toLowerCase().replace("cpu", "")
84             )-1];
85     } catch (Exception e) {
86         System.out.println("!! " + Lib.getlang("INVALID_CPU_CODE"));
87         continue;
88     }
89     try {
90         this.don_gia = Long.valueOf(cpu_price_arr[
91             Integer.valueOf(ma_cpu.toLowerCase().replace("cpu", ""))
92             - 1
93             ]
94         );
95     } catch (Exception e) {
96         this.don_gia = Lib.int_input(
97             Lib.getlang("PRICE"),
98             null,
99             true,
100             true,
101             true
102         );
103     }
104
105     this.ma_cpu = ma_cpu;
106     System.out.println("    "
107         + Lib.getlang("PRICE")
108         + ": "
109         + String.valueOf(this.don_gia)
110         + " USD");
111     break;
112 }
113 }
114
115 public String getHang_cpu() {
116     return hang_cpu;
117 }
118
119 public void setHang_cpu(String hang_cpu) {
120     this.hang_cpu = hang_cpu;
121 }
122
123 public String getMa_cpu() {
124     return ma_cpu;
125 }
126
127 public void setMa_cpu(String ma_cpu) {
128     this.ma_cpu = ma_cpu;
129 }
130
131 public String getLoai_cpu() {
132     return loai_cpu;
133 }
134
135 public void setLoai_cpu(String loai_cpu) {
136     this.loai_cpu = loai_cpu;
137 }

```



```

138
139     public long getDon_gia() {
140         return don_gia;
141     }
142
143     public void setDon_gia(long don_gia) {
144         this.don_gia = don_gia;
145     }
146
147     public String export_config() {
148         return String.valueOf(this.ma_cpu)
149             + "/"
150             + String.valueOf(this.hang_cpu)
151             + "/"
152             + String.valueOf(this.loai_cpu)
153             + "/"
154             + String.valueOf(this.don_gia) + ", ";
155     }
156
157     @Override
158     public String toString(){
159         return String.valueOf(this.ma_cpu)
160             + " - "
161             + String.valueOf(this.hang_cpu)
162             + " "
163             + String.valueOf(this.loai_cpu)
164             + " ("
165             + String.valueOf(this.don_gia)
166             + " "
167             + Lib.getlang("MONEY")
168             + ")\n";
169     }
170
171 }
172

```

3. Class RAM:

```

1  import java.util.Arrays;
2
3  public class RAM {
4      private int so_luong; // 1 thanh, 2 thanh
5      private String[] ma_ram; // raml, ram2
6      private String loai_ram; // DDR3, DDR4
7      private long don_gia; // 1G DDR3 100, 1G DDR4 200
8      private int dung_luong; // 1G
9      private final String[] ram_arr = Lib.read_data("ram_type");
10     private final String[] ram_price_arr = Lib.read_data("ram_price");
11
12     public RAM(){
13         this.so_luong = 0;
14         this.ma_ram = null;
15         this.loai_ram = "";
16         this.dung_luong = 0;
17         this.don_gia = 0;
18     }
19
20     public RAM(int so_luong, String[] ma_ram, String loai_ram, int dung_luong, long don_gia) {
21         this.so_luong = so_luong;
22         this.ma_ram = ma_ram;
23         this.loai_ram = loai_ram;
24         this.dung_luong = dung_luong;
25         this.don_gia = don_gia;
26     }

```

```

27
28 public void nhapTTRam(int type_laptop) {
29     String str;
30     System.out.print(">> RAM:\n");
31     int tmp = 0;
32     String[] tmpArr = {};
33     this.so_luong = (int)Lib.int_input(
34         Lib.getlang("NUMBER_OF_RAM"),
35         null,
36         true,
37         true,
38         false
39     );
40     while(true){
41         tmpArr = Lib.readDataLaptop(type_laptop)[1].split("/");
42         if(tmpArr[0].toLowerCase().equals("null")) {
43             tmp = (int)Lib.int_input(
44                 Lib.getlang("RAM_TYPE"),
45                 Lib.arrToListString(
46                     this.ram_arr,
47                     null,
48                     true
49                 ),
50                 true,
51                 true,
52                 false
53             );
54         } else {
55             if(tmpArr.length < 2) {
56                 tmp = Lib.getIndexArr(this.ram_arr, tmpArr[0]);
57                 System.out.println(
58                     " "
59                     + Lib.getlang("RAM_TYPE")
60                     + ": "
61                     + String.valueOf(tmpArr[0])
62                 );
63             } else {
64                 tmp = (int)Lib.int_input(
65                     Lib.getlang("RAM_TYPE"),
66                     Lib.arrToListString(
67                         tmpArr,
68                         null,
69                         true
70                     ),
71                     true,
72                     true,
73                     false
74                 );
75                 try {
76                     if(!Arrays.asList(this.ram_arr).contains(tmpArr[tmp-1])){
77                         Integer.valueOf("KhanhNguyen9872");
78                     }
79                 } catch (Exception e) {
80                     System.out.println(
81                         "!! "
82                         + Lib.getlang("INVALID_RAM")
83                     );
84                     continue;
85                 }
86             }
87         }
88         try {
89             this.loai_ram = this.ram_arr[tmp-1];
90         } catch (Exception e) {
91             System.out.println("!! " + Lib.getlang("INVALID_RAM"));
92             continue;
93         }
94
95         try {
96             this.don_gia = Long.valueOf(this.ram_price_arr[tmp-1]);
97         } catch (Exception e) {
98             this.don_gia = Lib.int_input(
99                 Lib.getlang("PRICE"),

```

```

100         null,
101         true,
102         true,
103         true
104     );
105     }
106     break;
107 }
108 this.dung_luong = (int)Lib.int_input(
109     Lib.getlang("CAPACITY"),
110     "(GB)",
111     true,
112     true,
113     false
114 );
115 this.don_gia = this.don_gia * this.dung_luong;
116 this.ma_ram = new String[so_luong];
117 for(int i=0;i<so_luong; i++) {
118     System.out.print("      "
119         + Lib.getlang("RAM_CODE")
120         + " "
121         + String.valueOf(i+1)
122         + ": "
123     );
124     this.ma_ram[i] = Lib.str_input();
125 }
126 System.out.println("      "
127     + Lib.getlang("PRICE")
128     + ": "
129     + String.valueOf(this.don_gia)
130     + " "
131     + Lib.getlang("MONEY")
132 );
133 System.out.println("      "
134     + Lib.getlang("TOTAL_PRICE")
135     + ": "
136     + String.valueOf(this.don_gia * this.so_luong)
137     + " "
138     + Lib.getlang("MONEY")
139 );
140 }
141
142 public int getSo_luong() {
143     return so_luong;
144 }
145
146 public void setSo_luong(int so_luong) {
147     this.so_luong = so_luong;
148 }
149
150 public String[] getMa_ram() {
151     return ma_ram;
152 }
153
154 public void setMa_ram(String[] ma_ram) {
155     this.ma_ram = ma_ram;
156 }
157
158 public String getLoai_ram() {
159     return loai_ram;
160 }
161
162 public void setLoai_ram(String loai_ram) {
163     this.loai_ram = loai_ram;
164 }
165
166 public long getDon_gia() {
167     return don_gia;
168 }
169

```

```

170     public void setDon_gia(long don_gia) {
171         this.don_gia = don_gia;
172     }
173
174     public int getDung_luong() {
175         return dung_luong;
176     }
177
178     public void setDung_luong(int dung_luong) {
179         this.dung_luong = dung_luong;
180     }
181
182     public String export_config() {
183         String str = "";
184         for(String z : this.getMa_ram()) {
185             str += String.valueOf(z)
186                 + "/"
187                 + String.valueOf(this.loai_ram)
188                 + "/"
189                 + String.valueOf(this.dung_luong)
190                 + "/"
191                 + String.valueOf(this.don_gia)
192                 + ", ";
193         }
194         return str;
195     }
196
197     @Override
198     public String toString(){
199         int count = 1;
200         String str = "";
201         for(String z : this.getMa_ram()) {
202             str += " "
203                 + String.valueOf(count)
204                 + ". "
205                 + String.valueOf(z)
206                 + " - "
207                 + String.valueOf(this.loai_ram)
208                 + " - "
209                 + String.valueOf(this.dung_luong)
210                 + " GB ("
211                 + String.valueOf(this.don_gia)
212                 + " "
213                 + Lib.getlang("MONEY")
214                 + ")";
215             str += "\n";
216             count++;
217         }
218         return str;
219     }
220
221 }
222

```

4. Class Screen:

