

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI TP.HCM

KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO THỰC TẬP CƠ SỞ



Nhóm sinh viên : Nguyễn Ngọc Vương (leader)

Lê Thị Mỹ Hòa

Lớp : CN14B

Người hướng dẫn: Thầy Tô Bá Lâm

I. Giới thiệu bài toán:

1. Yêu cầu bài toán

Một công ty có nhu cầu quản lý nhân viên của mình.

Mỗi nhân viên có thông tin cần quản lý như sau:

- Họ và tên
- Chức vụ
- Ngày tháng năm sinh
- Hệ số lương

Dữ liệu các nhân viên sẽ được tổ chức và lưu trữ sử dụng cấu trúc dữ liệu **Linked list**

Anh/chị hãy viết chương trình cho phép thực hiện các thao tác sau:

1/ Đọc từ file text danh sách các nhân viên

2/ Sắp xếp các nhân viên theo thứ tự với tiêu chí sắp xếp sẽ được nhập vào

Ví dụ: tiêu chí: ngày tháng năm sinh, hệ số lương hay chức vụ

3/ Chèn một nhân viên vào danh sách theo đúng thứ tự với tiêu chí đã được xác định ở trên

4/ Xóa một hoặc nhiều nhân viên có thông tin liên quan đến từ khóa

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

5/ Tìm kiếm một nhân viên và trả về thông tin đầy đủ của nhân viên cũng như vị trí của nhân viên 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

6/ xuất toàn bộ danh sách nhân viên ra file text. Mỗi nhân viên là một dòng.

Định dạng file xuất như sau:

Họ và tên	Chức vụ	Ngày tháng năm sinh	Hệ số lương
Nguyễn văn A	Thư ký	10/11/1980	2.5

II. Tóm tắt các phần:

1. Chức năng:

- Đọc dữ liệu nhân viên từ file text.
- Sắp xếp nhân viên.
- Thêm nhân viên.
- Xóa nhân viên.
- Tìm kiếm nhân viên.
- Xuất danh sách nhân viên ra file text.

2. Các class và các hàm sử dụng:

1.1) Class Employee: Dùng để lưu trữ thông tin nhân viên.

Các phương thức:

- + `private bool Employee()`: Khởi tạo một đối tượng nhân viên.
- + `public bool Equals(Employee employee)`: So sánh hai đối tượng nhân viên có giống nhau không.
- + `public Employee Clone()`: Tạo một bản sao nhân viên.
- + `public int IndexOfPosition(string[] positionArray)`: Lấy chỉ số chức vụ của nhân viên dựa trên một mảng quy định sẵn.

1.2) Class Date: dùng để lưu trữ thông tin ngày giờ.

Các phương thức:

- + `public Date()`: Khởi tạo một đối tượng ngày giờ mặc định.
- + `public Date(int day, int month, int year)`: Khởi tạo một đối tượng ngày giờ từ các tham số ngày tháng năm.
- + `public static int NumberDayOfMonth(int month, int year)`: Lấy ra số ngày của tháng thuộc một năm nào đó.
- + `public static Date Parse(string stringDate)`: Chuyển một string sang kiểu Date.
- + `public int CompareTo(Date date)`: So sánh 2 đối tượng Date.

1.3) Class ListEmployee (Linked list): dùng để lưu trữ danh sách nhân viên.

Các phương thức:

- + `public int Count()`: Đếm số nhân viên có trong danh sách.
- + `public void Display()`: Hiển thị danh sách nhân viên lên giao diện.
- + `public bool Contains(Node<Employee> employee)`: Kiểm tra danh sách có chứa một nhân viên nào đó hay không.
- + `private void Swap(Node<Employee> emp1, Node<Employee> emp2)`: Hoán vị thông tin hai nhân viên trong danh sách.
- + `public void AddLast(Node<Employee> employee)`: Thêm 1 nhân viên vào cuối danh sách.
- + `public void AddFirst(Node<Employee> employee)`: Thêm 1 nhân viên vào đầu danh sách.
- + `private void SortEmployeeByPosition(int typeSort)`: Sắp xếp danh sách nhân viên theo chức vụ.

