**ĐÀ NẴNG**

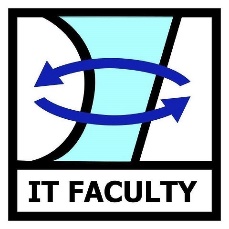
**THÁNG 11 NĂM 2022**

**ĐẠI HỌC ĐÀ NẴNG**

**TRƯỜNG ĐẠI HỌC BÁCH KHOA**

**KHOA CÔNG NGHỆ THÔNG TIN**

**---🙣\*🙡---**

****

**BÁO CÁO ĐỒ ÁN**

**PBL2: DỰ ÁN CƠ SỞ LẬP TRÌNH**

**ỨNG DỤNG HỖ TRỢ**

**QUẢN LÍ QUÁN CÀ PHÊ**

Giáo Viên Hướng Dẫn: **TS. VÕ ĐỨC HOÀNG**

Sinh Viên Thực Hiện:

**LƯU VĂN DUY LỚP: 21T-DT**

**NHÓM: 21.10**

**PHAN THANH TÂM LỚP: 21T-DT**

**NHÓM: 21.10**

**LỜI MỞ ĐẦU**

Trong những năm gần đây, sự phát triển không ngừng của Công nghệ thông tin nói chung và Internet nói riêng đã và đang mang lại những sự thay đổi đáng kể trong cuộc sống con người. Công nghệ tin học được ứng dụng trong nhiều lĩnh vực nghiên cứu khoa học, phát triển kinh tế, quân sự, nghệ thuật và đặc biệt là lĩnh vực dịch vụ. Công nghệ thông tin được ứng dụng rộng rãi để giải quyết các nhu cầu thường ngày của con người: từ các công việc đồng áng,tưới tiêu đến các công việc kĩ thuật máy móc và không thể không nhắc đến sự hỗ trợ của công nghệ thông tin trong việc quản lí kinh doanh.

Với việc quản lý một lượng thông tin lớn về nguyên vật liệu, mặt hàng, nhân viên và công việc bán hang, việc lưu trữ và xử lí thông tin là vô cùng quan trọng, yêu cầu tính logic và độ tin cậy cao. Việc này không chỉ đòi hỏi sự chính xác tuyệt đối về số liệu mà còn yêu cầu về tổ chức cơ sở dữ liệu sao cho hợp lý, tránh các trường hợp giả mạo, gian dối và đem lại trải nghiệm tốt, an toàn cho người sử dụng. Những yêu cầu kể trên đòi hỏi không chỉ dừng lại bằng việc quản lý bằng sổ sách, giấy tờ ghi chép mà cần phải có một hướng chuyên nghiệp, đơn giản, gọn nhẹ hơn.

Chính vì lý do này, nhóm chúng em đã chọn cho mình đề tài: “***Ứng dụng hỗ trợ quản lí quán cà phê*** ” nhằm giải quyết các vấn đề trong công tác quản lí của một quán cà phê và đưa ra các hướng phát triển trong tương lai.

Chúng em xin gửi lời cảm ơn chân thành đến thầy Võ Đức Hoàng, người đã giúp đỡ và định hướng cho chúng em hoàn thành đề tài Đồ án cơ sở này. Trong quá trình hoàn thiện, thầy đã giúp đỡ chúng em rất nhiều, cả về mặt kiến thức lẫn kỹ năng lập trình và đưa ra rất nhiều lời khuyên hữu ích cho chúng em.

Mặc dù đã rất cố gắng hoàn thành đồ án nhưng chắc chắn chúng em sẽ không thể tránh khỏi những thiếu sót. Chúng em mong nhận được sự thông cảm, góp ý và chỉ bảo tận tình của quý thầy cô.

Chúng em xin chân thành cảm ơn.

Đà Nẵng, ngày 15 tháng 11 năm 2021

**Nhóm sinh viên thực hiện đề tài**

**Lưu Văn Duy**

**Phan Thanh Tâm**

**MỤC LỤC**

[.](#_Toc19885203)

[LỜI MỞ ĐẦU 2](#_Toc19885204)

[MỤC LỤC 3](#_Toc19885205)

[DANH MỤC HÌNH VẼ 4](#_Toc19885206)

[1. GIỚI THIỆU ĐỀ TÀI 5](#_Toc19885207)

[1.1. Đặt vấn đề 5](#_Toc19885209)

[1.2. Giới thiệu đề tài 5](#_Toc19885210)

[2. PHÂN TÍCH CHỨC NĂNG HỆ THỐNG](#_Toc19885208) 6

[2.1. Quản lí nhân viên 7](#_Toc19885209)

[2.2. Quản lí phòng ban 8](#_Toc19885210)

[2.3. Quản lí bảng lương 9](#_Toc19885211)

[3. THIẾT KẾ CẤU TRÚC DỮ LIỆU VÀ THUẬT TOÁN 10](#_Toc19885212)

[3.1. Phát biểu bài toán 10](#_Toc19885213)

[3.2. Phân tích và ứng dụng cấu trúc dữ liệu và thuật toán trong hệ thống 10](#_Toc19885214)

a) [Phân tích cấu trúc dữ liệu và thuật toán 10](#_Toc19885214)

b) [Ứng dụng cấu trúc dữ liệu và thuật toán trong hệ thống 13](#_Toc19885214)

[4. PHÂN TÍCH HƯỚNG ĐỐI TƯỢNG VÀ TRIỂN KHAI HỆ THỐNG 19](#_Toc19885216)

[4.1. Cấu trúc hệ thống hướng đối tượng (class, object, relation) 19](#_Toc19885217)

a) [Lớp (class)  19](#_Toc19885214)

b) [Đối tượng(object) 23](#_Toc19885214)

c) [Quan hệ giữa các lớp (relation) 24](#_Toc19885214)

[4.2. Kết quả 25](#_Toc19885218)

[4.2.1. Giao diện chính của chương trình 25](#_Toc19885219)

[4.2.2. Kết quả thực thi của chương trình 25](#_Toc19885220)

a) [Thực hiện chức năng đối với quản lí nhân viên 26](#_Toc19885214)

b) [Thực hiện chức năng đối với quản lí phòng ban 31](#_Toc19885214)

c) [Thực hiện chức năng đối với quản lí bảng lương 33](#_Toc19885214)

[4.2.3. Nhận xét 35](#_Toc19885221)

5. [KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 36](#_Toc19885222)

[5.1. Kết luận 36](#_Toc19885223)

[5.2. Hướng phát triển 36](#_Toc19885224)

[TÀI LIỆU THAM KHẢO 37](#_Toc19885225)

**DANH MỤC HÌNH VẼ**

Hình 2.1. Sơ đồ khối chức năng chương trình..................................................................6  
Hình 2.2. Mô tả chức năng quản lí nhân viên…………………………..……….............7  
Hình 2.3. Mô tả chức năng quản lí phòng ban…….........................................................8  
Hình 2.4. Mô tả chức năng quản lí bảng lương………..................................................9  
Hình 3.1. Biểu diễn danh sách liên kết kép……….…………………….……..............9  
Hình 3.2. Mô tả thuật toán sắp xếp đổi chỗ trực tiếp……........................................10  
Hình 3.3. Mô tả thuật toán tìm kiếm tuần tự……………………….…......................13  
Hình 4.1. Mô tả các lớp dữ liệu (NhanVien.h, PhongBan.h, Luong.h) của chương trình..20  
Hình 4.2. Mô tả các lớp quản lí (QL.h, QLNV.h, QLPB.h, QLL.h) của chương trình….22  
Hình 4.3. Mô tả quan hệ giữa các class………….............................................................24  
Hình 4.4. Menu tổng quát của chương trình……………………………………………25  
Hình 4.5. File dữ liệu phòng ban (PhongBan.txt)...........................................................25  
Hình 4.6. File dữ liệu nhân viên (NhanVien.txt)..........................................................25  
Hình 4.7. File dữ liệu lương (Luong.txt)........................................................................26  
Hình 4.8. Menu nhân viên……………………..............................................................26  
Hình 4.9. Giao diện thực hiện chức năng xem danh sách nhân viên……...................26  
Hình 4.10. Giao diện thực hiện chức năng xem sơ yếu lí lịch nhân viên theo mã nhân viên………………………………………………………………………………….........27  
Hình 4.11. Giao diện thực hiện chức năng cập nhật thông tin nhân viên theo mã nhân viên……………………………………………………………………………………….28  
Hình 4.12. Giao diện thực hiện chức năng thêm nhân viên……………………………29  
Hình 4.13. Kết quả sau khi thực hiện chức năng thêm nhân viên (nhân viên đã được thêm vào danh sách)………...………………………………………………………………....29  
Hình 4.14. Giao diện chức năng xóa nhân viên…………………………………..............30  
Hình 4.15. Giao diện thực hiện chức năng sắp xếp danh sách nhân viên……………….....30  
Hình 4.16. Kết quả thực hiện chức năng xuất file danh sách nhân viên .………………….30  
Hình 4.17. Menu phòng ban……………………………………………………………...31