```
1  import java.util.Arrays;
2
3  public class Screen {
4      private String ma_manhinh;
5      private String hang_sanxuat;
6      private String do_phangiai;
7      private long don_gia;
8      private int chong_loa;
9      private final String[] screen_arr = Lib.read_data("screen_type");
10     private final String[] screen_fullhd_arr = Lib.read_data("screen_FullHD");
11     private final String[] screen_price_arr = Lib.read_data("screen_price");
12
13     public Screen() {
14         this.ma_manhinh = "";
15         this.hang_sanxuat = "";
16         this.do_phangiai = "";
17         this.don_gia = 0;
18         this.chong_loa = 0;
19     }
20
21     public Screen(String ma_manhinh,
22         String hang_sanxuat,
23         String do_phangiai,
24         long don_gia,
25         int chong_loa) {
26         this.ma_manhinh = ma_manhinh;
27         this.hang_sanxuat = hang_sanxuat;
28         this.do_phangiai = do_phangiai;
29         this.don_gia = don_gia;
30         this.chong_loa = chong_loa;
31     }
32
33     public void nhapTTScreen(int type_laptop) {
34         int tmp;
35         String[] tmp2;
36         String[] tmpArr = {};
37         System.out.print(">> " + Lib.getlang("SCREEN") + ": \n");
38         System.out.print("      " + Lib.getlang("SCREEN_CODE") + ": ");
39         this.ma_manhinh = Lib.str_input();
40         System.out.print("      " + Lib.getlang("MANUFACTURER") + ": ");
41         this.hang_sanxuat = Lib.str_input();
42         while(true) {
43             tmpArr = Lib.readDataLaptop(type_laptop)[2].split("/");
44             System.out.print("      " + Lib.getlang("SCREEN_TYPE"));
45             if(tmpArr[0].toLowerCase().equals("null")) {
46                 System.out.print(
47                     " "
48                     + String.valueOf(
49                         Lib.arrToListString(
50                             this.screen_arr,
51                             null,
52                             false
53                         )
54                     )
55                     + ": "
56                 );
57                 this.do_phangiai = Lib.str_input();
58             } else {
59                 if(tmpArr.length < 2) {
60                     this.do_phangiai = String.valueOf(tmpArr[0]);
61                     System.out.println(": " + this.do_phangiai);
62                 } else {
```

```

62     } else {
63         System.out.print(
64             " "
65             + String.valueOf(
66                 Lib.arrToListString(
67                     tmpArr,
68                     null,
69                     false
70                 )
71             + ": "
72         )
73     );
74     this.do_phangiai = Lib.str_input();
75     try {
76         if(!Arrays.asList(this.screen_arr).contains(
77             tmpArr[Lib.getIndexArr(tmpArr, this.do_phangiai)]
78         )){
79             Integer.valueOf("KhanhNguyen9872");
80         }
81     } catch (Exception e) {
82         System.out.println(
83             "!! "
84             + Lib.getlang("SCREEN")
85             + " "
86             + Lib.getlang("IS_NOT_VALID")
87         );
88         continue;
89     }
90 }
91 }
92
93 tmp2 = Lib.read_data("screen_" + this.do_phangiai);
94 if(tmp2[0].toLowerCase().equals("null")) {
95     this.don_gia = Long.valueOf(
96         this.screen_price_arr[
97             Lib.getIndexArr(
98                 this.screen_arr,
99                 this.do_phangiai
100             )
101         ]
102     );
103 } else {
104     while(true) {
105         tmp = (int)Lib.int_input(
106             Lib.getlang("SCREEN_STD"),
107             Lib.arrToListString(tmp2, null, false),
108             true,
109             true,
110             false
111         );
112         try {
113             this.don_gia = Long.valueOf(
114                 this.screen_price_arr[
115                     Lib.getIndexArr(
116                         this.screen_arr,
117                         this.do_phangiai
118                     )
119                 ].split("/") [tmp-1]
120             );
121         } catch (Exception e) {
122             System.out.println(
123                 "!! "
124                 + Lib.getlang("INVALID_SCREEN_STD")
125             );
126             continue;
127         }
128         this.chong_loa = tmp;
129         break;
130     }
131 }

```

```

132         System.out.println(
133             " "
134             + Lib.getlang("PRICE")
135             + ": "
136             + String.valueOf(this.don_gia)
137             + " "
138             + Lib.getlang("MONEY")
139         );
140         break;
141     }
142 }
143
144 public String getMa_manhinh() {
145     return ma_manhinh;
146 }
147
148 public void setMa_manhinh(String ma_manhinh) {
149     this.ma_manhinh = ma_manhinh;
150 }
151
152 public String getHang_sanxuat() {
153     return hang_sanxuat;
154 }
155
156 public void setHang_sanxuat(String hang_sanxuat) {
157     this.hang_sanxuat = hang_sanxuat;
158 }
159
160 public String getDo_phangiai() {
161     return do_phangiai;
162 }
163
164 public void setDo_phangiai(String do_phangiai) {
165     this.do_phangiai = do_phangiai;
166 }
167
168 public long getDon_gia() {
169     return don_gia;
170 }
171
172 public void setDon_gia(long don_gia) {
173     this.don_gia = don_gia;
174 }
175
176 public int getChong_loa() {
177     return chong_loa;
178 }
179
180 public void setChong_loa(int chong_loa) {
181     this.chong_loa = chong_loa;
182 }
183
184 public String export_config() {
185     return String.valueOf(this.ma_manhinh)
186         + "/" + String.valueOf(this.hang_sanxuat)
187         + "/" + String.valueOf(this.do_phangiai)
188         + "/" + String.valueOf(this.chong_loa)
189         + "/" + String.valueOf(this.don_gia);
190 }
191
192 @Override
193 public String toString(){
194     String str = "";
195     str += String.valueOf(this.ma_manhinh)
196         + " - "
197         + String.valueOf(this.hang_sanxuat)
198         + " - "
199         + String.valueOf(this.do_phangiai)
200         + " - "
201         + String.valueOf(this.don_gia)
202     ;
203     if(this.chong_loa != 0) {
204         str += " - " + String.valueOf(this.chong_loa);
205     }
206     str += " ("
207         + String.valueOf(this.don_gia)
208         + " "
209         + Lib.getlang("MONEY")
210         + ")\n";
211     return str;
212 }
213 }
214

```

5. Class HardDisk:

```
1 import java.util.Arrays;
2
3 public class HardDisk {
4     private String kieu_ocung = null;
5     private String ma_ocung;
6     private long don_gia;
7     private int dung_luong;
8     private int so_vong_quay;
9     private int toc_do_doc;
10    private int toc_do_ghi;
11    private final String[] harddisk_arr = Lib.read_data("harddisk_type");
12    private final String[] harddisk_price = Lib.read_data("harddisk_price");
13
14    public HardDisk() {
15        this.ma_ocung = "";
16        this.don_gia = 0;
17        this.dung_luong = 0;
18        this.so_vong_quay = 0;
19        this.toc_do_doc = 0;
20        this.toc_do_ghi = 0;
21    }
22
23    public HardDisk(
24        String ma_ocung,
25        String kieu_ocung,
26        long don_gia,
27        int dung_luong,
28        int so_vong_quay,
29        int toc_do_doc,
30        int toc_do_ghi
31    ) {
32        this.ma_ocung = ma_ocung;
33        this.kieu_ocung = kieu_ocung;
34        this.don_gia = don_gia;
35        this.dung_luong = dung_luong;
36        this.so_vong_quay = so_vong_quay;
37        this.toc_do_doc = toc_do_doc;
38        this.toc_do_ghi = toc_do_ghi;
39    }
40
41    public void set_type_harddisk(int type_laptop){
42        String[] tmpArr;
43        String tmp;
44        tmpArr = Lib.readDataLaptop(type_laptop)[3].split("/");
45        if(tmpArr[0].toLowerCase().equals("null")) {
46            System.out.print(
47                " "
48                + Lib.getlang("HARDDISK_TYPE")
49                + " "
50                + String.valueOf(
51                    Lib.arrToListString(
52                        this.harddisk_arr,
53                        null,
54                        false
55                    )
56                )
57                + ": "
58            );
59            this.kieu_ocung = Lib.str_input();
60        } else {
```



```

61         while(true) {
62             System.out.print("    " + Lib.getlang("HARDDISK_TYPE"));
63             if(tmpArr.length < 2) {
64                 this.kieu_ocung = tmpArr[0];
65                 System.out.println(": " + this.kieu_ocung);
66             } else {
67                 System.out.print(
68                     " "
69                     + String.valueOf(
70                         Lib.arrToListString(
71                             tmpArr,
72                             null,
73                             false
74                         )
75                     + ": "
76                 )
77             );
78             this.kieu_ocung = Lib.str_input().toUpperCase();
79             try {
80                 if(!Arrays.asList(this.harddisk_arr).contains(
81                     tmpArr[Lib.getIndexArr(tmpArr, this.kieu_ocung)]
82                 )){
83                     Integer.valueOf("KhanhNguyen9872");
84                 }
85                 tmp = this.harddisk_price[
86                     Lib.getIndexArr(this.harddisk_arr, this.kieu_ocung)
87                 ];
88             } catch (Exception e) {
89                 System.out.println("!!" + Lib.getlang("INVALID_HARDDISK_TYPE"));
90                 continue;
91             }
92         }
93         break;
94     }
95 }
96
97
98 public void nhapTTHardDisk(int type_laptop) {
99     if(this.kieu_ocung == null) {
100         set_type_harddisk(type_laptop);
101     }
102     String tmp = "";
103     while(true) {
104         try {
105             tmp = this.harddisk_price[
106                 Lib.getIndexArr(this.harddisk_arr, this.kieu_ocung)
107             ];
108         } catch (Exception e) {
109             set_type_harddisk(type_laptop);
110             continue;
111         }
112         break;
113     }
114     System.out.print("    " + Lib.getlang("HARDDISK_CODE") + ": ");
115     this.ma_ocung = Lib.str_input();
116     this.dung_luong = (int)Lib.int_input(
117         Lib.getlang("HARDDISK_CAPACITY"),
118         "(GB)",
119         true,
120         true,
121         false
122     );
123
124     if(tmp.toLowerCase().equals("null")) {
125         this.don_gia = Lib.int_input(
126             Lib.getlang("PRICE"),
127             "(USD)",
128             true,
129             true,
130             true
131         );

```