- + **private void** SortEmployeeByBirthDay(**int** typeSort): Sắp xếp danh sách nhân viên theo ngày sinh(tuổi).
- + **private void** SortEmployeeByCoefficientSalary(**int** typeSort): Sắp xếp danh sách nhân viên theo hệ số lương.
- + **private void** SortEmployeeByName(**int** typeSort): Sắp xếp danh sách nhân viên theo tên.
- + **public void** Sort(**string** propertyName, **int** typeSort): Gọi các hàm sắp xếp ở trên tùy vào propertyName truyền vào.
- + **public void** Remove(**Node**<**Employee**> employee = **null**): Xóa một nhân viên trong danh sách.
- + **public void** RemoveMultiple(**string** keyword): Xóa những nhân viên có thông tin liên quan đến từ khóa(keyword).
- + **public Node**<**Employee**> SearchFirst(**string** keyword): Tìm nhân viên đầu tiên có thông tin liên quan đến từ khóa.
- + **public ListEmployee** SearchMultiple(**string** keyword): Tìm tất cả nhân viên trong danh sách có thông tin liên quan đến từ khóa.
- + **public int** IndexOf(**Node**<**Employee**> employee): Lấy ra vị trí của nhân viên trong danh sách.

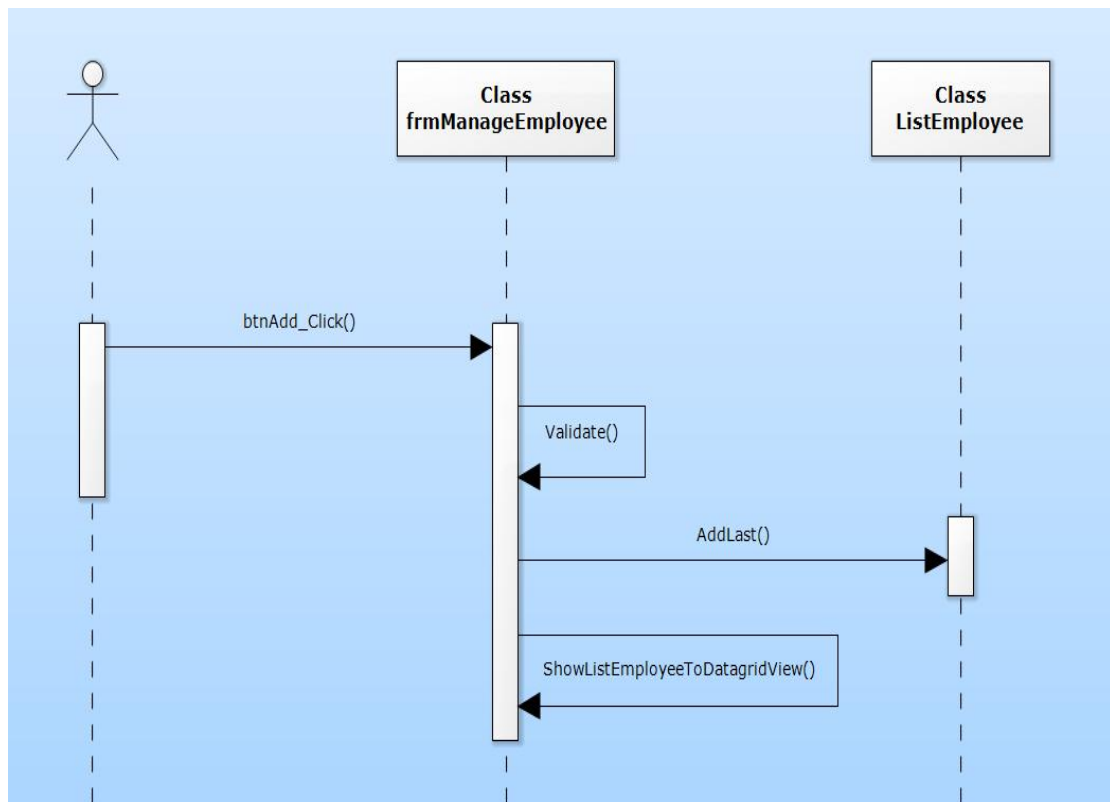
1.4) Class frmManageEmployee: Xử lý giao diện.