Hình 4.18. Giao diện thực hiện chức năng xem danh sách phòng ban.................................31  
Hình 4.19. Giao diện thực hiện chức năng xem danh sách nhân viên theo phòng ban …....31  
Hình 4.20. Kết quả khi đồng ý xuất file danh sách nhân viên theo phòng ban ………........32  
Hình 4.21. Giao diện thực hiện chức năng chuyển nhân sự phòng ban................................32  
Hình 4.22. Giao diện thực hiện chức năng cập nhật lịch làm việc phòng ban…..................33  
Hình 4.23. Menu lương…………………………………….............................................33  
Hình 4.24. Giao diện chức năng tìm kiếm xem bảng lương…………………...................33  
Hình 4.25. Giao diện thực hiện chức năng tra cứu lương nhân viên …………..................34  
Hình 4.26. Giao diện chức năng cập nhật bảng chỉ số lương……………………………34  
Hình 4.27. Giao diện thực hiện chức năng xem danh sách tiền lương nhân viên………….35

Hình 4.28. Kết quả khi đồng ý xuất file danh sách tiền lương nhân viên)……………….35

# GIỚI THIỆU ĐỀ TÀI

## 1.1 Đặt vấn đề

# “Nhân lực” là tài sản quan trọng nhất của mỗi doanh nghiệp. Sự thành công của doanh nghiệp phụ thuộc vào tính hiệu quả của việc “quản lý nhân sự” bao gồm cả cách quản lý nhân viên, quản lý tiền lương như thế nào. Điều này đem lại những giá trị góp phần vào thành công của doanh nghiệp. Để đạt được mục tiêu và thực hiện các kế hoạch chiến lược cần phải liên kết chặt chẽ các chính sách nhân sự và các thủ tục với mục tiêu kinh doanh. Trước tiên quản lý nguồn lực vừa là nghệ thuật vừa là khoa học làm cho những mong muốn của doanh nghiệp và mong muốn của nhân viên cùng đạt đến mục tiêu. Nhân viên trông đợi mức lương thỏa đáng, điều kiện làm việc an toàn, sự gắn bó với tổ chức, những nhiệm vụ có tính thách thức, trách nhiệm và quyền hạn. Hiện tại ở một số công ty còn tồn tại cách quản lý thủ công. Cách quản lý này rườm rà và nặng nề. Khi tìm kiếm thông tin một nhân viên trên giấy rất khó khăn và mất thời gian. Đồng thời để thống kê, tổng kết, viết 1 báo cáo cũng rất khó khăn.

## 1.2 Giới thiệu đề tài

Việc tin học hóa quản lý nhân sự mang lại nhiều lợi ích hơn so với quản lý thủ công. Quản lý thông tin về cán bộ, công nhân viên là một bài toán quan trọng và có nhiều ứng dụng trong việc quản lý nguồn nhân lực, chính sách cán bộ… nhằm đưa ra các quyết định trong lĩnh vực xây dựng đội ngũ lao động đủ khả năng và trình độ đáp ứng các nhu cầu trong giai đoạn mới. Nhằm giảm sự phức tạp trong việc quản lí nguồn nhân lực ở doanh nghiệp cũng như của các công ty, chúng em xây dựng, thiết kế một hệ thống quản lý nhân sự đơn giản bằng ngôn ngữ C++ với các tác vụ thêm, sửa, xóa, tìm kiếm và tạo thông tin theo ý của người dùng.

# 2. PHÂN TÍCH CHỨC NĂNG HỆ THỐNG

* Hệ thống được viết thành một chương trình đơn giản, người sử dụng có thể tương tác với nhiều đối tượng, mỗi đối tượng được sử dụng với những chức năng khác nhau.
* Hệ thống sẽ có 3 chức năng tổng quát:

🡪 *Chức năng 1*: Quản lí nhân viên.

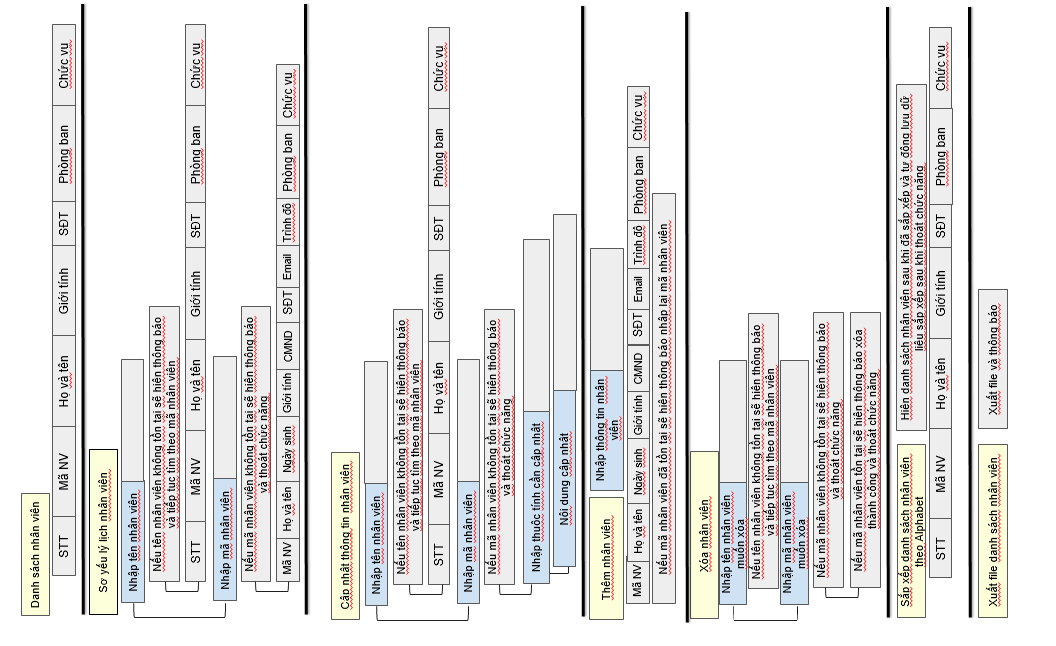
🡪 *Chức năng 2*: Quản lí phòng ban.

🡪 *Chức năng 3*: Quản lí bảng lương theo chức vụ.

****

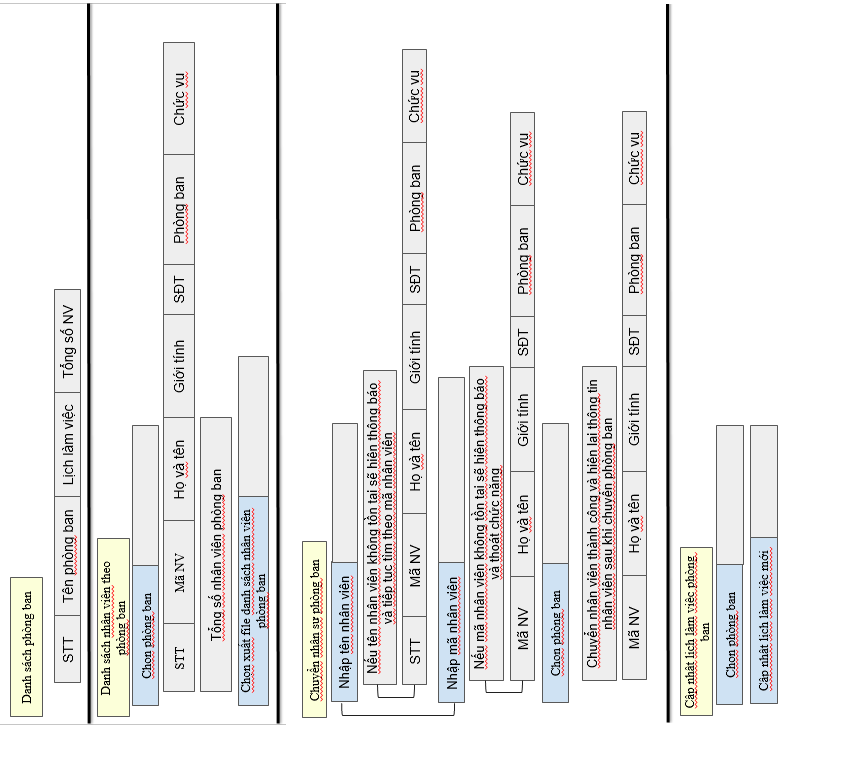
***Hình 2.1:*** *Sơ đồ khối chức năng chương trình*

**2.1. Quản lí nhân viên**



# *Hình 2.2: Mô tả chức năng quản lí nhân viên*

**2.2 Quản lí phòng ban**



***Hình 2.3:*** *Mô tả chức năng quản lí phòng ban*

# 2.3 Quản lí bảng lương

# 

***Hình 2.4:*** *Mô tả chức năng quản lí bảng lương*

# 3. THIẾT KẾ CẤU TRÚC DỮ LIỆU VÀ THUẬT TOÁN

# 3.1. Phát biểu bài toán

Một doanh nghiệp (công ty) có sẵn thông tin, bao gồm sơ yếu lí lịch về nhân viên, sự phân bố nhân viên theo phòng ban, thông tin về phòng ban, số liệu về bảng lương được tính theo chức vụ. Xây dựng chương trình quản lí ba thành phần (nhân viên, phòng ban, bảng lương nhân viên tính theo chức vụ).