```

132     } else {
133         this.don_gia = Long.valueOf(tmp);
134     }
135     if(this.kieu_ocung.equals("HDD")) {
136         this.so_vong_quay = (int)Lib.int_input(
137             Lib.getlang("NUMBER_ROTATIONS"),
138             "(rpm)",
139             true,
140             true,
141             false
142         );
143     } else if(this.kieu_ocung.equals("SSD")) {
144         this.toc_do_doc = (int)Lib.int_input(
145             Lib.getlang("READ_SPEED"),
146             "(mb/s)",
147             true,
148             true,
149             false
150         );
151         this.toc_do_ghi = (int)Lib.int_input(
152             Lib.getlang("WRITE_SPEED"),
153             "(mb/s)",
154             true,
155             true,
156             false
157         );
158     }
159 }
160
161 public String getKieu_ocung() {
162     return kieu_ocung;
163 }
164
165 public void setKieu_ocung(String kieu_ocung) {
166     this.kieu_ocung = kieu_ocung;
167 }
168
169 public String getMa_ocung() {
170     return ma_ocung;
171 }
172
173 public void setMa_ocung(String ma_ocung) {
174     this.ma_ocung = ma_ocung;
175 }
176
177 public long getDon_gia() {
178     return don_gia;
179 }
180
181 public void setDon_gia(long don_gia) {
182     this.don_gia = don_gia;
183 }
184
185 public int getDung_luong() {
186     return dung_luong;
187 }
188
189 public void setDung_luong(int dung_luong) {
190     this.dung_luong = dung_luong;
191 }
192
193 public int getSo_vong_quay() {
194     return so_vong_quay;
195 }
196
197 public void setSo_vong_quay(int so_vong_quay) {
198     this.so_vong_quay = so_vong_quay;
199 }
200

```

```

201 public int getToc_do_doc() {
202     return toc_do_doc;
203 }
204
205 public void setToc_do_doc(int toc_do_doc) {
206     this.toc_do_doc = toc_do_doc;
207 }
208
209 public int getToc_do_ghi() {
210     return toc_do_ghi;
211 }
212
213 public void setToc_do_ghi(int toc_do_ghi) {
214     this.toc_do_ghi = toc_do_ghi;
215 }
216
217 public String export_config() {
218     return String.valueOf(this.ma_ocung)
219         + "/" + String.valueOf(this.kieu_ocung)
220         + "/" + String.valueOf(this.dung_luong)
221         + "/" + String.valueOf(this.toc_do_doc)
222         + "/" + String.valueOf(this.toc_do_ghi)
223         + "/" + String.valueOf(this.so_vong_quay)
224         + "/" + String.valueOf(this.don_gia) + ", ";
225 }
226
227 @Override
228 public String toString(){
229     String str = "";
230     str += String.valueOf(this.ma_ocung)
231         + " - " + String.valueOf(this.kieu_ocung)
232         + " ("
233         + String.valueOf(this.dung_luong)
234         + " GB)";
235     if(this.kieu_ocung.equals("HDD")) {
236         str += " - ["
237             + String.valueOf(this.so_vong_quay)
238             + "rpm]";
239     } else if(this.kieu_ocung.equals("SSD")) {
240         str += " - [R: "
241             + String.valueOf(this.toc_do_doc)
242             + " MB/s - W: "
243             + String.valueOf(this.toc_do_ghi) + " MB/s]";
244     }
245     str += " ("
246         + String.valueOf(this.don_gia)
247         + " "
248         + Lib.getlang("MONEY")
249         + ")\n";
250     return str;
251 }
252 }

```

6. Class Quanly:

```
1  import java.io.IOException;
2  import java.util.ArrayList;
3
4  public class Quanly {
5      private final String conf_path = Lib.read_config("conf_path");
6      private ArrayList<Laptop> dsLaptop;
7      public Quanly(){
8          this.dsLaptop = new ArrayList<Laptop>();
9      }
10     public void themLaptop(int so_luong){
11         Laptop newlaptop = null;
12         for(int i=0; i<so_luong; i++) {
13             System.out.println("-----");
14             newlaptop = new Laptop();
15             // Laptop
16             System.out.print(
17                 ">> "
18                 + String.valueOf(Lib.getlang("ENTER_PARAMETERS_OF_LAPTOP"))
19                 + " "
20                 + String.valueOf(i+1)
21                 + ":\n"
22             );
23             newlaptop.nhapTTLaptop(this.dsLaptop);
24             // CPU
25             CPU newCPU = new CPU();
26             newCPU.nhapTTCPU(newlaptop.getDong_laptop());
27             newlaptop.dsCPU.add(newCPU);
28             // RAM
29             RAM newRAM = new RAM();
30             newRAM.nhapTTRam(newlaptop.getDong_laptop());
31             newlaptop.dsRAM.add(newRAM);
32             // Screen
33             Screen newScreen = new Screen();
34             newScreen.nhapTTScreen(newlaptop.getDong_laptop());
35             newlaptop.dsScreen.add(newScreen);
36             // HardDisk
37             int n = (int)Lib.int_input(
38                 ">> " + Lib.getlang("NUMBER_OF_HARDDISK"),
39                 null,
40                 false,
41                 true,
42                 false
43             );
44             for(int ii = 0; ii<n; ii++){
45                 System.out.println(
46                     ">> "
47                     + Lib.getlang("HARDDISK")
48                     + " "
49                     + String.valueOf(ii+1)
50                     + ":\n"
51                 );
52                 HardDisk newHardDisk = new HardDisk();
53                 newHardDisk.nhapTTHardDisk(newlaptop.getDong_laptop());
54                 newlaptop.dsHardDisk.add(newHardDisk);
55             }
56             //
57             if(Lib.yes_or_no(Lib.getlang("SAVE").toLowerCase())){
58                 this.dsLaptop.add(newlaptop);
59                 System.out.print(
60                     ">> "
61                     + Lib.getlang("SAVE_COMPLETED")
62                     + "\n"
63                 );
64             }
65         }
66     }
67 }
```

```

65     }
66 }
67
68 private void menu_edit(Laptop i) {
69     System.out.println(
70         " >> Laptop: "
71         + String.valueOf(i.getMa_laptop())
72         + " - "
73         + String.valueOf(i.getTen_laptop())
74     );
75
76     System.out.println(" " + Lib.getlang("SEL_ITEM_EDIT"));
77     System.out.println(
78         "1. "
79         + Lib.getlang("LAPTOP_SERIES")
80         + " ("
81         + String.valueOf(i.getnameDong_laptop())
82         + ")"
83     );
84     System.out.println(
85         "2. "
86         + Lib.getlang("LAPTOP_CODE")
87         + " ("
88         + String.valueOf(i.getMa_laptop())
89         + ")"
90     );
91     System.out.println(
92         "3. "
93         + Lib.getlang("LAPTOP_NAME")
94         + " ("
95         + String.valueOf(i.getTen_laptop())
96         + ")"
97     );
98     System.out.print("4. " + Lib.getlang("CPU"));
99     if(i.dsCPU.isEmpty()) System.out.print(" (NO DATA)");
100    System.out.print("\n5. " + Lib.getlang("RAM"));
101    if(i.dsRAM.isEmpty()) System.out.print(" (NO DATA)");
102    System.out.print("\n6. " + Lib.getlang("SCREEN"));
103    if(i.dsScreen.isEmpty()) System.out.print(" (NO DATA)");
104    System.out.print("\n7. " + Lib.getlang("HARDDISK"));
105    if(i.dsHardDisk.isEmpty()) System.out.print(" (NO DATA)");
106    System.out.println("\n0. " + Lib.getlang("BACK"));
107 }
108
109 public void suaTTLaptop() {
110     if(Lib.isNotEmpty(this.dsLaptop)) {
111         Lib.printAll(this.dsLaptop);
112         System.out.println("-----");
113         System.out.print(
114             Lib.getlang("INPUT")
115             + " "
116             + Lib.getlang("LAPTOP_CODE")
117             + ": "
118         );
119         int[] info = Lib.kt_ma_laptop(
120             this.dsLaptop,
121             Lib.str_input()
122         );
123         if(info[0] == 0){
124             System.out.print(
125                 "!! "
126                 + Lib.getlang("LAPTOP_CODE_NOT_EXIST")
127                 + "\n"
128             );
129             Lib.pause_console();
130             return;
131         }
132     }

```

```

132 Laptop oldlaptop = this.dsLaptop.get(info[1]);
133 Laptop newlaptop = null;
134 int n;
135 while(true) {
136     newlaptop = new Laptop();
137     Lib.clear_console();
138     menu_edit(oldlaptop);
139     n = (int)Lib.int_input(
140         Lib.getlang("CHOOSE"),
141         null,
142         false,
143         false,
144         false
145     );
146     Lib.clear_console();
147     switch(n) {
148         case 1:
149             System.out.println(
150                 "!! "
151                 + Lib.getlang("WARNING")
152                 + ": "
153                 + Lib.getlang("WARNING001")
154             );
155             System.out.println(
156                 ">> "
157                 + Lib.getlang("LAPTOP_SERIES")
158                 + " "
159                 + Lib.getlang("CURRENT").toLowerCase()
160                 + ": "
161                 + String.valueOf(oldlaptop.getNameDong_laptop())
162             );
163             newlaptop.nhapDongLaptop();
164             if(Lib.yes_or_no(Lib.getlang("SAVE").toLowerCase())){
165                 if(oldlaptop.getDong_laptop() != newlaptop.getDong_laptop()) {
166                     oldlaptop.setDong_laptop(newlaptop.getDong_laptop());
167                     oldlaptop.reset_dsALL();
168                     System.out.print(
169                         " !! "
170                         + Lib.getlang("WARNING")
171                         + ": "
172                         + Lib.getlang("RESET_INFORMATION_COMPLETED")
173                         + "!\n"
174                     );
175                 }
176                 System.out.print(
177                     ">> "
178                     + Lib.getlang("SAVE_COMPLETED")
179                     + "\n"
180                 );
181                 Lib.pause_console();
182             }
183             break;
184         case 2:
185             System.out.println(
186                 ">> "
187                 + Lib.getlang("LAPTOP_CODE")
188                 + " "
189                 + Lib.getlang("CURRENT").toLowerCase()
190                 + ": "
191                 + String.valueOf(oldlaptop.getMa_laptop())
192             );
193             newlaptop.nhapMaLaptop(this.dsLaptop);
194             if(Lib.yes_or_no(Lib.getlang("SAVE").toLowerCase())){
195                 oldlaptop.setMa_laptop(newlaptop.getMa_laptop());
196                 System.out.print(
197                     ">> "
198                     + Lib.getlang("SAVE_COMPLETED")
199                     + "\n"
200                 );
201                 Lib.pause_console();
202             }
203             break;

```

