

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI TP. HỒ CHÍ MINH**  
**KHOA CÔNG NGHỆ THÔNG TIN**



# **BÁO CÁO THỰC TẬP CƠ SỞ**

**Đề tài: Quản Lý Nhân Viên**

**Sinh viên thực hiện : Lê Quý Nhất - 1451120119**  
**Phan Chí Công -1451120083**

**Lớp : CN14B**

**Giáo viên hướng dẫn : TS Tô Bá Lâm**

***TP Hồ Chí Minh, tháng 06 năm 2017***

# MỤC LỤC

1	Giới thiệu chương trình: .....	1
2	Các chức năng của chương trình: .....	1
2.1	Đọc danh sách nhân viên từ file text.....	1
2.2	Sắp xếp nhân viên theo tiêu chí .....	1
2.3	Thêm nhân viên mới .....	2
2.4	Xóa nhân viên theo từ khóa .....	2
2.5	Tìm kiếm nhân viên .....	2
2.6	Xuất danh sách nhân viên ra file text.....	2
3	Cấu trúc chương trình: .....	2
3.1	Tổng quan các class có trong chương trình .....	2
3.2	Cấu trúc từng class.....	3
3.2.1	Class Date .....	3
3.2.2	Class Employee .....	3
3.2.3	Class Node.....	4
3.2.4	Class LinkedList .....	4
3.2.5	Class MyProcess.....	5
3.2.6	Class frmEmployeeManagement.....	6
3.2.7	Class Program.....	7
4	Quy trình hoạt động:.....	7
4.1	Thêm nhân viên mới .....	7
4.2	Xóa nhân viên .....	8
4.3	Sắp xếp nhân viên .....	8
4.4	Tìm kiếm nhân viên .....	8
4.5	Đọc dữ liệu từ file text .....	9
4.6	Xuất dữ liệu ra file text .....	9

## 1 Giới thiệu chương trình:

STT	Họ tên	Chức vụ	Ngày sinh	Hệ số lương
1	Lê Quý Nhất	Tổng giám đốc	24/10/1996	10
2	Phan Chí Công	Giám đốc	29/05/2017	8.5
3	Võ Thị Tuyết Vân	Phó giám đốc	05/07/1995	7.5
4	Nguyễn Văn Anh	Thư ký	01/06/1994	3.5
5	Phạm Nguyễn Quỳnh T...	Thư ký	28/04/1987	6
6	Châu Thị Kim Anh	Thư ký	14/07/1995	5
7	Võ An Phước Thiện	Trưởng phòng	25/01/1989	5
8	Nguyễn Thị Thanh Bích	Trưởng phòng	02/06/1993	5
9	Phan Huỳnh Ngọc Dung	Phó trưởng phòng	21/06/1995	4
10	Lê Minh Vương	Phó trưởng phòng	15/07/1993	4
11	Trần Thị Kim Ngân	Nhân viên	06/05/1993	2.5
12	Vương Thu Hiền	Nhân viên	07/07/1993	3
13	Bùi Kim Quyên	Nhân viên	08/07/1991	2.5
14	Nguyễn Thế Vinh	Nhân viên	14/07/1994	3
15	Dương Hoài Phương	Nhân viên	12/03/1987	2.5
16	Đoàn Thị Mỹ Xuân	Nhân viên	11/05/1993	3

### Giao diện chương trình

- Chương trình quản lý nhân viên của một công ty, được tổ chức và lưu trữ sử dụng cấu trúc dữ liệu Linked List.
- Chương trình cho phép thực hiện:
  - Đọc dữ liệu từ file text.
  - Sắp xếp danh sách nhân viên theo thứ tự với tiêu chí sắp xếp được nhập vào.
  - Chèn một nhân viên vào danh sách theo đúng tiêu chí xác định ở trên.
  - Xóa một hoặc nhiều nhân viên có liên quan đến từ khóa nhập vào.
  - Tìm kiếm nhân viên trong danh sách.
  - Xuất danh sách nhân viên ra file text.

## 2 Các chức năng của chương trình:

### 2.1 Đọc danh sách nhân viên từ file text

Đọc file text lấy danh sách nhân viên đưa vào danh sách liên kết.