Cấu trúc nhân viên gồm: Mã nhân viên, họ tên, ngày sinh, giới tính, CMND, số điện thoại, email, địa chỉ, trình độ, phòng ban, chức vụ, tiền lương.

Cấu trúc phòng ban gồm: Mã phòng ban, tên phòng ban, lịch làm việc, tổng số nhân viên. Với 3 phòng ban cố định: Kĩ thuật, kinh doanh, tài chính.

Cấu trúc bảng lương tính theo chức vụ: Chức vụ, hệ số lương, lương cơ bản(triệu VNĐ). Với 3 chức vụ cố định: Trưởng phòng, phó phòng, nhân viên.

**a) Input:** Nhập thủ công hoặc đọc file chứa thông tin của từng nhân viên, nhập file thông tin về phòng ban, file số liệu về bảng lương được tính theo chức vụ.

**b) Output:** Các thao tác chức năng quản lí thông tin đối với nhân viên, phòng ban, bảng lương nhân viên tính theo chức vụ.

**3.2. Phân tích và ứng dụng cấu trúc dữ liệu và thuật toán trong hệ thống**

**a) Phân tích cấu trúc dữ liệu và thuật toán:**

**Danh sách liên kết kép**

Danh sách liên kết kép là danh sách mà mỗi phần tử trong danh sách có kết nối với phần từ đứng ngay trước và phần tử đứng ngay sau nó. Dưới đây là một số khái niệm quan trọng cần ghi nhớ về **Danh sách liên kết kép**.

* **Link:** mỗi link của *“Danh sách liên kết kép”* có thể lưu giữ một dữ liệu và được gọi là một phần tử.
* **Next:** mỗi link của một Danh sách liên kết có thể chứa một link tới next link và được gọi là Next.
* **Prev:** mỗi link của một Danh sách liên kết có thể chứa một link tới previous link và được gọi là Prev.
* **First và Last:** một Danh sách liên kết chứa link kết nối tới first link được gọi là First và tới last link được gọi là Last.



***Hình 3.1:*** *Biểu diễn danh sách liên kết kép*

Như hình minh họa trên (hình 5):

* Danh sách liên kết đôi chứa một phần tử link và được gọi là First và Last.
* Mỗi link mang một trường dữ liệu và một trường link được gọi là Next.
* Mỗi link được liên kết với phần tử kế tiếp bởi sử dụng Next Link.
* Mỗi link được liên kết với phần tử phía trước bởi sử dụng Prev Link.
* Last Link mang một link trỏ tới NULL để đánh dầu phần cuối của Danh sách liên kết.

**Thuật toán sắp xếp danh sách bằng phương pháp “Đổi chỗ trực tiếp” (Interchange sort)**

* Khái niệm nghịch thế:
* Xét một mảng các đối tượng a[0], a[1], … a[n-1]
* Nếu có i<j và a[i] > a[j] hay một sự khác biệt, thì ta gọi đó là một nghịch thế.
* Mảng chưa sắp xếp sẽ có nghịch thế, mảng đã có thứ tự sẽ không chứa nghịch thế.
* Nhận xét: Để sắp xếp một dãy số hay một mảng các đối tượng, ta có thể xét các nghịch thế có trong dãy và làm triệt tiêu dần chúng đi.
* Ý tưởng:
* Xuất phát từ đầu dãy, tìm tất cả nghịch thế chứa phần tử này, triệt tiêu chúng bằng cách đổi chỗ phần tử này với phần tử tương ứng trong cặp nghịch thế.
* Lặp lại xử lý trên với các phần tử tiếp theo trong dãy.

#### Các bước tiến hành thuật toán khi sắp xếp tăng dần:

#### Bước 1: i = 0;//Bắt đầu từ đầu dãy

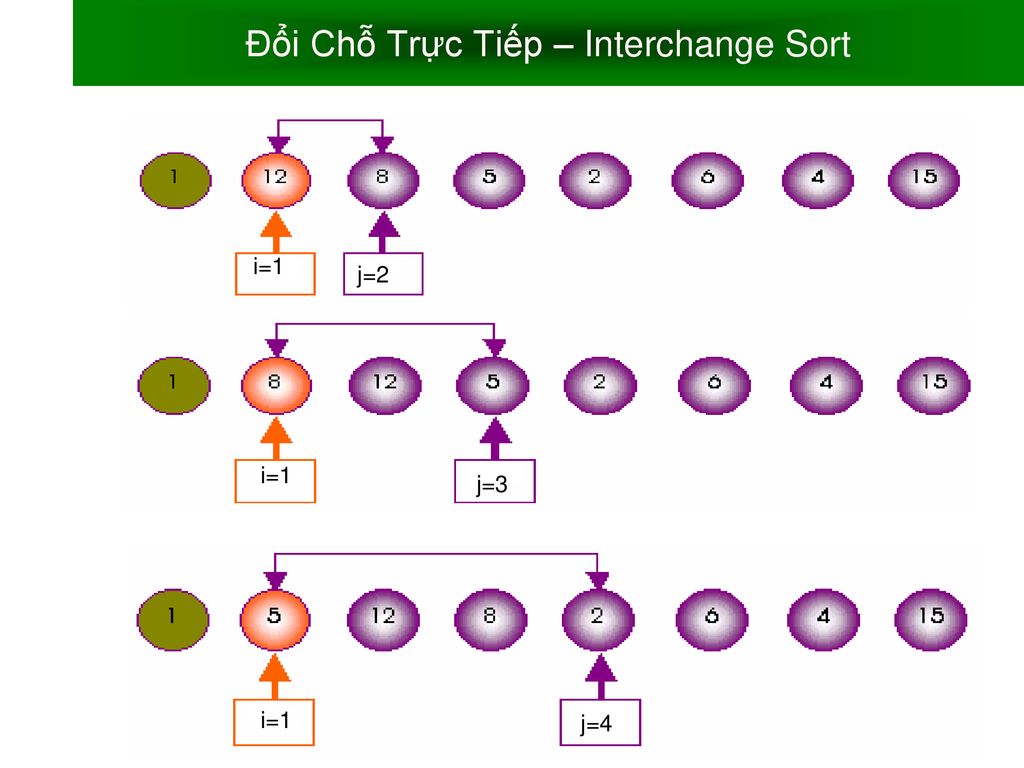
Bước 2: j=i+1;//Thực hiện phép so sánh giữa a[i] và a[j] với j > i.

Bước 3:

Khi j < n thì kiểm tra: nếu phép so sánh đúng thì hoán vị a[j] và a[i].

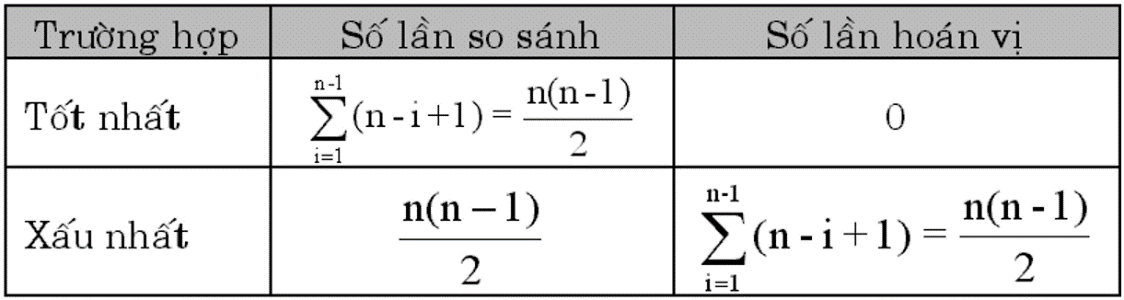
Gán j=j+1; rồi thực hiện lại bước 3.

Bước 4: i=i+1; nếu i < n-1 thì lặp lại bước 2, ngược lại -> Dừng.