```

204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276

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))
    );

```



```

277     Screen newScreen = new Screen();
278     newScreen.nhapTTScreen(oldlaptop.getDong_laptop());
279     newlaptop.dsScreen.add(newScreen);
280     if(Lib.yes_or_no(Lib.getlang("SAVE").toLowerCase())){
281         oldlaptop.dsScreen = newlaptop.dsScreen;
282         System.out.print(
283             ">> "
284             + Lib.getlang("SAVE_COMPLETED")
285             + "\n"
286         );
287         Lib.pause_console();
288     }
289     break;
290 case 7:
291     System.out.print(
292         ">> "
293         + Lib.getlang("HARDDISK")
294         + " "
295         + Lib.getlang("CURRENT").toLowerCase()
296         + ": "
297         + String.valueOf(Lib.gettext_cpu(oldlaptop, false))
298     );
299     n = (int)Lib.int_input(
300         ">> Số lượng ổ cứng",
301         null,
302         false,
303         true,
304         false
305     );
306     for(int ii = 0; ii<n; ii++){
307         System.out.println(
308             ">> Ổ cứng "
309             + String.valueOf(ii+1)
310             + ":"
311         );
312         HardDisk newHardDisk = new HardDisk();
313         newHardDisk.nhapTTHardDisk(oldlaptop.getDong_laptop());
314         newlaptop.dsHardDisk.add(newHardDisk);
315     }
316     if(Lib.yes_or_no(Lib.getlang("SAVE").toLowerCase())){
317         oldlaptop.dsHardDisk = newlaptop.dsHardDisk;
318         System.out.print(
319             ">> "
320             + Lib.getlang("SAVE_COMPLETED")
321             + "\n"
322         );
323         Lib.pause_console();
324     }
325     break;
326 case 0:
327     return;
328 default:
329     continue;
330 }
331 }
332 } else {
333     System.out.println("!! " + Lib.getlang("LIST_EMPTY"));
334 }
335 }
336
337 public void xoaTTLaptop() {
338     if(Lib.isNotEmpty(this.dsLaptop)) {
339         while(true) {
340             Lib.clear_console();
341             System.out.print(
342                 "1. "
343                 + Lib.getlang("DEL_WITH_LAPTOP_CODE")
344                 + "\n2. "
345                 + Lib.getlang("DEL_ALL_LAPTOP")
346                 + "\n0. "
347                 + Lib.getlang("BACK")
348                 + "\n"
349             );

```



```

350 switch((int)
351     Lib.int_input(
352         Lib.getlang(
353             "CHOOSE"),
354         null,
355         false,
356         false,
357         false
358     )
359 ) {
360 case 1:
361     Lib.clear_console();
362     Lib.printAll(this.dsLaptop);
363     System.out.print(
364         Lib.getlang("INPUT")
365         + " "
366         + Lib.getlang("LAPTOP_CODE")
367         + ": "
368     );
369     int[] info = Lib.kt_ma_laptop(
370         this.dsLaptop,
371         Lib.str_input()
372     );
373     if(info[0] == 0){
374         System.out.print(
375             "!! "
376             + Lib.getlang("INVALID_LAPTOP_CODE")
377             + "\n"
378         );
379         return;
380     }
381     System.out.println("-----");
382     System.out.print(
383         Lib.getlang("LAPTOP_NAME")
384         + ": "
385         + String.valueOf(
386             this.dsLaptop.get(info[1]).getTen_laptop()
387         )
388         + "\n"
389     );
390     if(Lib.yes_or_no(Lib.getlang("DEL_LAPTOP"))){
391         ArrayList<Laptop> dsTam = new ArrayList<Laptop>();
392         int count = 0;
393         for(Laptop i : this.dsLaptop) {
394             if(count == info[1]){
395                 count++;
396                 continue;
397             }
398             dsTam.add(i);
399             count++;
400         }
401         this.dsLaptop = dsTam;
402         System.out.print(
403             ">>> "
404             + Lib.getlang("DEL_COMPLETED")
405             + "\n"
406         );
407     }
408     break;
409 case 2:
410     Lib.clear_console();
411     Lib.printAll(this.dsLaptop);
412     if(Lib.yes_or_no(Lib.getlang("DEL_ALL_LAPTOP"))){
413         this.dsLaptop = new ArrayList<Laptop>();
414         System.out.print(
415             ">>> "
416             + Lib.getlang("DEL_ALL_COMPLETED")
417             + "\n"
418         );
419     }
420     break;
421 case 0:
422     return;
423 default:
424     continue;
425 }
426 }

```

```

427     } else {
428         System.out.println("!! " + Lib.getlang("LIST_EMPTY"));
429     }
430 }
431
432 public void xemTTLaptop(){
433     String full_str = Lib.gettext(this.dsLaptop, false);
434     if(full_str == null) {
435         return;
436     } else {
437         System.out.print(full_str);
438     }
439 }
440
441 public void xuatT2File() throws IOException{
442     if(Lib.isEmpty(this.dsLaptop)) {
443         Lib.fwrite("ds_laptop.txt", Lib.gettext(this.dsLaptop, false), false);
444         System.out.print(">> " + Lib.getlang("EXPORT_COMPLETED") + " [ds_laptop.txt]\n");
445     } else {
446         System.out.println("!! " + Lib.getlang("LIST_EMPTY"));
447     }
448 }
449
450 public void nhapCONF() {
451     int count = 1;
452     int count2;
453     Laptop newLaptop = null;
454     String name = "";
455     String tmpStr = "";
456     String[] tmp = {};
457     String[] tmp2 = {};
458     String[] list_target = {
459         "series",
460         "code",
461         "name",
462         "cpu",
463         "ram",
464         "screen",
465         "harddisk"
466     };
467     System.out.println(">> Các file data tại [" + this.conf_path + "]\n");
468     for(String s : Lib.listdir(this.conf_path)) {
469         if(s.contains(".conf")) {
470             System.out.println("=> " + s.substring(0, s.length() - 5));
471         }
472     }
473     System.out.println("-----");
474     // input file name conflatopbut
475     String str = this.conf_path
476         + Lib.str_input(
477             Lib.getlang("INPUT_FILE_NAME")
478             + ": "
479         )
480         + ".conf";
481     tmp = Lib.read_data(str, "CONFIG", false, false);
482     if(tmp == null) {
483         System.out.println("!! " + Lib.getlang("DATA_ERROR"));
484         return;
485     } else if(!tmp[0].equals("Hoa_Don_Laptop")) {
486         System.out.println("!! " + Lib.getlang("DATA_ERROR"));
487         return;
488     }
489     System.out.println(">> " + Lib.getlang("PROCESS_DATA") + "....");
490     while(true) {
491         newLaptop = new Laptop();
492         for(String target : list_target) {
493             name = target + String.valueOf(count);
494             tmp = Lib.read_data(str, name, false, true);
495             if(tmp == null) {
496                 System.out.println(
497                     "\n>>"
498                     + Lib.getlang("IMPORT_CONFIG_COMPLETED")
499                 );
500                 return;
501             }

```

```

502 if(tmp.length <= 0) continue;
503 // series
504 if(target.equals(list_target[0])) {
505     newLaptop.setDong_laptop(
506         Lib.getIndexArr(
507             Lib.read_data("laptop_type"),
508             tmp[0]
509         )
510         + 1
511     );
512 }
513 // code
514 if(target.equals(list_target[1])) {
515     tmpStr = Lib.arrToString(tmp);
516     int[] info = Lib.kt_ma_laptop(this.dsLaptop, tmpStr);
517     if(info[0]==1){
518         System.out.println(
519             "!! "
520             + Lib.getlang("LAPTOP_CODE_EXIST")
521             + " ["
522             + Lib.getlang("LAPTOP_CODE")
523             + ": "
524             + tmpStr
525             + "]"
526         );
527         if(Lib.yes_or_no(
528             Lib.getlang("CHANGE_LAPTOP_CODE").toLowerCase()
529         )) {
530             newLaptop.nhapMaLaptop(this.dsLaptop);
531             continue;
532         } else {
533             return;
534         }
535     } else {
536         System.out.println(
537             " - "
538             + Lib.getlang("LAPTOP_CODE")
539             + ": "
540             + tmpStr
541         );
542     }
543     newLaptop.setMa_laptop(tmpStr);
544 }
545 // name
546 if(target.equals(list_target[2])) {
547     newLaptop.setTen_laptop(Lib.arrToString(tmp));
548 }
549 // cpu
550 if(target.equals(list_target[3])) {
551     for(String cpu : tmp) {
552         if(cpu.isBlank()) continue;
553         tmp2 = cpu.split("/");
554         CPU newCPU = new CPU(
555             tmp2[0],
556             tmp2[1],
557             tmp2[2],
558             Long.valueOf(tmp2[3])
559         );
560         newLaptop.dsCPU.add(newCPU);
561     }
562 }
563 // ram
564 if(target.equals(list_target[4])) {
565     int len_ram = 0;
566     for(String ram : tmp) {
567         if(ram.isBlank()) continue;
568         len_ram++;
569     }
570     String[] ma_ram = new String[len_ram];
571     count2 = 0;
572     for(String ram : tmp) {
573         if(ram.isBlank()) continue;
574         tmp2 = ram.split("/");
575         ma_ram[count2] = tmp2[0];
576         count2++;
577     }

```