Các phương thức:

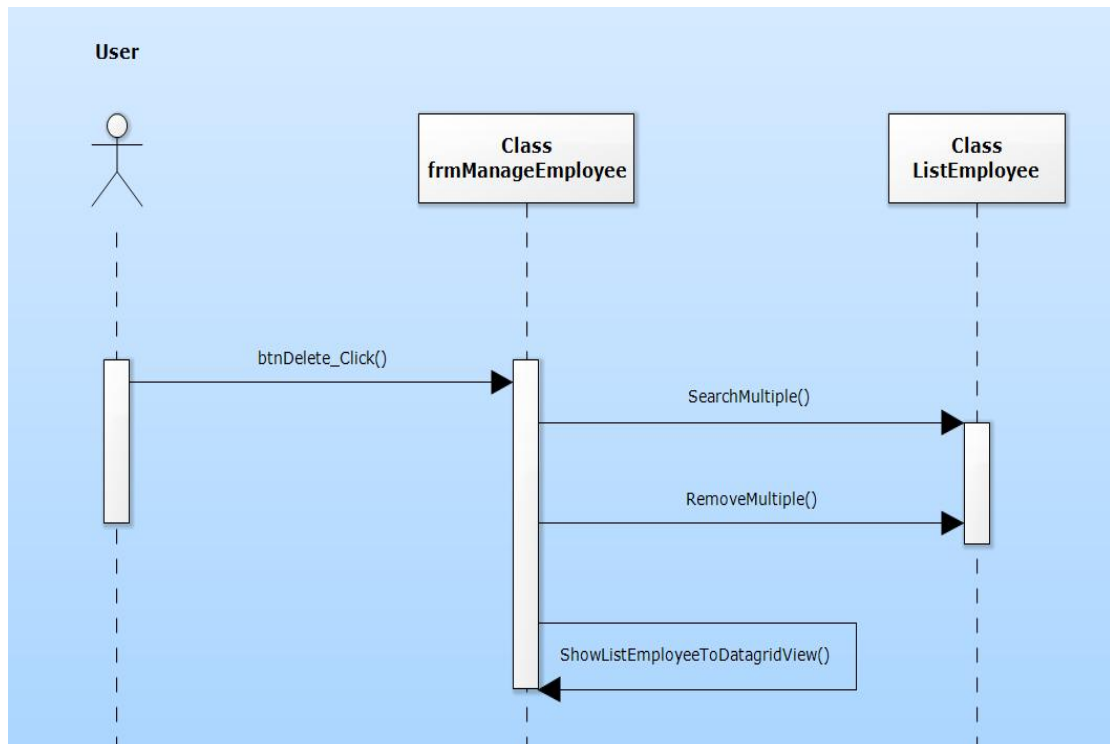
- + **public** frmManageEmployee(): Khởi tạo giao diện(form quản lý nhân viên).
- + **private void** Form1_Load(**object** sender, **EventArgs** e): Xử lý khi form được load.
- + **public bool** ValidateForm(): Kiểm tra tính hợp lệ của dữ liệu trước khi thêm 1 nhân viên.
- + **private void** ShowListEmployeeToDataGridView(**ListEmployee**): Hiển thị danh sách nhân viên lên giao diện.
- + **public string** EmployeeRelateWithKeyword(**string** keyword): Lấy thông tin các nhân viên có liên quan đến từ khóa.
- + **private string** GetPropertyToSort(): Lấy thuộc tính cần sắp xếp trước khi sắp xếp.
- + **private void** WriteToFile(**StreamWriter** sw): Ghi thông tin nhân viên từ danh sách nhân viên ra file text.
- + **private void** LoadDataFromFile(**string** filePath): Đọc dữ liệu từ filePath truyền vào và hiển thị lên giao diện.
- + **private void** btnSort_Click(**object** sender, **EventArgs** e): Thực hiện sắp xếp danh sách nhân viên.
- + **private void** btnAdd_Click(**object** sender, **EventArgs** e): Thực hiện thêm 1 nhân viên.
- + **private void** btnDelete_Click(**object** sender, **EventArgs** e): Thực hiện xóa nhân viên.
- + **private void** btnSearch_Click(**object** sender, **EventArgs** e): Thực hiện tìm kiếm nhân viên.
- + **private void** btnSave_Click(**object** sender, **EventArgs** e): Thực hiện xuất danh sách nhân viên ra file text.
- + **private void** btnRefresh_Click(**object** sender, **EventArgs** e): Đọc lại file và hiển thị lại thông tin lên giao diện.