***Hình 3.2:*** *Mô tả thuật toán sắp xếp đổi chỗ trực tiếp*

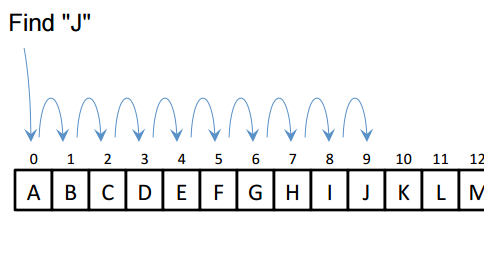
* Đánh giá:
* Số lượng các phép so sánh xảy ra không phụ thuộc vào tình trạng của dãy số ban đầu.
* Số lượng phép hoán vị thực hiện tùy thuộc vào kết quả so sánh.



*Bảng**đánh giá độ phức tạp thuật toán sắp xếp đổi chỗ trực tiếp*

**Thuật toán tìm kiếm tuần tự (Linear search)**

* Là phương pháp tìm kiếm một phần tử cho trước trong một danh sách bằng cách duyệt lần lượt từng phần từ của danh sách đó đến khi nào tìm được giá trị mong muốn hay đã duyệt hết qua hết danh sách.
* Ý tưởng:
  + Bắt đầu từ bản ghi đầu tiên của mảng, duyệt từ đầu mảng đến cuối mảng với x.
  + Nếu phần tử đang duyệt bằng x thì trả về vị trí.
  + Nếu không tìm thấy bất cứ phần từ nào khi đã duyệt hết thì trả về -1.



***Hình 3.3:*** *Mô tả thuật toán tìm kiếm tuần tự*

* Độ phức tạp thời gian: O(n) khi phần tử tìm kiếm nằm cuối danh sách hoặc không có trong danh sách
* Thời gian chạy tốt nhất: O(1) khi phần tử cần tìm nằm ngay đầu danh sách
* Độ phức tạp không gian: O(n)

**b) Ứng dụng cấu trúc dữ liệu và thuât toán trong hệ thống**

**Danh sách liên kết kép**

* Tạo một khuôn mẫu lớp **List** để xây dựng cấu trúc dữ liệu cho chương trình, trong đó dùng kiểu dữ liệu Struct để định nghĩa một lớp đối tượng, và dùng cấu trúc dữ liệu danh sách liên kết kép để tạo và lưu trữ một danh sách lớp đối tượng theo một khuông mẫu lớp (Class Template).

**\* Khai báo khuôn mẫu lớp**

template <class T>

class List

**\* Khai báo thuộc tính private**

private:

int \_size;

**\* Định nghĩa cấu trúc một Node**

**//** Các node: Mỗi node bao gồm phần dữ liệu, một con trỏ prev để trỏ đến node ngay trước nó và một con trỏ next để trỏ tới node ngay tiếp sau nó.

    struct NODE

    {

        T data; /\* Phần dữ liệu của một node \*/

        NODE \*pNext; /\* Con trỏ tới node kế tiếp \*/

        NODE \*pPrev; /\* Con trỏ tới node trước nó \*/

        NODE(T data)

        {

            this->data = data;

            this->pNext = this->pPrev = NULL;

        }

    };

    NODE \*pHead; /\* Con trỏ tới node đầu tiên\*/

    NODE \*pTail; /\* Con trỏ tới node cuối cùng\*/

**\* Hàm dựng**

    List()

    {

        this->pHead = this->pTail = NULL;

        this->\_size = 0;

    }

**\* Hàm hủy**

    ~List()

    {

        for (int i = 0; i < this->\_size - 2; i++)

        {

            pHead = pHead->pNext;

            delete pHead->pPrev;

        }

        delete pHead;

    }

**\* Cấp phát bộ nhớ cho node**

    NODE \*contructorNode(T data)

    {

        NODE \*node = new NODE(data);

        return node;

    }

**\* Thêm 1 node vào cuối danh sách**

void push\_back(T data)

    {

        NODE \*node = contructorNode(data);

        if (pHead == NULL)

        {

            this->pHead = this->pTail = node;

            this->\_size++;

            return;

        }

        this->pTail->pNext = node;

        node->pPrev = this->pTail;

        this->pTail = node;

        this->\_size++;

    }

**\* Trả về kích thước của danh sách các node**

    const int getSize()

    {

        return this->\_size;

    }

**\* Đa năng hóa toán tử [] tạo thành một mảng các node**

    T &operator[](int index)

    {

        NODE \*node = this->pHead;

        for (int i = 0; i < index; i++)

        {

            node = node->pNext;

        }

        return node->data;

    }

**\* Xoá 1 node bất kì**

**/\*** Dùng để xóa một node ra khỏi danh sách với điều kiện là ta đã biết địa chỉ của node cần xóa.\*/

void erase(int index)

{

// Trường hợp vị trí node không tồn tại trong danh sách

    if (index < 0 || index >= this->\_size)

    return;

// Trường hợp kích thước danh sách bằng 0

else if (this->\_size == 0)

{

     return;

}

// Trường hợp kích thước danh sách bằng 1

     else if (this->\_size == 1)

     {

      delete pHead;

          pHead = pTail = NULL;

      }

/\* Trường hợp node ở vị trí đầu, ta chỉ việc chuyển con trỏ head để trỏ vào vị trí kế tiếp của node đầu tiên và giải phóng bộ nhớ đã cấp cho node đầu tiên đó.\*/

else if (index == 0)

{

NODE \*node = pHead->pNext;

pHead->pNext->pPrev = NULL;

delete pHead;

pHead = node;

}

/\* Trường hợp node ở vị trí cuối cùng của danh sách, ta duyệt đến vị trí của node đứng ngay trước node cuối cùng, sau đó gán cho con trỏ next của node đó trỏ tới NULL và giải phóng bộ nhớ đã cấp cho node cuối cùng.\*/

else if (index == this->\_size - 1)

{

     NODE \*node = pTail->pPrev;

     pTail->pPrev->pNext = NULL;

     delete pTail;

            pTail = node;

        }

/\* Trường hợp node ở giữa danh sách phức tạp hơn một chút bao gồm các bước:

Bước 1: Gán con trỏ cho node đầu tiên của mảng và thực hiện vòng lặp lần lượt gán con trỏ kế tiếp cho đến vị trí node cần xóa.

Bước 2: Chuyển con trỏ next của node đứng ngay trước node cần xóa tới vị trí của node ngay sau node cần xóa đó.

Bước 3: Chuyển con trỏ prev của node đứng ngay sau node cần xóa về vị trí của node ngay trước node cần xóa đó.

Bước 4: Giải phóng bộ nhớ đã cấp cho node vừa xóa.\*/

else

       {

NODE \*node = this->pHead;

for (int i = 0; i < index; i++)

{

  node = node->pNext;

}

     node->pPrev->pNext = node->pNext;

     node->pNext->pPrev = node->pPrev;

     delete node;

}

  this->\_size--;

}

**\* Clear dữ liệu trong list và xóa vùng nhớ**

void clear()

{

// Trường hợp kích thước danh sách bằng 0

if (\_size == 0)

return;

// Trường hợp kích thước danh sách bằng 1

if (\_size == 1)

{

delete pHead;

    pHead = pTail = NULL;

    \_size = 0;

    return;

}

/\* Tạo hai con trỏ phụ trỏ tới node đầu tiên và node tiếp theo của node đầu tiên.\*/

NODE \*tempPrev = pHead;

NODE \*tempNext = pHead->pNext;

/\* Thực hiện vòng lặp for cho chạy từ đầu đến cuối mảng lần lượt thực hiện các thao tác ở mỗi vòng lặp: xóa vùng nhớ node đầu tiên, node tiếp theo trở thành node đầu tiên của danh sách, và tiếp tục thực hiện cho con trỏ phụ tempPrev trỏ đến node đầu tiên, và con trỏ phụ tempNext trỏ đến node tiếp theo của node đầu tiên. Thực hiện cho đến khi kích thước danh sách bằng 1, rồi xóa vùng nhớ của node còn lại, kích thước danh sách trở về 0.\*/

        for (int i = 0; i < \_size - 1; i++)

        {

            delete tempPrev;

            tempPrev = tempNext;

            tempNext = tempNext->pNext;

        }

        delete tempPrev;

        \_size = 0;

}

**\* Clear dữ liệu trong list nhưng không xóa vùng nhớ**

void clearData()

{

    pHead = pTail = NULL;

    \_size = 0;

}

**Sắp xếp danh sách nhân viên theo bảng Alphabet bằng thuật toán sắp xếp “Đổi chỗ trực tiếp” (Interchange sort)**

void QLNV::Sapxepdanhsach()

{

// Thuật toán sắp xếp họ tên đệm trước rồi đến sắp xếp tên.