```

578         RAM newRAM = new RAM(
579             len_ram,
580             ma_ram,
581             tmp2[1],
582             Integer.valueOf(tmp2[2]),
583             Long.valueOf(tmp2[3])
584         );
585         newLaptop.dsRAM.add(newRAM);
586     }
587     // Screen
588     if(target.equals(list_target[5])) {
589         for(String screen : tmp) {
590             if(screen.isBlank()) continue;
591             tmp2 = screen.split("/");
592             Screen newScreen = new Screen(
593                 tmp2[0],
594                 tmp2[1],
595                 tmp2[2],
596                 Long.valueOf(tmp2[4]),
597                 Integer.valueOf(tmp2[3])
598             );
599             newLaptop.dsScreen.add(newScreen);
600         }
601     }
602     // HardDisk
603     if(target.equals(list_target[6])) {
604         for(String harddisk : tmp) {
605             if(harddisk.isBlank()) continue;
606             tmp2 = harddisk.split("/");
607             HardDisk newHardDisk = new HardDisk(
608                 tmp2[0],
609                 tmp2[1],
610                 Long.valueOf(tmp2[6]),
611                 Integer.valueOf(tmp2[2]),
612                 Integer.valueOf(tmp2[5]),
613                 Integer.valueOf(tmp2[3]),
614                 Integer.valueOf(tmp2[4])
615             );
616             newLaptop.dsHardDisk.add(newHardDisk);
617         }
618     }
619 }
620 this.dsLaptop.add(newLaptop);
621 count++;
622 }
623 }
624
625 public void xuatCONF() throws IOException {
626     if(Lib.isEmpty(this.dsLaptop)) {
627         String str = this.conf_path
628             + Lib.str_input(
629                 Lib.getlang("INPUT_FILE_NAME")
630                 + ": "
631             )
632             + ".conf";
633         Lib.fwrite(str,
634             "CONFIG = Hoa_Don_Laptop"
635             + "\n"
636             + Lib.gettext(this.dsLaptop, true),
637             false
638         );
639         System.out.print(
640             ">> "
641             + Lib.getlang("EXPORT_CONFIG_COMPLETED")
642             + " ["
643             + str
644             + "]\n"
645         );
646     } else {
647         System.out.println("!! " + Lib.getlang("LIST_EMPTY"));
648     }
649 }
650

```

```

651 public static void menu(){
652     System.out.print(" >> " + Lib.getlang("MAIN_PROGRAM") + " <<\n\n");
653     System.out.print("1. " + Lib.getlang("ADD_LAPTOP_INFO") + "\n");
654     System.out.print("2. " + Lib.getlang("EDIT_LAPTOP_INFO") + "\n");
655     System.out.print("3. " + Lib.getlang("VIEW_LAPTOP_INFO") + "\n");
656     System.out.print("4. " + Lib.getlang("DEL_LAPTOP_INFO") + "\n");
657     System.out.print("5. " + Lib.getlang("EXPORT_LAPTOP_INFO") + "\n");
658     System.out.print("6. " + Lib.getlang("IMPORT_LAPTOP_CONFIG") + "\n");
659     System.out.print("7. " + Lib.getlang("EXPORT_LAPTOP_CONFIG") + "\n");
660     System.out.print("8. " + Lib.getlang("EXIT_PROGRAM") + "\n\n");
661 }
662 }
663

```

7. Class Lib

[illegible]

```

85         System.out.println(
86             "!! "
87             + Lib.getlang("ENTER_NUM_PL2")
88         );
89         Lib.keyboard.next();
90         continue;
91     }
92     if(var<=0 && greater_than_zero) {
93         System.out.println(
94             "!! "
95             + name
96             + " "
97             + Lib.getlang("IS_NOT_VALID")
98         );
99         continue;
100     }
101     break;
102 }
103 return var;
104 }
105
106 protected static int[] kt_ma_laptop(
107     ArrayList<Laptop> dsLaptop,
108     String ma_laptop
109 ) {
110     int count = 0;
111     int[] info = {0, 0};
112     for (Laptop i: dsLaptop){
113         if(i.getMa_laptop().equals(ma_laptop)){
114             info[0] = 1;
115             info[1] = count;
116         }
117         count++;
118     }
119     return info;
120 }
121
122 protected static void printAll(ArrayList<Laptop> dsLaptop){
123     int count = 0;
124     System.out.println(
125         ">>> "
126         + Lib.getlang("LIST_LAPTOP")
127         + ": "
128     );
129     for(Laptop i : dsLaptop) {
130         System.out.println(" "
131             + String.valueOf(count+1)
132             + ". "
133             + String.valueOf(i.getMa_laptop())
134         );
135         count++;
136     }
137 }
138
139 protected static String gettext_cpu(Laptop i, boolean is_conf) {
140     if (i.dsCPU.size() == 0) {
141         if (is_conf) {
142             return "";
143         }
144         return Lib.getlang("NO_INFORMATION")
145             + " "
146             + Lib.getlang("CPU")
147             + "\n";
148     }
149     String str = "";
150     for(CPU j : i.dsCPU) {
151         if(is_conf) {
152             str += j.export_config();
153         } else {
154             str += j;
155         }
156     }
157     return str;
158 }
159
160 protected static String gettext_ram(Laptop i, boolean is_conf) {
161     if (i.dsRAM.size() == 0) {
162         if (is_conf) {
163             return "";
164         }
165         return Lib.getlang("NO_INFORMATION")
166             + " "
167             + Lib.getlang("RAM")
168             + "\n";

```

```

169     }
170     String str;
171     if(is_conf) str = "";
172     else str = "\n";
173     for(RAM j : i.dsRAM) {
174         if(is_conf) {
175             str += j.export_config();
176         } else {
177             str += j;
178         }
179     }
180     return str;
181 }
182
183 protected static String gettext_screen(Laptop i, boolean is_conf){
184     if (i.dsScreen.size() == 0) {
185         if (is_conf) {
186             return "";
187         }
188         return Lib.getlang("NO_INFORMATION")
189             + " "
190             + Lib.getlang("SCREEN")
191             + "\n";
192     }
193     String str = "";
194     for(Screen j : i.dsScreen) {
195         if(is_conf) {
196             str += j.export_config();
197         } else {
198             str += j;
199         }
200     }
201     return str;
202 }
203
204 protected static String gettext_harddisk(Laptop i, boolean is_conf) {
205     if (i.dsHardDisk.size() == 0) {
206         if (is_conf) {
207             return "";
208         }
209         return Lib.getlang("NO_INFORMATION")
210             + " "
211             + Lib.getlang("HARDDISK")
212             + "\n";
213     }
214     int count = 1;
215     String str;
216     if(is_conf) str = "";
217     else str = "\n";
218     for(HardDisk j : i.dsHardDisk) {
219         if(is_conf) {
220             str += j.export_config();
221         } else {
222             str += " " + String.valueOf(count) + ". " + j;
223         }
224         count++;
225     }
226     return str;
227 }
228
229 protected static String gettext(
230     ArrayList<Laptop> dsLaptop,
231     boolean is_conf
232 ){
233     if(isNotEmpty(dsLaptop)) {
234         String full_str = "";
235         int count = 1;
236         for (Laptop i: dsLaptop) {
237             if(!is_conf) {
238                 full_str += "-----\n";
239             } else {
240                 full_str += "# Config by KhanhNguyen9872\n\n";
241             }
242
243             // number laptop
244             if(!is_conf) {
245                 full_str +=
246                     " >> "
247                     + Lib.getlang("NUM_OF_LAPTOP")
248                     + " "
249                     + String.valueOf(count)
250                     + "\n";
251             }

```



```

252
253 // laptop series
254 if(!is_conf) {
255     full_str += "- " + Lib.getlang("LAPTOP_SERIES") + ": ";
256 } else {
257     full_str += "series" + String.valueOf(count) + " = ";
258 }
259 full_str += String.valueOf(i.getNameDong_laptop()) + "\n";
260
261 // laptop code
262 if(!is_conf) {
263     full_str += "- " + Lib.getlang("LAPTOP_CODE") + ": ";
264 } else {
265     full_str += "code" + String.valueOf(count) + " = ";
266 }
267 full_str += String.valueOf(i.getMa_laptop()) + "\n";
268
269 // laptop name
270 if(!is_conf) {
271     full_str += "- " + Lib.getlang("LAPTOP_NAME") + ": ";
272 } else {
273     full_str += "name" + String.valueOf(count) + " = ";
274 }
275 full_str += String.valueOf(i.getTen_laptop()) + "\n";
276
277 // CPU
278 if(!is_conf) {
279     full_str += "- " + Lib.getlang("CPU") + ": ";
280 } else {
281     full_str += "cpu" + String.valueOf(count) + " = ";
282 }
283 full_str += gettext_cpu(i, is_conf);
284
285 // RAM
286 if(!is_conf) {
287     full_str += "- " + Lib.getlang("RAM") + ": ";
288 } else {
289     full_str += "\nram" + String.valueOf(count) + " = ";
290 }
291 full_str += gettext_ram(i, is_conf);
292
293 // SCREEN
294 if(!is_conf) {
295     full_str += "- " + Lib.getlang("SCREEN") + ": ";
296 } else {
297     full_str += "\nscreen" + String.valueOf(count) + " = ";
298 }
299 full_str += gettext_screen(i, is_conf);
300
301 // HARDDISK
302 if(!is_conf) {
303     full_str += "- " + Lib.getlang("HARDDISK") + ": ";
304 } else {
305     full_str += "\nharddisk" + String.valueOf(count) + " = ";
306 }
307 full_str += gettext_harddisk(i, is_conf);
308
309 full_str += "\n";
310 if(!is_conf) {
311     full_str +=
312         ">> "
313         + Lib.getlang("TOTAL_PRICE")
314         + ": "
315         + String.valueOf(i.getTong_gia())
316         + " "
317         + Lib.getlang("MONEY")
318         + "\n";
319 } else {
320     full_str += "\n";
321 }
322 count++;
323 }
324 if(!is_conf) {
325     full_str += "-----\n";
326 }
327 return full_str;
328 } else {
329     System.out.println("!! " + Lib.getlang("LIST_EMPTY"));
330 }
331 return null;
332 }
333

```