3. Quan hệ giữa các hàm:

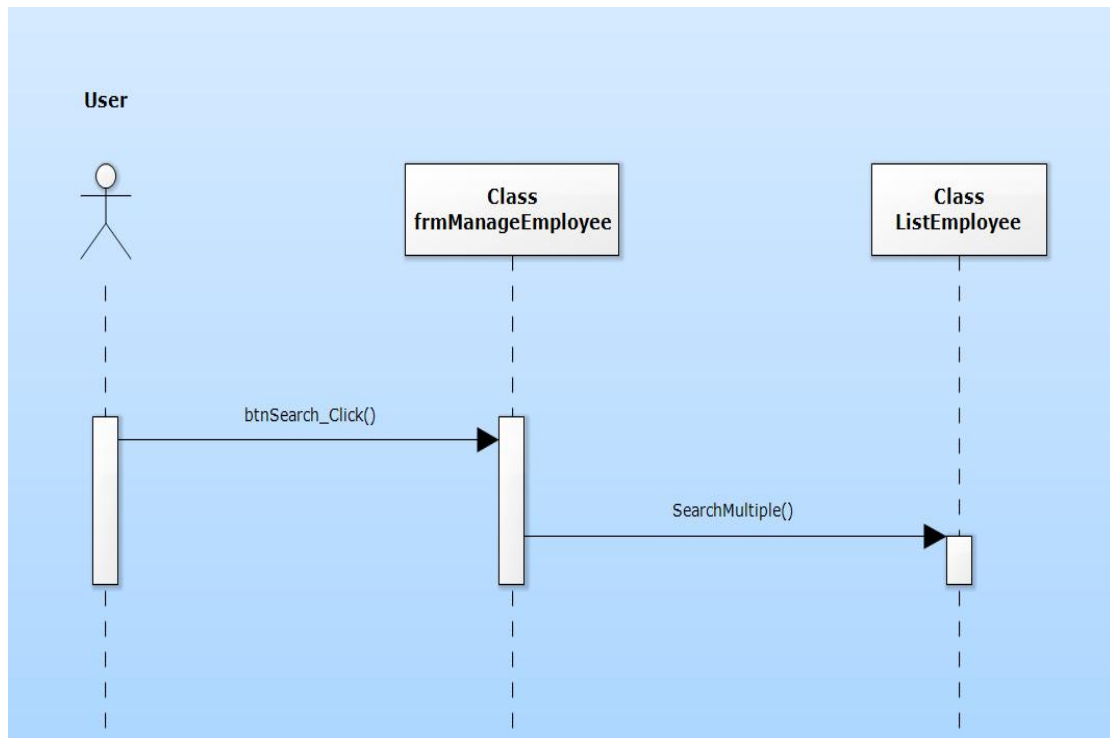
- Thao tác thêm nhân viên



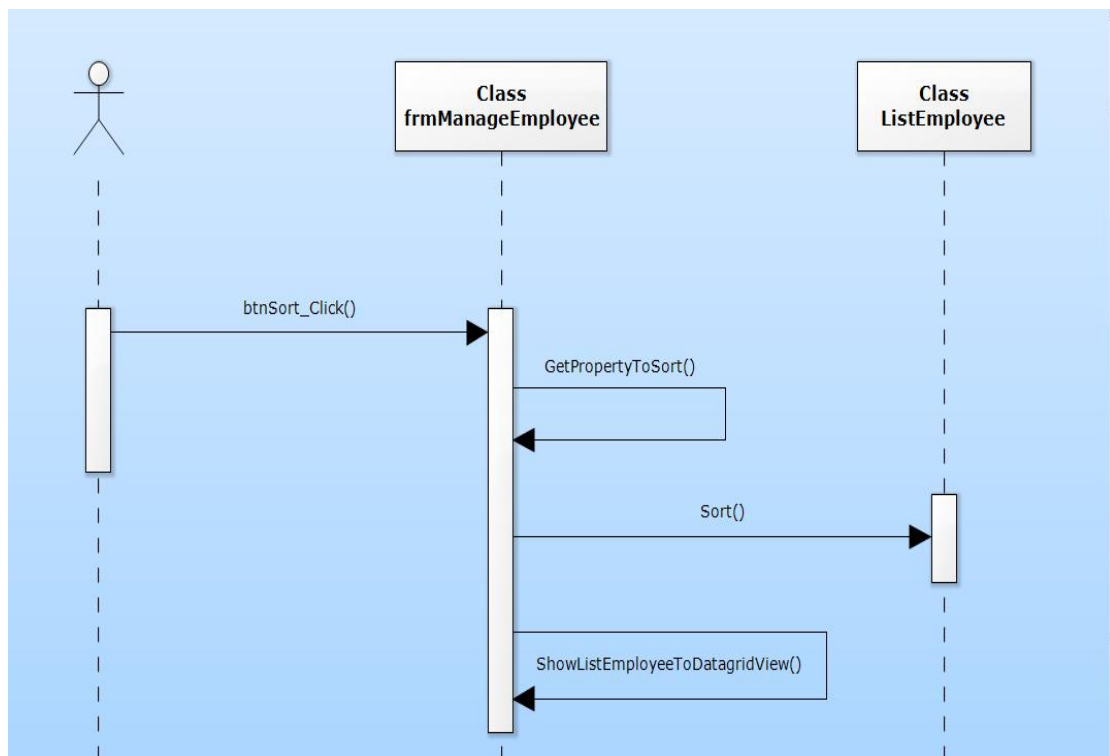