/\* Thực hiện lồng vòng lặp, so sánh hai chuỗi mang thuộc tính họ tên đệm của hai nhân viên thứ i và j, nếu kết quả nhỏ hơn 0 thì hoán đổi 2 vị trí cho nhau.\*/

for (int i=0; i< DatabaseNV.getSize()-1; i++)

{

for (int j=DatabaseNV.getSize()-1; j>i; j--)

    {

if (strcmp(this->DatabaseNV[j]->GetHotendem().c\_str(),

this->DatabaseNV[j-1]->GetHotendem().c\_str())<0)

{

NhanVien \*k=DatabaseNV[j];

    DatabaseNV[j]=DatabaseNV[j-1];

    DatabaseNV[j-1]=k;

}

    }

}

/\* Thực hiện lồng vòng lặp, so sánh hai chuỗi mang thuộc tính tên của hai nhân viên thứ i và j, nếu kết quả nhỏ hơn 0 thì hoán đổi 2 vị trí cho nhau.\*/

for (int i=0; i< DatabaseNV.getSize()-1; i++)

{

for (int j=DatabaseNV.getSize()-1; j>i; j--)

    {

if (strcmp(this->DatabaseNV[j]->Getten().c\_str(),

this->DatabaseNV[j-1]->Getten().c\_str())<0)

{

NhanVien \*k=DatabaseNV[j];

   DatabaseNV[j]=DatabaseNV[j-1];

    DatabaseNV[j-1]=k;

}

}

}

}

**Tìm kiếm tuần tự đối với danh sách nhân viên**

// Tìm kiếm nhân viên theo mã nhân viên

int QLNV::FindIndex(const string &index)

{

bool check = false;

for (int i = 0; i < this->DatabaseNV.getSize(); i++)

{

if (DatabaseNV[i]->GetMaNV() == index)

return i;

}

return -1;

}

// Tìm kiếm nhân viên theo tên nhân viên

int QLNV::FindIndexTen(const string &index)

{

    bool check = false;

    for (int i = 0; i < this->DatabaseNV.getSize(); i++)

{

    if (DatabaseNV[i]->Getten() == index)

        return i;

return -1

}

}

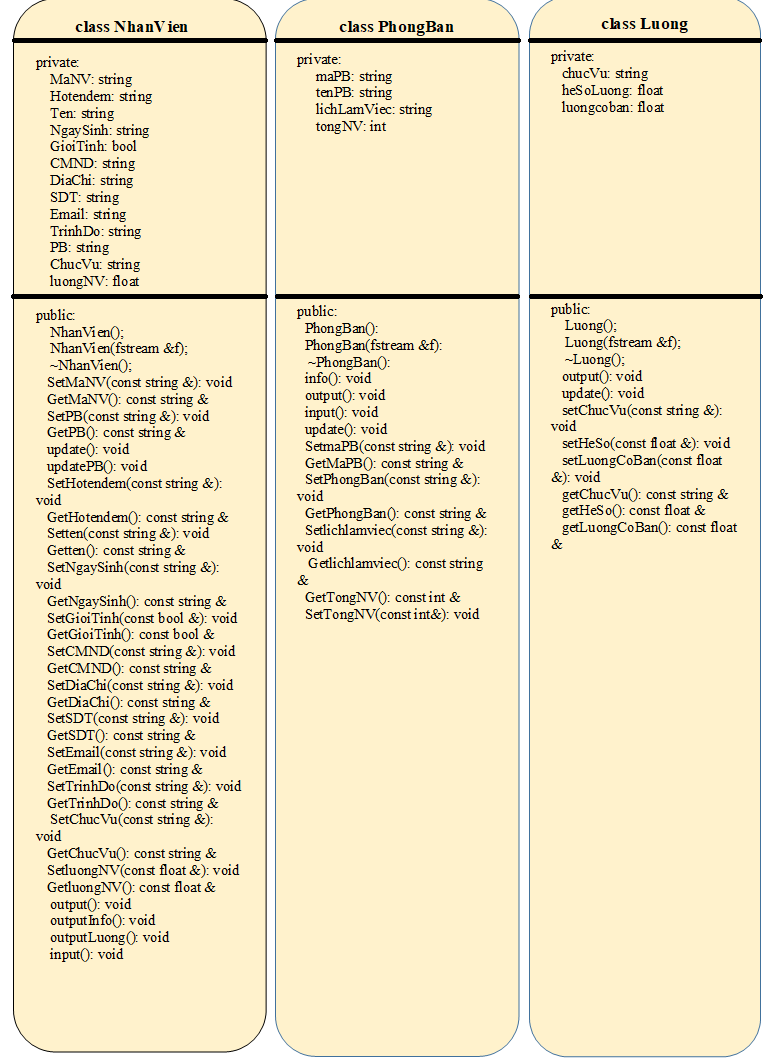
**4. PHÂN TÍCH HƯỚNG ĐỐI TƯỢNG VÀ TRIỂN KHAI HỆ THỐNG**

**4.1. Cấu trúc hệ thống hướng đối tượng (class, object, relation)**

Xây dựng mã nguồn chương trình theo các lớp và đối tượng đáp ứng bốn tính chất (đóng gói, kế thừa, đa hình, trừu tượng) của ***“Lập trình hướng đối tượng”*** (tiếng Anh: Object-oriented programming, viết tắt: OOP).

**a) Lớp (class)**

* + Lớp dữ liệu: Lớp nhân viên (NhanVien.h, NhanVien.cpp), lớp phòng ban (PhongBan.h, PhongBan.cpp), lớp lương (Luong.h, Luong.cpp).
* Các lớp dữ liệu chứa các thuộc tính private đặc trưng cho từng đối tượng đảm bảo tính chất đóng gói của ***“Lập trình hướng đối tượng”***.
* Các đối tượng được khởi tạo mặc định, bên cạnh đó còn được khởi tạo với dữ liệu nhập từ file,không những thế còn được tạo và truy cập thông qua các hàm Get – Set. Đáp ứng tính đa hình của ***“Lập trình hướng đối tượng”***, đa hình với nạp chồng phương thức.
* Từng lớp chứa các phương thức riêng phù hợp cho việc truy xuất dữ liệu của từng đối tượng.



***Hình 4.1:*** *Mô tả các lớp dữ liệu (NhanVien.h, PhongBan.h, Luong.h) của chương trình*

* Lớp quản lí: Lớp quản lí, lớp quản lí nhân viên, lớp quản lí phòng ban, lớp quản lí lương.

\* Tạo một lớp cơ sở Quản lí là lớp trừu tượng (QL.h, QL.cpp), trong đó:

* Tạo 3 thuộc tính với quyền truy cập protected với kiểu dữ liệu List:

protected:

        List<NhanVien\*> DatabaseNV; //Danh sách nhân viên

        List<PhongBan\*> DatabasePB; //Danh sách phòng ban

        List<Luong\*> DatabaseL; //Danh sách bảng lương

* Tao các phương thức với quyền truy cập public:

public:

QL();

  virtual ~QL(){}; //Hàm ảo

    virtual void InDanhSach() = 0; // Hàm thuần ảo

    void UpdateData();

\* Tạo 3 lớp dẫn xuất (lớp quản lí nhân viên, lớp quản lí phòng ban, lớp quản lí lương) kế thừa lớp trừu tượng Quản lí.

* Lớp Quản lí nhân viên (QLNV.h, QLNV.cpp):

class QLPB: public QL{

public:

QLNV();

~QLNV() override;// Đa hình lại hàm hủy từ lớp Quản lí

void InDanhSach() override;

/\*Đa hình lại hàm thuần ảo InDanhSach() từ lớp Quản lí, đa hình với ghi đè phương thức\*/

void InDanhSachTen(const string &); //In danh sách tất cả NV tim theo ten

int FindIndex(const string&);

int FindIndexTen(const string&);

void UpdateInfo(const string); //Cập nhật thông tin NV được chọn

void Them(); //Thêm NV

void ShowInfo(const string&); //In sơ yếu lý lịch NV

void Delete(const string); //Xoá NV

void Luufile();

void Sapxepdanhsach();};

* Lớp Quản lí phòng ban (QLPB.h, QLPB.cpp):

class QLPB: public QL{

private:

List<NhanVien\*> DatabaseNVPB; //Thuộc tính riêng, danh sách nhân viên theo phòng ban.

public:

QLPB();

~QLPB();

void InDanhSach() override; //In danh sách phòng ban,

void showNV(); //In danh sách nhân viên theo phòng ban

/\*Đa hình lại hàm thuần ảo InDanhSach() từ lớp Quản lí, đa hình với ghi đè phương thức\*/

void UpdateInfo(); //Cap nhat lich lam viec phong ban

void Luufile(); // Xuất file danh sách nhân viên theo phòng ban

void ChuyenNV(const string &); //Chuyển nhân sự phòng ban};