### 2.2 Sắp xếp nhân viên theo tiêu chí

Sắp xếp danh sách nhân viên trong danh sách liên kết theo tiêu chí lựa chọn.

Các tiêu chí bao gồm: họ tên, chức vụ, ngày sinh, hệ số lương.

### **2.3 Thêm nhân viên mới**

Thêm một nhân viên vào danh sách liên kết theo đúng thứ tự sắp xếp với tiêu chí được lựa chọn ở trên.

### **2.4 Xóa nhân viên theo từ khóa**

Cho phép xóa một hoặc nhiều nhân viên liên quan tới từ khóa nhập vào.

Ví dụ: khi nhập vào 1992 thì sẽ xóa tất cả các nhân viên có năm sinh là 1992 hoặc khi nhập vào là Tâm thì sẽ xóa tất cả các nhân viên có chuỗi “Tâm” trong trường Họ và Tên.

### **2.5 Tìm kiếm nhân viên**

Tìm kiếm nhân viên thông qua từ khóa và trả về thông tin đầy đủ của nhân viên cũng như vị trí của nhân viên đó trong danh sách.

Ví dụ: khi nhập vào Tổng giám đốc thì trả về thông tin đầy đủ của nhân viên bao gồm họ và tên, chức vụ (là tổng giám đốc), ngày tháng năm sinh và hệ số lương cũng như vị trí của nhân viên trong danh sách nhân viên.

### **2.6 Xuất danh sách nhân viên ra file text**

Xuất danh sách nhân viên trong danh sách liên kết ra file text, mỗi nhân viên là một dòng.

## **3 Cấu trúc chương trình:**

### **3.1 Tổng quan các class có trong chương trình**

Các class được sử dụng trong chương trình bao gồm:

- Class Date
- Class Employee
- Class Node
- Class LinkedList
- Class MyProcess
- Class frmEmployeeManagement
- Class Program

## 3.2 Cấu trúc từng class

### 3.2.1 Class Date

Class Date
- int day - int month - int year
+ get() + set() + bool ValidateDate(int day, int month, int year) + Date(int day, int month, int year) + Date ParseDate(string dateString) + int CompareTo(Date objDate) + string ToString()

- Class Date dùng để lưu trữ thông tin ngày tháng năm.
- `private static bool ValidateDate(int day, int month, int year)`: kiểm tra ngày tháng có hợp lệ hay không.
  - `public Date(int day, int month, int year)`: khởi tạo đối tượng ngày.
  - `public static Date ParseDate(string dateString)`: chuyển chuỗi dạng dd/mm/YYYY sang kiểu Date.
  - `public int CompareTo(Date objDate)`: so sánh 2 đối tượng date với nhau.
  - `public override string ToString()`: chuyển đối tượng Date sang kiểu string.

### 3.2.2 Class Employee

Class Employee
- string name - string office - Date birthday - double salary
+ get() + set() + Employee() + Employee(string name, string office, Date birthday, double salary) + Employee(Employee e) + string Display() + bool Equals(Employee objEmployee)

- Class Employee dùng để lưu trữ thông tin nhân viên.
- `public Employee()`: khởi tạo đối tượng nhân viên.
  - `public Employee(string name, string office, Date birthday, double salary)`: khởi tạo đối tượng nhân viên có tham số truyền vào.

- `public Employee(Employee e):` khởi tạo đối tượng nhân viên từ một đối tượng nhân viên khác.
- `public string Display():` dùng để trả về thông tin tóm tắt của một nhân viên.

### 3.2.3 Class Node

Class Node
- Employee data - Node pNext
+ get() + set() + Node() + Node(Employee objEmployee)

➤ Class Node dùng để lưu trữ thông tin một node.