```

334 private static boolean is_load(String file) {
335     for(String s : Lib.is_load_path) {
336         try {
337             if(s.equals(file)) {
338                 return true;
339             }
340         } catch (Exception e) {
341             return false;
342         }
343     }
344     return false;
345 }
346
347 protected static void load_language() {
348     Language();
349 }
350
351 protected static String getlang(String code) {
352     if(current_lang == null) {
353         load_language();
354     }
355     boolean a = false;
356     String str = "";
357     for(String s : Lib.read_data(current_path_lang, code, true, false)) {
358         if(a) {
359             str += ",";
360         }
361         str += s;
362         a = true;
363     }
364     return str;
365 }
366
367 protected static String read_config(String target) {
368     return Lib.arrToString(read_data(FILE_CONFIG, target, true, false));
369 }
370
371 protected static String[] read_data(String target) {
372     return read_data(Lib.read_config("data_path"), target, true, false);
373 }
374
375 protected static String[] read_data(
376     String file, String target,
377     boolean is_exit, boolean is_err_null
378 ) {
379     Properties tmp_properties = new Properties();
380     if(!is_load(file)) {
381         if(Lib.loaded >= Lib.MAX_DATA) {
382             System.out.println(
383                 "\nCannot load data! Program limited "
384                 + Lib.MAX_DATA
385                 + " data!\n"
386             );
387             Lib.pause_console();
388             return null;
389         }
390         try {
391             Lib.inputStream = new FileInputStream(new File(file));
392             tmp_properties.load(
393                 new InputStreamReader(
394                     Lib.inputStream,
395                     Charset.forName("UTF-8")
396                 )
397             );
398         } catch (Exception e) {
399             if(!is_err_null) {
400                 System.out.println(
401                     "!! "
402                     + Lib.getlang("READ_FAILED")
403                     + ": "
404                     + String.valueOf(file)
405                 );
406             }
407             if(is_exit) {
408                 System.exit(1);
409             } else {
410                 return null;
411             }
412         }
413         Lib.properties.add(tmp_properties);
414         Lib.is_load_path[Lib.loaded] = file;
415         Lib.loaded++;
416     }
417 }

```

```

418     try {
419         if(target == null) {
420             return null;
421         } else {
422             return rmSpaceArr(
423                 rmBlankArr(
424                     Lib.properties.get(
425                         getIndexArr(
426                             Lib.is_load_path,
427                             file
428                         )
429                     ).getProperty(target).split(",")
430                 )
431             );
432         }
433     } catch (Exception e) {
434         if(!is_err_null) {
435             System.out.println(
436                 "\n!! Missing data ["
437                 + String.valueOf(target)
438                 + "] from ["
439                 + String.valueOf(file)
440                 + "] !!\n"
441             );
442         }
443         if(is_exit) {
444             System.exit(1);
445         } else {
446             return null;
447         }
448     }
449     return null;
450 }
451
452 protected static String[] rmBlankArr(String[] arr) {
453     int len = arr.length, count = 0;
454     String[] newArr = null;
455     for(String s : arr) {
456         if(s.isBlank()) {
457             len -= 1;
458         }
459     }
460     newArr = new String[len];
461     for(String s : arr) {
462         if(!s.isBlank()) {
463             newArr[count] = s;
464             count++;
465         }
466     }
467     return newArr;
468 }
469
470 protected static String[] rmSpaceArr(String[] arr) {
471     int count = 0;
472     for(String s : arr) {
473         arr[count] = rmSpace(s);
474         count++;
475     }
476     return arr;
477 }
478
479 protected static int getIndexArr(String[] str, String target) {
480     int count = 0;
481     for(String s : str) {
482         if(s.equals(target)) {
483             return count;
484         }
485         count++;
486     }
487     return -1;
488 }
489
490 protected static String rmSpace(String s) {
491     if (s.charAt(0) == ' ') {
492         s = s.substring(1, s.length());
493     }
494     if (s.charAt(s.length()-1) == ' ') {
495         s = s.substring(0, s.length()-1);
496     }
497     return s;
498 }
499

```

```

500     protected static String[] readDataLaptop(int type_laptop){
501         String[] Empty = {};
502         int count = 1;
503         for(String s : read_data("laptop_type")) {
504             if(count == type_laptop) {
505                 return read_data(s + "_data");
506             }
507             count++;
508         }
509         return Empty;
510     }
511
512     protected static String arrToString(String[] arr) {
513         return arrToString(arr, ",");
514     }
515
516     protected static String arrToString(String[] arr, String var) {
517         if(arr == null) return null;
518         String tmpStr = "";
519         for(String ss : arr) {
520             if(!tmpStr.isBlank()) tmpStr += var;
521             tmpStr += ss;
522         }
523         return tmpStr;
524     }
525
526     protected static String arrToListString(
527         String[] arr,
528         String name,
529         boolean is_id
530     ) {
531         if(name == null) {
532             name = "";
533         }
534         int count = 1;
535         String str = "(";
536         for(String s : arr) {
537             if(count > 1) {
538                 str += ", ";
539             }
540             if(is_id) {
541                 str += name
542                     + String.valueOf(count)
543                     + " - ";
544             }
545             str += String.valueOf(s);
546             count++;
547         }
548         str += ")";
549         return str;
550     }
551
552     public static String[] lsdir(String path) {
553         String[] lstFile = null;
554         File folder = new File(path);
555         File[] lstOF = folder.listFiles();
556         int len = lstOF.length;
557
558         for (int i = 0; i < lstOF.length; i++) {
559             if (!lstOF[i].isFile()) {
560                 len -= 1;
561             }
562         }
563
564         lstFile = new String[len];
565         int count = 0;
566
567         for(int i = 0; i < lstOF.length; i++) {
568             if (lstOF[i].isFile()) {
569                 lstFile[count] = lstOF[i].getName();
570                 count++;
571             }
572         }
573
574         return lstFile;
575     }
576
577     protected static void fwrite(
578         String filename, String s,
579         boolean is_append
580     ) throws IOException {
581         Writer output;

```

```
582         output = new BufferedWriter(  
583             new FileWriter(  
584                 filename,  
585                 is_append  
586             )  
587         );  
588         output.append(s);  
589         output.close();  
590     }  
591  
592     protected static boolean isEmpty(ArrayList ds) {  
593         return ds.size() > 0;  
594     }  
595 }  
596
```

8. Class Language

```
1 public class Language {
2     private static String lang_path = null;
3     private static String[] all_language = new String[100];
4     private static String[] all_path_lang = new String[100];
5     private static int count = 0;
6     protected static String current_lang = null;
7     protected static String current_path_lang = null;
8
9     public static void Language() {
10         load_language();
11         set_language();
12     }
13
14     private static void set_language() {
15         int count;
16         int choose;
17         while(true) {
18             count = 1;
19             Lib.clear_console();
20             System.out.println(">> SET LANGUAGE <<");
21             for(String s : Language.all_language) {
22                 if(s == null) {
23                     continue;
24                 }
25                 System.out.println(
26                     String.valueOf(count)
27                     + ". "
28                     + String.valueOf(s)
29                     + " ("
30                     + String.valueOf(
31                         Language.all_path_lang[count-1].replace(
32                             Language.lang_path,
33                             ""
34                         )
35                     )
36                     + ")");
37             };
38             count++;
39         }
40         System.out.println();
41         choose = (int)Lib.int_input(" - Choose", null, false, true, false)-1;
42         try {
43             if(Language.all_language[choose] != null) {
44                 Language.current_lang = Language.all_language[choose];
45                 Language.current_path_lang = Language.all_path_lang[choose];
46                 break;
47             }
48         } catch (Exception e) {
49             continue;
50         }
51     }
52 }
53
54 private static void load_language() {
55     Language.lang_path = Lib.read_config("lang_path");
56     for(String s : Lib.listdir(Language.lang_path)) {
57         if(s.contains(".lang")) {
58             Language.all_language[Language.count] = Lib.read_data(
59                 Language.lang_path + s,
60                 "name",
61                 true,
62                 false
63             )[0];
64             Language.all_path_lang[Language.count] = Language.lang_path + s;
65             Language.count++;
66         }
67     }
68 }
69 }
70 }
```

9. Class main