+ Lớp Quản lí bảng lương (QLL.h, QLL.cpp):

class QLL: public QL {

public:

QLL();

~QLL();

void InDanhSach() override; //In danh sách hệ số lương

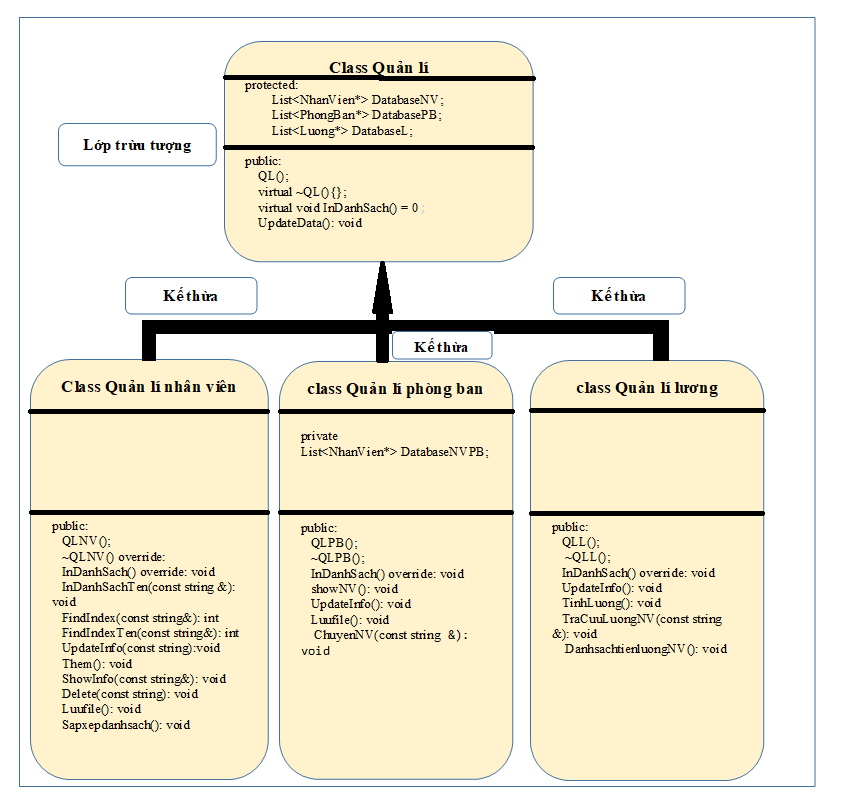
/\*Đa hình lại hàm thuần ảo InDanhSach() từ lớp Quản lí, đa hình với ghi đè phương thức\*/

void UpdateInfo(); //Cập nhật hệ số lương

void TinhLuong();

void TraCuuLuongNV(const string &);

void DanhsachtienluongNV();};



***Hình 4.2:*** *Mô tả các lớp quản lí (QL.h, QLNV.h,QLPB.h, QLL.h) của chương trình*

* Lớp Menu (Menu.h, Menu.cpp) chứa các menu thao tác tương ứng với từng đối tượng quản lí

class Menu

{

public:

//In thông tin PBL

void info();

//Menu đối tượng thao tác chính

void mainMenu();

//Menu các thao tác với nhân viên

void menuNV();

//Menu các thao tác với phòng ban

void menuPB();

//Menu name PB

void namePB();

//Menu các thao tác với lương

void menuLuong();

void TiepTuc();

void Tieude();

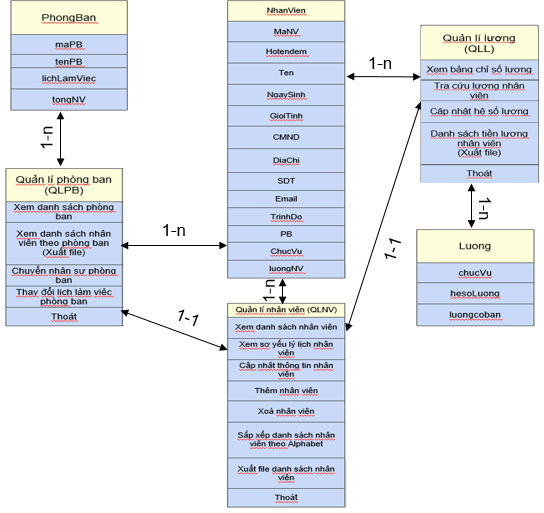
};

* Hàm main(); //Thực thi chương trình

**b) Đối tượng (object)**

* Các lớp dữ liệu tập hợp ba đối tượng: nhân viên, phòng ban, lương
* Đối tượng nhân viên chứa các thuộc tính bao gồm mã nhân viên, họ tên, ngày sinh, giới tính, CMND, số điện thoại, email, địa chỉ, trình độ, phòng ban, chức vụ, tiền lương; và được truy cập bởi các phương thức trong lớp nhân viên.
* Đối tượng phòng ban chứa các thuộc tính bao gồm phòng ban, tên phòng ban, lịch làm việc, tổng số nhân viên; và được truy cập bởi các phương thức trong lớp phòng ban.
* Đối tượng lương chứa các thuộc tính bao gồm chức vụ, hệ số lương, lương cơ bản (triệu VNĐ); và được truy cập bởi các phương thức trong lớp lương.
* Các lớp quản lí tập hợp ba đối tượng: danh sách nhân viên, danh sách phòng ban, danh sách bảng lương. Tùy thuộc quan hệ giữa các lớp mà từng danh sách đối tượng được truy cập bởi các phương thức phù hợp.

**c) Quan hệ giữa các lớp (relation)**

****

***Hình 4.3:*** *Quan hệ giữa các class*

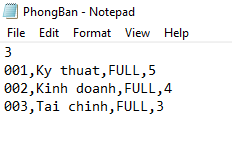
**4.2. Tổ chức chương trình**

**4.2.1. Giao diện chính của chương trình**

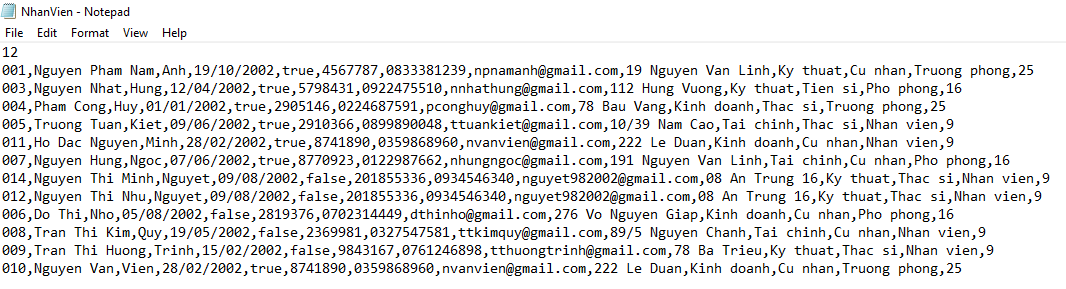
****

***Hình 4.4:*** *Menu tổng quát của chương trình*

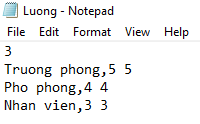
**4.2.2. Kết quả thực thi của chương trình**

****

***Hình 4.5:*** *File dữ liệu phòng ban (PhongBan.txt)*

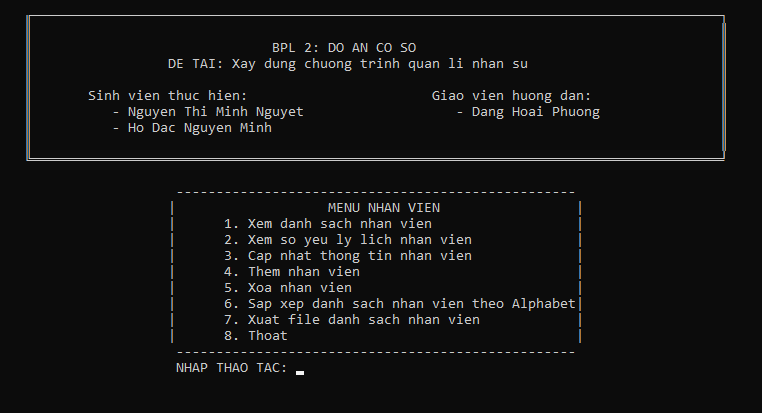
****

***Hình 4.6:*** *File dữ liệu nhân viên (NhanVien.txt)*

****

***Hình 4.7:*** *File dữ liệu lương (Luong.txt)*

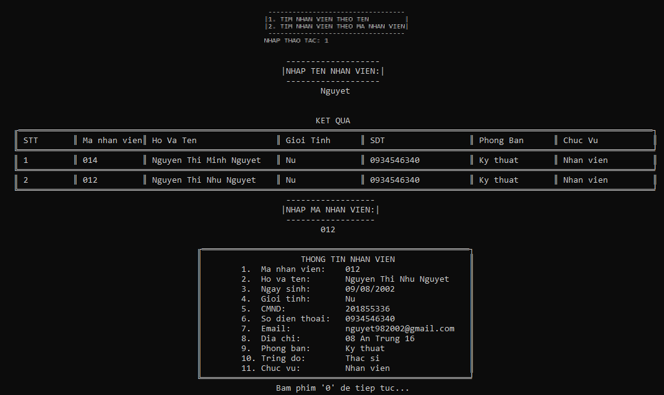
1. **Thực hiện chức năng đối với quản lí nhân viên**