- `public Node():` khởi tạo một node.
- `public Node(Employee objEmployee):` Khởi tạo một node với đối tượng nhân viên truyền vào

### 3.2.4 Class LinkedList

Class LinkedList
- Node pHead - Node pTail
+ get() + set() + LinkedList() + void AddHead(ref LinkedList l, Node p) + void AddTail(ref LinkedList l, Node p) + void Insert(ref LinkedList l, Node employeeNode, string sortBy, string typeSort) + void RemoveHead(ref LinkedList l) + void RemoveTail(ref LinkedList l) + void RemoveAfterNode(ref LinkedList l, Node nodeRemove) + void Remove(ref LinkedList l, string keyword) + void RemoveNode(ref LinkedList l, Node nodeRemove) + void QuickSort(ref LinkedList l, bool typeSort, int sortBy) + void QuickSortBySalary(ref LinkedList l, string typeSort) + void QuickSortByBirthday(ref LinkedList l, string typeSort) + void QuickSortByOffice(ref LinkedList l, string typeSort) + void QuickSortByName(ref LinkedList l, string typeSort) + void Sort(ref LinkedList l, string typeSort, string sortBy) + void CopyNodeAddToList(ref LinkedList listAdd, Node nodeCopy) + LinkedList Search(LinkedList l, string keyword) + int IndexOf(Employee objEmployee) + void EmptyList()

➤ Class LinkedList dùng để lưu trữ danh sách nhân viên và các thao tác xử lý trên danh sách liên kết.

- `public LinkedList()`: khởi tạo danh sách.
- `public void AddHead(Node p)`: thêm node vào đầu danh sách liên kết.
- `public void AddTail(Node p)`: thêm node vào cuối danh sách liên kết.
- `public void Insert(Node employeeNode, string sortBy, string typeSort)`: thêm node vào danh sách theo tiêu chí sắp xếp truyền vào.
- `public void RemoveHead()`: xóa node đầu tiên trong danh sách.
- `public void RemoveTail()`: xóa node cuối cùng trong danh sách.
- `public void RemoveNode(Node nodeRemove)`: xóa một node bất kỳ trong danh sách.
- `public void RemoveAfterNode(Node nodeRemove)`: xóa node sau một node cho trước.
- `public void Remove(string keyword)`: xóa node dựa vào từ khóa truyền vào.
- `public void QuickSortBySalary(string typeSort)`: sắp xếp danh sách liên kết theo tiêu chí hệ số lương.
- `public void QuickSortByBirthday(string typeSort)`: sắp xếp danh sách liên kết theo tiêu chí ngày sinh.
- `public void QuickSortByOffice(string typeSort)`: sắp xếp danh sách liên kết theo tiêu chí chức vụ.
- `public void QuickSortByName(string typeSort)`: sắp xếp danh sách liên kết theo tiêu chí họ tên.
- `public void Sort(string typeSort, string sortBy)`: sắp xếp danh sách liên kết theo tiêu chí truyền vào.
- `public void CopyNodeAddToList(ref LinkedList listAdd, Node nodeCopy)`: copy dữ liệu của một node sang một danh sách liên kết khác.
- `public LinkedList Search(string keyword)`: tìm kiếm nhân viên trong danh sách liên kết dựa vào từ khóa.
- `public int IndexOf(Employee objEmployee)`: lấy vị trí của một nhân viên trong danh sách.
- `public void EmptyList()`: xóa toàn bộ dữ liệu trong danh sách liên kết.

### 3.2.5 Class MyProcess

Class MyProcess
+ string[] arrOffice
+ MyProcess() + string GetInfoEmployeeByKeyword(LinkedList l, string keyword) + void ImportDataFromFile(ref LinkedList l, string filePath) + void LoadToDataGridView(DataGridView dg, LinkedList list) + void LoadSearchResultToDataGridView(DataGridView dg, LinkedList currentList, LinkedList resultList) + int GetIndexOfArray(string office) + string GetNameSelected(int selectedIndex) + string[] ReverseArrayString(string[] arrStrings) + int CompareTwoString(string firstString, string secondString) + void SaveFile(LinkedList l, string filePath)

- Class MyProcess dùng xử lý các tác vụ cần thiết trong chương trình.
- `public MyProcess()`: khởi tạo đối tượng xử lý và mảng chức vụ.