```
1  import java.io.IOException;
2  import java.io.PrintStream;
3
4  public class Main {
5      public static void main(String args[]) throws IOException{
6          System.setOut(new PrintStream(System.out, true, "UTF8"));
7          Lib.read_config(null);
8          Lib.read_data(null);
9          Lib.load_language();
10         Quanly quanLy = new Quanly();
11         int choose = 0;
12         while(true){
13             Lib.clear_console();
14             Quanly.menu();
15             System.out.print(Lib.getlang("CHOOSE") + ": ");
16             try {
17                 choose = Lib.keyboard.nextInt();
18             } catch (Exception e) {
19                 Lib.keyboard.next();
20                 continue;
21             }
22             Lib.clear_console();
23             switch(choose) {
24                 case 1:
25                     System.out.println(Lib.getlang("PRESS_0_TO_BACK"));
26                     System.out.println("-----");
27                     quanLy.themLaptop((int)Lib.int_input(
28                         Lib.getlang("NUMBER_OF_LAPTOP"),
29                         null,
30                         false,
31                         false,
32                         false)
33                     );
34                     Lib.pause_console();
35                     break;
36                 case 2:
37                     quanLy.suaTTLaptop();
38                     break;
39                 case 3:
40                     quanLy.xemTTLaptop();
41                     Lib.pause_console();
42                     break;
43                 case 4:
44                     quanLy.xoaTTLaptop();
45                     Lib.pause_console();
46                     break;
47                 case 5:
48                     quanLy.xuatTT2File();
49                     Lib.pause_console();
50                     break;
51                 case 6:
52                     quanLy.nhapCONF();
53                     Lib.pause_console();
54                     break;
55                 case 7:
56                     quanLy.xuatCONF();
57                     Lib.pause_console();
58                     break;
59                 case 8:
60                     Lib.clear_console();
61                     if(Lib.yes_or_no(Lib.getlang("EXIT").toLowerCase())) {
62                         System.exit(0);
63                     }
64                 default:
65                     continue;
66             }
67         }
68     }
69 }
70
```

10. File config.properties

```
1 lang_path = ./data/lang/
2 conf_path = ./data/conf/
3 data_path = ./data/data.properties
4
```

11. File data.properties

```
1 # type must not null
2 # type
3 laptop_type = Butterfly, Fly, Bee
4 cpu_type = i3, i5, i7
5 ram_type = DDR3, DDR4
6 screen_type = HD, FullHD
7 harddisk_type = HDD, SSD
8
9 # price (USD)
10 cpu_price = 100, 200, 300
11 ram_price = 100, 200
12 screen_price = 1000, 2000/3000/4000
13 harddisk_price = null, null
14
15 # data screen
16 screen_HD = null
17 screen_FullHD = 1, 2, 3
18
19 # data laptop: cpu, ram, screen, harddisk
20 # write null if not required!
21 Butterfly_data = i7, null, FullHD, SSD
22 Fly_data = i5, null, HD, null
23 Bee_data = i3/i5, null, HD, null
24
25 cpu_company = Intel
26
```


12. File ngôn ngữ - vi.lang

```
1 name = Vietnamese
2
3 ## lang
4 ADD_LAPTOP_INFO = Nhập thông tin Laptop
5 BACK = Quay về trước
6 CAPACITY = Dung lượng
7 CHANGE_LAPTOP_CODE = thay đổi mã Laptop
8 CHOOSE = Lựa chọn
9 CPU = CPU
10 CPU_CODE = Mã CPU
11 CPU_COMPANY = Hãng CPU
12 CURRENT = Hiện tại
13 DATA_ERROR = Dữ liệu lỗi!
14 DEL_ALL_COMPLETED = Đã xóa tất cả!
15 DEL_ALL_LAPTOP = Xóa tất cả Laptop
16 DEL_COMPLETED = Đã xóa!
17 DEL_LAPTOP = Xóa Laptop
18 DEL_LAPTOP_INFO = Xóa thông tin Laptop
19 DEL_WITH_LAPTOP_CODE = Xóa theo mã Laptop
20 DO_YOU_WANT = Bạn có muốn
21 EDIT_LAPTOP_INFO = Sửa thông tin Laptop
22 ENTER_NUM_PLZ = Vui lòng nhập số!
23 ENTER_PARAMETERS_OF_LAPTOP = Nhập thông số laptop thứ
24 EXIT = Thoát
25 EXIT_PROGRAM = Thoát chương trình
26 EXPORT_COMPLETED = Đã xuất file!
27 EXPORT_CONFIG_COMPLETED = Xuất file dữ liệu thành công!
28 EXPORT_LAPTOP_CONFIG = Xuất dữ liệu ra file
29 EXPORT_LAPTOP_INFO = Xuất hóa đơn
30 HARDDISK = Ổ cứng
31 HARDDISK_CAPACITY = Dung lượng ổ cứng
32 HARDDISK_CODE = Mã ổ cứng
33 HARDDISK_TYPE = Loại ổ cứng
34 IMPORT_CONFIG_COMPLETED = Nhập file dữ liệu thành công!
35 IMPORT_LAPTOP_CONFIG = Nhập dữ liệu từ file
36 INPUT = Nhập
37 INPUT_FILE_NAME = Nhập tên file
38 INVALID_CPU = CPU không hợp lệ!
39 INVALID_CPU_CODE = Mã CPU không hợp lệ!
40 INVALID_HARDDISK_TYPE = Kiểu ổ cứng không hợp lệ!
41 INVALID_LAPTOP_CODE = Mã laptop này không tồn tại!
42 INVALID_RAM = RAM không hợp lệ!
43 INVALID_RAM_TYPE = Loại RAM không tồn tại!
44 INVALID_SCREEN_STD = Tiêu chuẩn màn hình không hợp lệ!
45 IS_NOT_VALID = không hợp lệ!
46 LAPTOP_CODE = Mã laptop
47 LAPTOP_CODE_EXIST = Mã laptop này đã tồn tại!
48 LAPTOP_CODE_NOT_EXIST = Mã laptop này không tồn tại!
49 LAPTOP_NAME = Tên laptop
50 LAPTOP_SERIES = Dòng laptop
51 LIST_EMPTY = Danh sách trống!
52 LIST_LAPTOP = Danh sách Laptop hiện có
53 MAIN_PROGRAM = QUẢN LÝ HÓA ĐƠN LAPTOP
54 MANUFACTURER = Hãng sản xuất
55 MONEY = USD
56 NO_INFORMATION = Không có thông tin
57 NUMBER_OF_HARDDISK = Số lượng ổ cứng
58 NUMBER_OF_LAPTOP = Số lượng Laptop
59 NUMBER_OF_RAM = Số lượng RAM
60 NUMBER_ROTATIONS = Số vòng quay
61 NUM_OF_LAPTOP = Laptop thứ
62 PRESS_0_TO_BACK = Nhập 0 để quay về menu
63 PRESS_ENTER_CONTINUE = Nhấn Enter để tiếp tục!
64 PRICE = Đơn giá
65 PROCESS_DATA = Đang nhập dữ liệu
66 RAM = RAM
```

```

64 PRICE = Đơn giá
65 PROCESS_DATA = Đang nhập dữ liệu
66 RAM = RAM
67 RAM_CODE = Mã RAM
68 RAM_TYPE = Loại RAM
69 READ_FAILED = Thất bại khi đọc file
70 READ_SPEED = Tốc độ đọc
71 RESET_INFORMATION_COMPLETED = Thông tin laptop đã được đặt lại
72 SAVE = Lưu
73 SAVE_COMPLETED = Đã lưu!
74 SCREEN = Màn hình
75 SCREEN_CODE = Mã màn hình
76 SCREEN_STD = Tiêu chuẩn va chạm
77 SCREEN_TYPE = Loại màn hình
78 SEL_ITEM_EDIT = Vui lòng chọn mục muốn sửa!
79 TOTAL_PRICE = Tổng giá
80 VIEW_LAPTOP_INFO = Xem thông tin Laptop
81 WARNING = Cảnh báo
82 WARNING001 = Các thông tin (CPU, RAM,...) sẽ bị đặt lại nếu bạn thay đổi!
83 WRITE_SPEED = Tốc độ ghi

```

V. 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

1. Nhập danh sách Laptop

Nhập 0 để quay về menu

Số lượng Laptop: 3

>> Nhập thông số laptop thứ 1:

Dòng laptop (1 - Butterfly, 2 - Fly, 3 - Bee): 1

Mã laptop: laptopbut

Tên laptop: laptop001

>> CPU:

Hãng CPU: Intel

CPU: CPU3 - i7

Đơn giá: 300 USD

>> RAM:

Số lượng RAM: 4

Loại RAM (1 - DDR3, 2 - DDR4): 2

Dung lượng (GB): 8

Mã RAM 1: ram1

Mã RAM 2: ram2

Mã RAM 3: ram3

Mã RAM 4: ram4

Đơn giá: 1600 USD

Tổng giá: 6400 USD

>> Màn hình:

Mã màn hình: scr001

Hãng sản xuất: MSI

Loại màn hình: FullHD

Tiêu chuẩn va chạm (1, 2, 3): 3

Đơn giá: 4000 USD

>> Số lượng ổ cứng: 2

>> Ổ cứng 1:

Loại ổ cứng: SSD

Mã ổ cứng: ssd001

Dung lượng ổ cứng (GB): 256

Đơn giá (USD): 120

Tốc độ đọc (mb/s): 400

Tốc độ ghi (mb/s): 381

>> Ổ cứng 2:

Loại ổ cứng: SSD

Mã ổ cứng: ssd002

Dung lượng ổ cứng (GB): 512

Đơn giá (USD): 240

Tốc độ đọc (mb/s): 401

Tốc độ ghi (mb/s): 381

>> Bạn có muốn lưu? [Y/n]: y

>> Đã lưu!