****

***Hình 4.8:*** *Menu nhân viên*

****

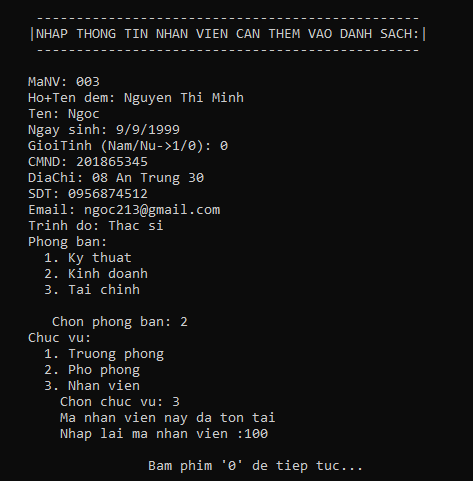
***Hình 4.9:*** *Giao diện thực hiện chức năng xem danh sách nhân viên*

****

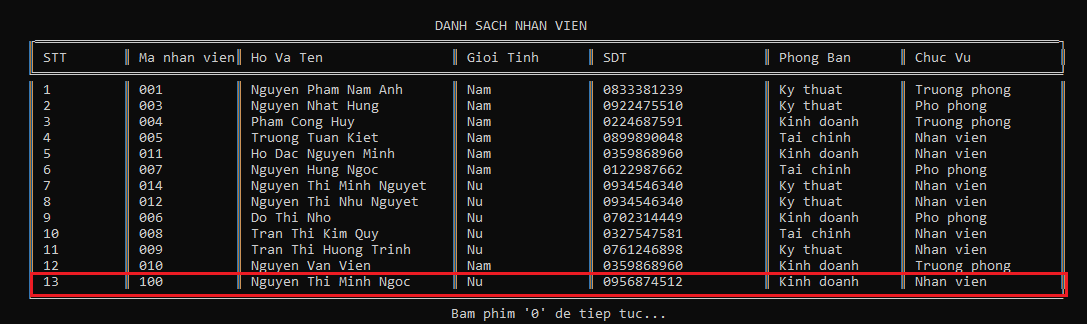
***Hình 4.10:*** *Giao diện thực hiện chức năng xem sơ yếu lí lịch nhân viên theo mã nhân viên*

****

***Hình 4.11:*** *Giao diện thực hiện chức năng cập nhật thông tin nhân viên theo mã nhân viên*

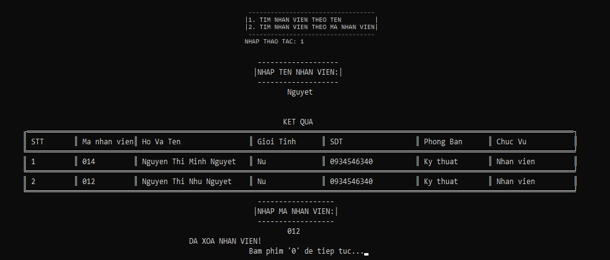
****

***Hình 4.12:*** *Giao diện thực hiện chức năng thêm nhân viên*

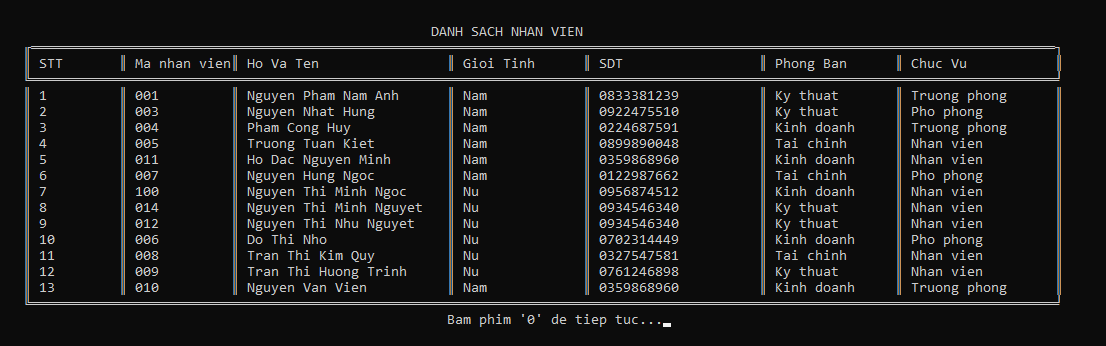
****

***Hình 4.13:*** *Kết quả sau khi thực hiện chức năng thêm nhân viên*

*(nhân viên đã được thêm vào danh sách)*

****

***Hình 4.14:*** *Giao diện thực hiện chức năng xóa nhân viên*

****

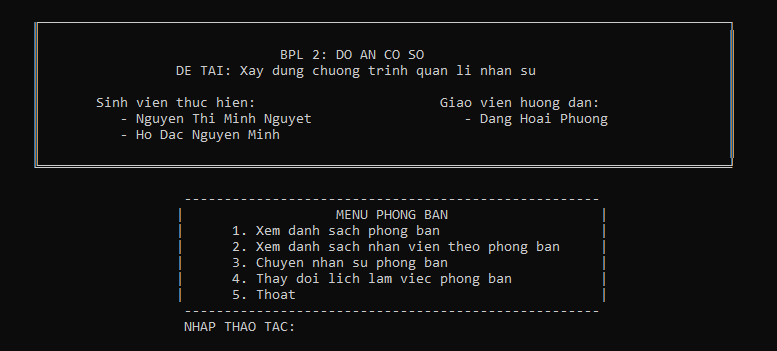
***Hình 4.15:*** *Giao diện thực hiện chức năng sắp xếp danh sách nhân viên*

****

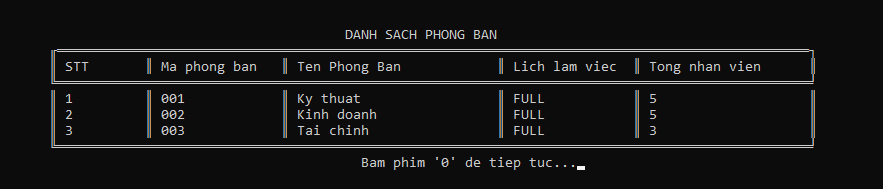
***Hình 4.16:*** *Kết quả thực hiện chức năng xuất file danh sách nhân viên*

*(File Danh\_sach\_nhan\_vien.txt)*

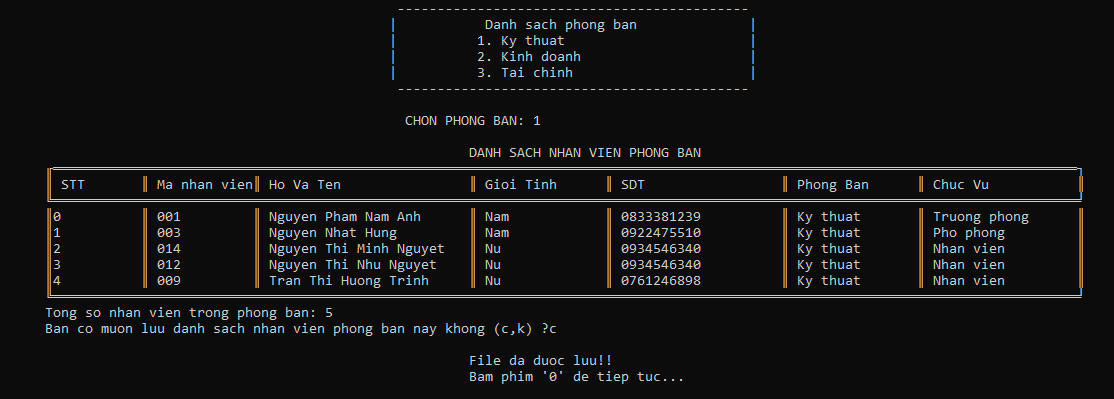
**b) Thực hiện chức năng đối với quản lí phòng ban**