- `public void ImportDataFromFile(ref LinkedList l, string filePath)`: lấy dữ liệu danh sách nhân viên đưa vào danh sách liên kết.
- `public void LoadToDataGridView(DataGridView dg, LinkedList list)`: lấy dữ liệu từ danh sách liên kết hiển thị lên datagridview.
- `public void LoadSearchResultToDataGridView(DataGridView dg, LinkedList currentList, LinkedList resultList)`: lấy dữ liệu từ danh sách kết quả khi tìm kiếm hiển thị lên datagridview.
- `public string GetInfoEmployeeByKeyword(LinkedList l, string keyword)`: hàm lấy thông tin nhân viên trước khi xóa.
- `public int GetIndexOfArray(string office)`: lấy chỉ số của chức vụ trong mảng chứa thông tin chức vụ.
- `public string GetNameSelected(int selectedIndex)`: lấy tên tiêu chí cần sắp xếp dựa vào chỉ số lấy từ form.
- `public string[] ReverseArrayString(string[] arrStrings)`: đảo ngược mảng chứa các từ.
- `public int CompareTwoString(string firstString, string secondString)`: so sánh 2 chuỗi với nhau.
- `public void SaveFile(LinkedList l, string filePath)`: lưu danh sách nhân viên xuống file text.

### 3.2.6 Class frmEmployeeManagement

Class frmEmployeeManagement
<ul style="list-style-type: none"><li>- LinkedList l</li><li>- MyProcess process</li><li>- string filePath</li><li>- bool isSaved</li></ul>
<ul style="list-style-type: none"><li>+ frmEmployeeManagement()</li><li>+ bool ValidateData()</li><li>+ frmEmployeeManagement_Load(object sender, EventArgs e)</li><li>+ void btnThem_Click(object sender, EventArgs e)</li><li>+ void btnThucHienSapXep_Click(object sender, EventArgs e)</li><li>+ void btnThucHienTimKiem_Click(object sender, EventArgs e)</li><li>+ void btnThucHienXoa_Click(object sender, EventArgs e)</li><li>+ void btnLamMoi_Click(object sender, EventArgs e)</li><li>+ void btnLuuLai_Click(object sender, EventArgs e)</li><li>+ void frmEmployeeManagement_FormClosing(object sender, FormClosingEventArgs e)</li><li>+ void ResetInput()</li></ul>

- Class frmEmployeeManagement dùng xử lý giao diện form.
- `private bool ValidateData()`: kiểm tra tính hợp lệ của dữ liệu nhập vào khi thêm nhân viên mới.
  - `private void frmEmployeeManagement_Load(object sender, EventArgs e)`: xử lý form khi được load.
  - `private void btnThem_Click(object sender, EventArgs e)`: xử lý thực hiện thêm một nhân viên mới.



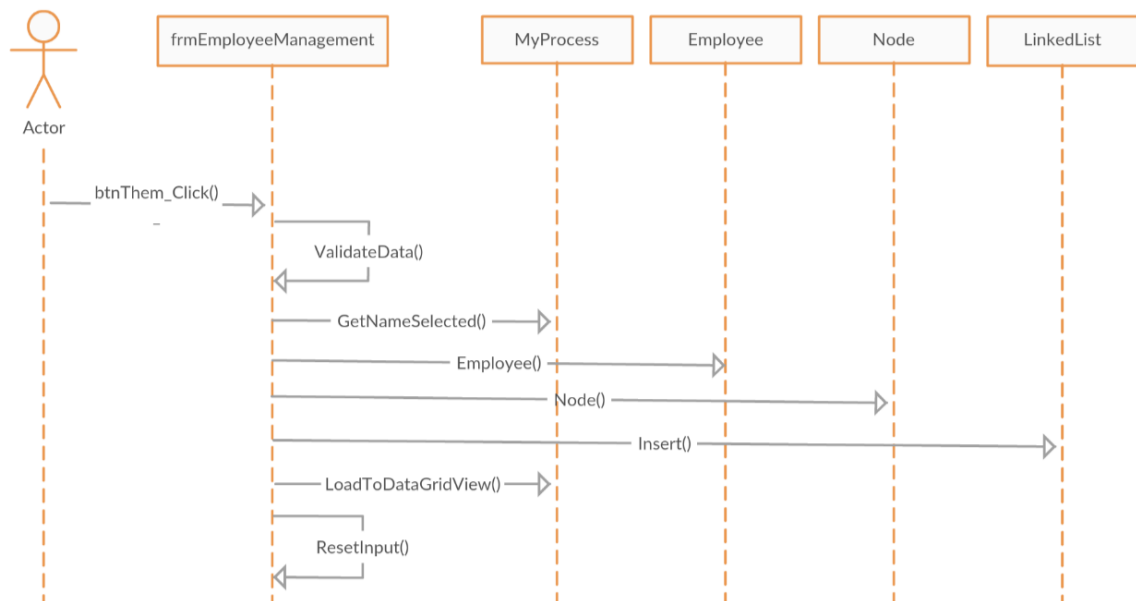
- `private void btnThucHienSapXep_Click(object sender, EventArgs e)`: xử lý khi sắp xếp danh sách nhân viên.
- `private void btnThucHienTimKiem_Click(object sender, EventArgs e)`: xử lý khi thực hiện tìm kiếm nhân viên.
- `private void btnThucHienXoa_Click(object sender, EventArgs e)`: xử lý khi thực hiện xóa nhân viên.
- `private void btnLamMoi_Click(object sender, EventArgs e)`: tải lại danh sách nhân viên.
- `private void btnLuuLai_Click(object sender, EventArgs e)`: xử lý lưu danh sách nhân viên xuống file.
- `private void frmEmployeeManagement_FormClosing(object sender, FormClosingEventArgs e)`: xử lý trước khi đóng chương trình.
- `private void ResetInput()`: xóa nội dung trong các ô input khi thêm nhân viên mới.