```
>> Nhập thông số laptop thứ 2:
Dòng laptop (1 - Butterfly, 2 - Fly, 3 - Bee): 2
Mã laptop: laptopfly
Tên laptop: laptop002
>> CPU:
    Hãng CPU: Intel
    CPU: CPU2 - i5
    Đơn giá: 200 USD
>> RAM:
    Số lượng RAM: 2
    Loại RAM (1 - DDR3, 2 - DDR4): 2
    Dung lượng (GB): 8
    Mã RAM 1: ram001
    Mã RAM 2: ram002
    Đơn giá: 1600 USD
    Tổng giá: 3200 USD
>> Màn hình:
    Mã màn hình: scr002
    Hãng sản xuất: Asus
    Loại màn hình: HD
    Đơn giá: 1000 USD
>> Số lượng ổ cứng: 1
>> Ổ cứng 1:
    Loại ổ cứng (HDD, SSD): SSD
    Mã ổ cứng: ssd002
    Dung lượng ổ cứng (GB): 256
    Đơn giá (USD): 125
    Tốc độ đọc (mb/s): 400
    Tốc độ ghi (mb/s): 390
>> Bạn có muốn lưu? [Y/n]: y
>> Đã lưu!
-----
```

```
-----
>> Nhập thông số laptop thứ 3:
Dòng laptop (1 - Butterfly, 2 - Fly, 3 - Bee): 3
Mã laptop: laptopbee
Tên laptop: laptop003
>> CPU:
    Hãng CPU: Intel
    Nhập Mã CPU (CPU1 - i3, CPU2 - i5): CPU1
    Đơn giá: 100 USD
>> RAM:
    Số lượng RAM: 1
    Loại RAM (1 - DDR3, 2 - DDR4): 1
    Dung lượng (GB): 8
    Mã RAM 1: ram01
    Đơn giá: 800 USD
    Tổng giá: 800 USD
>> Màn hình:
    Mã màn hình: scr003
    Hãng sản xuất: Samsung
    Loại màn hình: HD
    Đơn giá: 1000 USD
>> Số lượng ổ cứng: 2
>> Ổ cứng 1:
    Loại ổ cứng (HDD, SSD): SSD
    Mã ổ cứng: ssd01
    Dung lượng ổ cứng (GB): 128
    Đơn giá (USD): 100
    Tốc độ đọc (mb/s): 380
    Tốc độ ghi (mb/s): 350
>> Ổ cứng 2:
    Loại ổ cứng (HDD, SSD): HDD
    Mã ổ cứng: hdd01
    Dung lượng ổ cứng (GB): 512
    Đơn giá (USD): 129
    Số vòng quay (rpm): 7200
>> Bạn có muốn lưu? [Y/n]: y
>> Đã lưu!

>> Nhấn Enter để tiếp tục! |
```

2. Xem thông tin tất cả Laptop

```
-----
>> Laptop thứ 1
- Dòng laptop: Butterfly
- Mã laptop: laptopbut
- Tên laptop: laptop001
- CPU: CPU3 - Intel i7 (300 USD)
- RAM:
  1. ram1 - DDR4 - 8 GB (1600 USD)
  2. ram2 - DDR4 - 8 GB (1600 USD)
  3. ram3 - DDR4 - 8 GB (1600 USD)
  4. ram4 - DDR4 - 8 GB (1600 USD)
- Màn hình: scr001 - MSI - FullHD - 4000 - 3 (4000 USD)
- Ổ cứng:
  1. ssd001 - SSD (256 GB) - [R: 400 MB/s - W: 381 MB/s] (120 USD)
  2. ssd002 - SSD (512 GB) - [R: 401 MB/s - W: 381 MB/s] (240 USD)

>> Tổng giá: 6260 USD
-----
>> Laptop thứ 2
- Dòng laptop: Fly
- Mã laptop: laptopfly
- Tên laptop: laptop002
- CPU: CPU2 - Intel i5 (200 USD)
- RAM:
  1. ram001 - DDR4 - 8 GB (1600 USD)
  2. ram002 - DDR4 - 8 GB (1600 USD)
- Màn hình: scr002 - Asus - HD - 1000 (1000 USD)
- Ổ cứng:
  1. ssd002 - SSD (256 GB) - [R: 400 MB/s - W: 390 MB/s] (125 USD)

>> Tổng giá: 2925 USD
-----
>> Laptop thứ 3
- Dòng laptop: Bee
- Mã laptop: laptopbee
- Tên laptop: laptop003
- CPU: CPU1 - Intel i3 (100 USD)
- RAM:
  1. ram01 - DDR3 - 8 GB (800 USD)
- Màn hình: scr003 - Samsung - HD - 1000 (1000 USD)
- Ổ cứng:
  1. ssd01 - SSD (128 GB) - [R: 380 MB/s - W: 350 MB/s] (100 USD)
  2. hdd01 - HDD (512 GB) - [7200rpm] (129 USD)

>> Tổng giá: 2129 USD
-----

>> Nhấn Enter để tiếp tục! |
```

3. Sửa thông tin Laptop

>> Danh sách Laptop hiện có:

1. laptopbut
2. laptopfly
3. laptopbee

Nhập Mã laptop: **laptopbut**



>> Laptop: laptopbut - laptop001

Vui lòng chọn mục muốn sửa!

1. Dòng laptop (Butterfly)
2. Mã laptop (laptopbut)
3. Tên laptop (laptop001)
4. CPU
5. RAM
6. Màn hình
7. Ổ cứng
0. Quay về trước

Lựa chọn: **1**



!! Cảnh báo: Các thông tin (CPU, RAM, ...) sẽ bị đặt lại nếu bạn thay đổi!

>> Dòng laptop hiện tại: Butterfly

Dòng laptop (1 - Butterfly, 2 - Fly, 3 - Bee): 2

>> Bạn có muốn lưu? [Y/n]: y

!! Cảnh báo: Thông tin laptop đã được đặt lại!

>> Đã lưu!

>> Nhấn Enter để tiếp tục!



>> Laptop: laptopbut - laptop001

Vui lòng chọn mục muốn sửa!

1. Dòng laptop (Fly)
2. Mã laptop (laptopbut)
3. Tên laptop (laptop001)
4. CPU (NO DATA)
5. RAM (NO DATA)
6. Màn hình (NO DATA)
7. Ổ cứng (NO DATA)
0. Quay về trước

Lựa chọn:

4. Xóa thông tin Laptop

- 1. Xóa theo mã Laptop
 - 2. Xóa tất cả Laptop
 - 0. Quay về trước
- Lựa chọn: 1



```
>> Danh sách Laptop hiện có:
  1. laptopbut
  2. laptopfly
  3. laptopbee
Nhập Mã laptop: laptopbut
-----
Tên laptop: laptop001
>> Bạn có muốn Xóa Laptop? [Y/n]: y
```



```
>> Danh sách Laptop hiện có:
  1. laptopfly
  2. laptopbee
```

5. Xuất hóa đơn

>> Đã xuất file! [ds_laptop.txt]

>> Nhấn Enter để tiếp tục!



ds_laptop.txt - Notepad

File Edit Format View Help

```
-----
>> Laptop thứ 1
- Dòng laptop: Fly
- Mã laptop: laptopfly
- Tên laptop: laptop002
- CPU: CPU2 - Intel i5 (200 USD)
- RAM:
  1. ram001 - DDR4 - 8 GB (1600 USD)
  2. ram002 - DDR4 - 8 GB (1600 USD)
- Màn hình: scr002 - Asus - HD - 1000 (1000 USD)
- Ổ cứng:
  1. ssd002 - SSD (256 GB) - [R: 400 MB/s - W: 390 MB/s] (125 USD)

>> Tổng giá: 2925 USD
-----
>> Laptop thứ 2
- Dòng laptop: Bee
- Mã laptop: laptopbee
- Tên laptop: laptop003
- CPU: CPU1 - Intel i3 (100 USD)
- RAM:
  1. ram01 - DDR3 - 8 GB (800 USD)
- Màn hình: scr003 - Samsung - HD - 1000 (1000 USD)
- Ổ cứng:
  1. ssd01 - SSD (128 GB) - [R: 380 MB/s - W: 350 MB/s] (100 USD)
  2. hdd01 - HDD (512 GB) - [7200rpm] (129 USD)

>> Tổng giá: 2129 USD
-----
```

6. Xuất dữ liệu Laptop ra file

Nhập tên file: khanh_test

>> Xuất file dữ liệu thành công! [./data/conf/khanh_test.conf]

>> Nhấn Enter để tiếp tục!



```
CONFIG = Hoa_Don_Laptop
# Config by KhanhNguyen9872

series1 = Fly
code1 = laptopfly
name1 = laptop002
cpu1 = CPU2/Intel/i5/200,
ram1 = ram001/DDR4/8/1600, ram002/DDR4/8/1600,
screen1 = scr002/Asus/HD/0/1000
harddisk1 = ssd002/SSD/256/400/390/0/125,

# Config by KhanhNguyen9872

series2 = Bee
code2 = laptopbee
name2 = laptop003
cpu2 = CPU1/Intel/i3/100,
ram2 = ram01/DDR3/8/800,
screen2 = scr003/Samsung/HD/0/1000
harddisk2 = ssd01/SSD/128/380/350/0/100, hdd01/HDD/512/0/0/7200/129,
```

7. Nhập dữ liệu Laptop ra file

```
>> Các file data tại [./data/conf/]
=> data_04-07-2023
=> test
=> saasd
=> khanh_test
-----
Nhập tên file: khanh_test
>> Đang nhập dữ liệu....
  - Mã laptop: laptopfly
  - Mã laptop: laptopbee

>>Nhập file dữ liệu thành công!

>> Nhấn Enter để tiếp tục!
```