****

***Hình 4.17:*** *Menu phòng ban*

****

***Hình 4.18:*** *Giao diện thực hiện chức năng xem danh sách phòng ban*

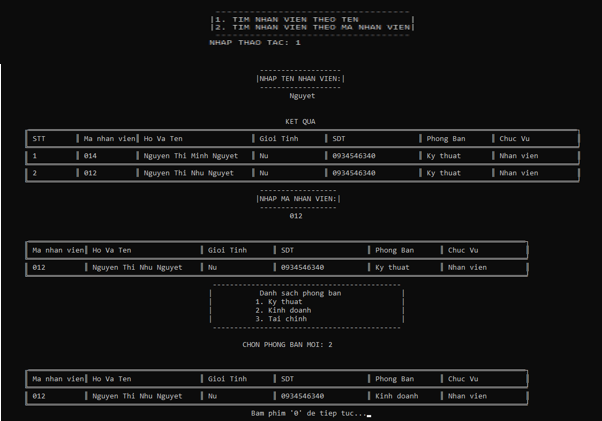
****

***Hình 4.19:*** *Giao diện thực hiện chức năng xem danh sách nhân viên theo phòng ban*

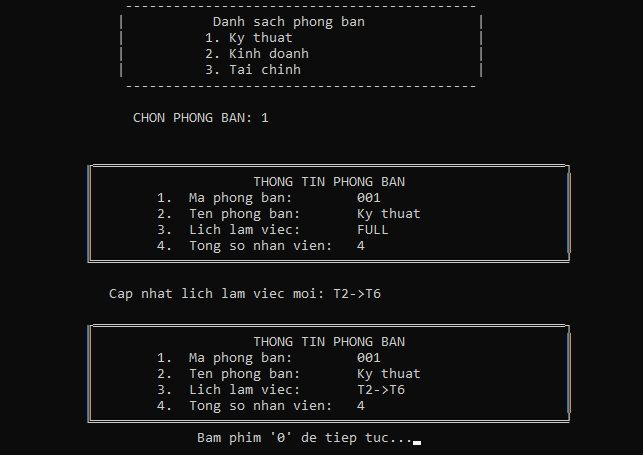
****

***Hình 4.20:*** *Kết quả khi đồng ý xuất file danh sách nhân viên theo phòng ban*

*(File Danh\_sach\_nhan\_vien\_theo\_phong\_ban.txt)*

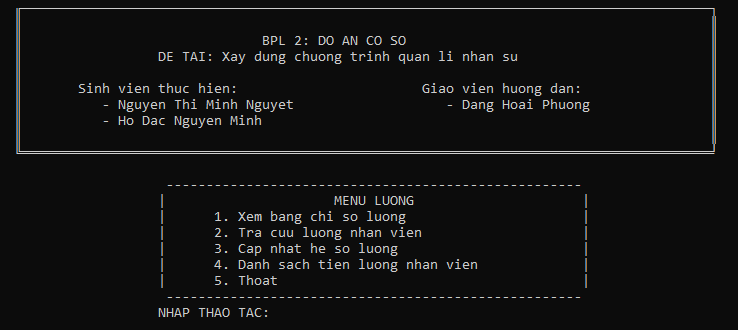
****

***Hình 4.21:*** *Giao diện thực hiện chức năng chuyển nhân sự phòng ban*

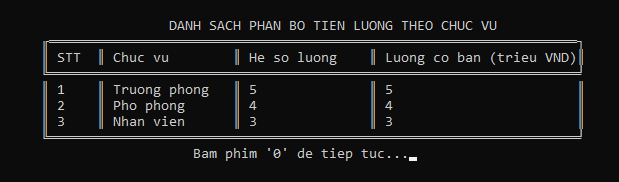
****

***Hình 4.22:*** *Giao diện thực hiện chức năng cập nhật lịch làm việc phòng ban*

**c) Thực hiện chức năng đối với quản lí bảng lương**

****

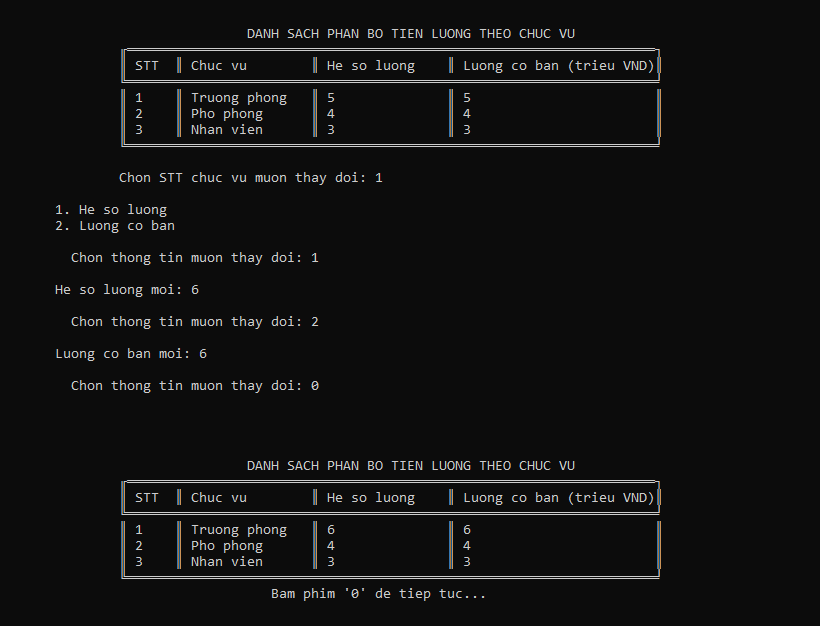
***Hình 4.23:*** *Menu Lương*

****

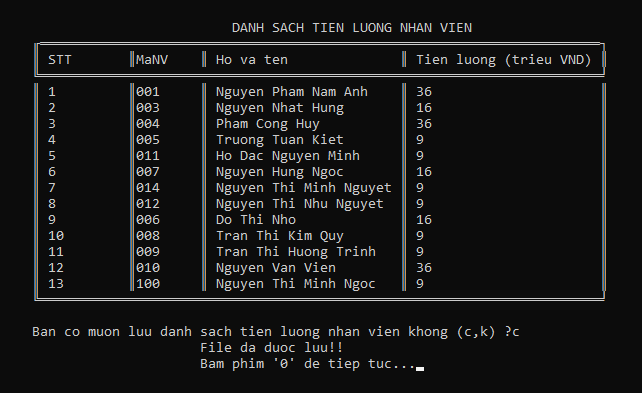
***Hình 4.24:*** *Giao diện thực hiện chức năng xem bảng lương*

****

***Hình 4.25:*** *Giao diện thực hiện chức năng tra cứu lương nhân viên*

****

***Hình 4.26:*** *Giao diện thực hiện chức năng cập nhật bảng chỉ số lương*

****

***Hình 4.27:*** *Giao diện thực hiện chức năng xem danh sách tiền lương nhân viên*

****

***Hình 4.28:*** *Kết quả khi đồng ý xuất file danh sách tiền lương nhân viên*

*(File Danh\_sach\_tien\_luong\_nhan\_vien.txt)*

**4.2.3. Nhận xét**

* Chương trình cho kết quả đúng yêu cầu.
* Đọc file input bằng cách nhập tên file từ bàn phím.
* Giao diện chính của chương trình chưa đặc biệt, đơn giản.

**5.  KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

**5.1. Kết luận**

Chương trình đáp ứng được nhu cầu của đề tài. Trong quá trình nghiên cứu và thực hiện đồ án, chúng em đã tìm hiểu và nắm vững những kiến thức liên quan như cấu trúc dữ danh sách liên kết kép, struct, lập trình hướng đối tượng…Sau quá trình thực hiện, chúng em đã hoàn thành được chương trình: “Xây dựng hệ thống quản lí nhân sự”. Tuy nhiên, thuật toán còn nhiều chỗ chưa thật sự tối ưu cũng như chưa được hoàn thiện một cách tốt nhất, giao diện đơn giản chưa được hoàn chỉnh.

**5.2. Hướng phát triển**

* Tiếp thu ý kiến, phát triển mã nguồn để đồ án được hoàn thiện.
* Phát triển chức năng nghỉ phép, bảng chấm công, quản lí hợp đồng nhân viên
* Xây dựng giao diện đồ họa hoàn chỉnh
* Tiếp tục phát triển thuật toán tối ưu, tạo giao diện hoàn chỉnh và rút ngắn thời gian debug của chương trình.
* Phát triển hệ thống ứng dụng vào hoàn cảnh thực tiễn.

**TÀI LIỆU THAM KHẢO**

# [1] Nguyễn Thanh Bình, Phạm Minh Tuấn, Đặng Thiên Bình, Phân tích và thiếtkế giải thuật, NXB GD, 2016.

# [2]Lê Thị Mỹ Hạnh, Giáo trình môn học Lập trình hướng đối tượng, 2002.

# [3] Lê Minh Hoàng, *Giải thuật & Lập trình*, Đại học Sư phạm Hà Nội, 2003.