### 3.2.7 Class Program

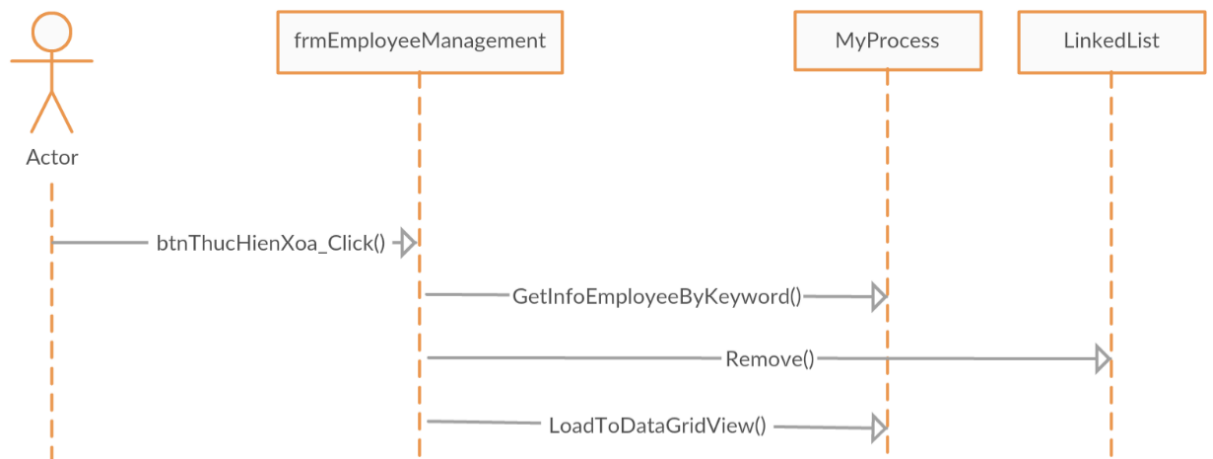
Dùng để chạy chương trình.

## 4 Quy trình hoạt động:

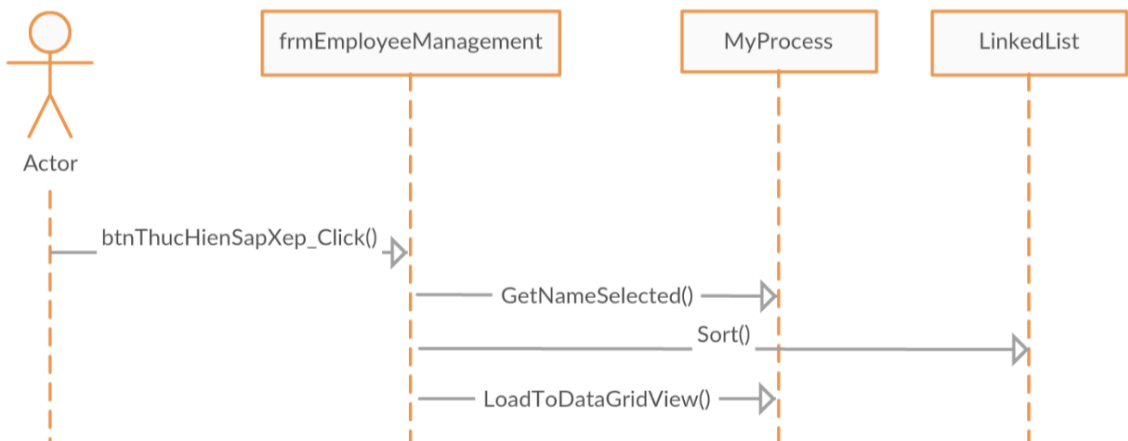
### 4.1 Thêm nhân viên mới



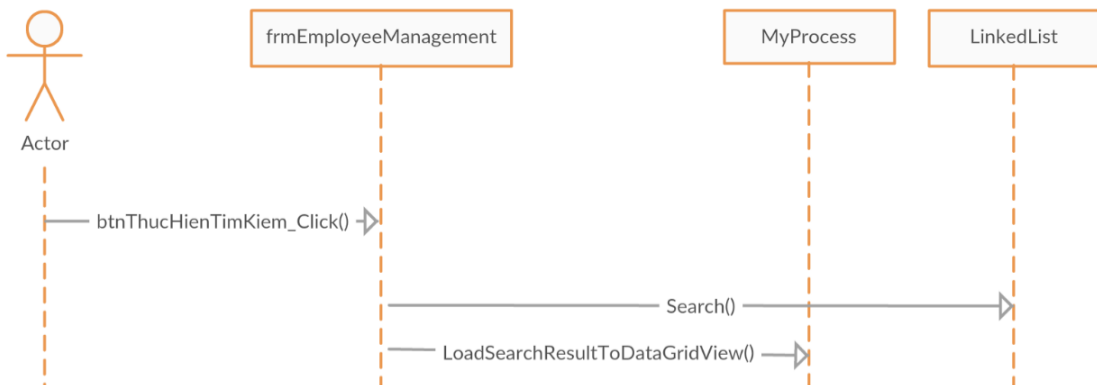
## 4.2 Xóa nhân viên



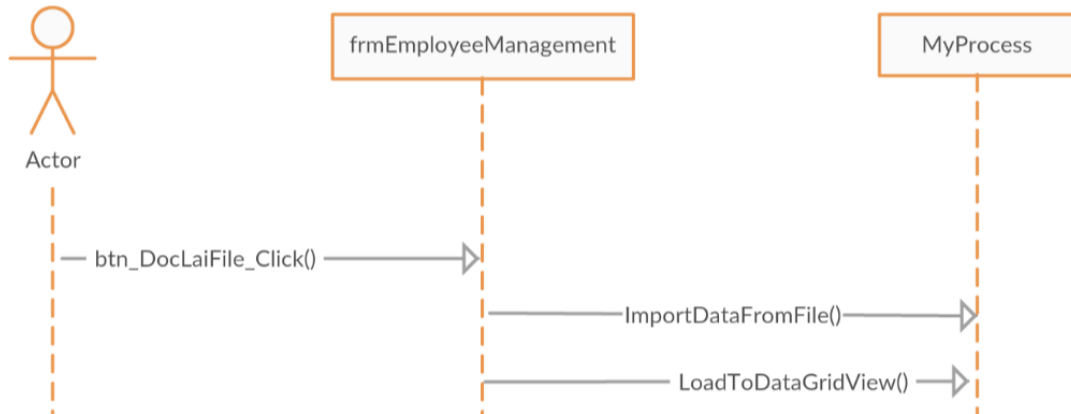
## 4.3 Sắp xếp nhân viên



## 4.4 Tìm kiếm nhân viên



#### 4.5 Đọc dữ liệu từ file text



#### 4.6 Xuất dữ liệu ra file text