- Thao tác xóa nhân viên



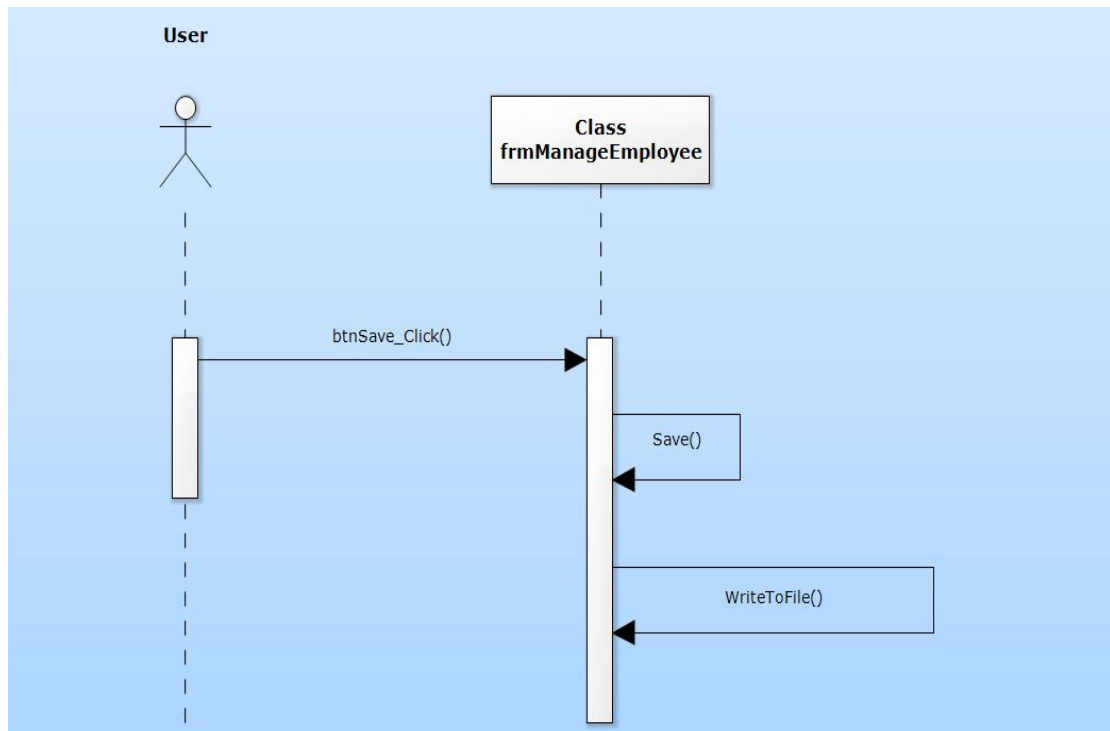
- Thao tác tìm kiếm



- Thao tác sắp xếp



- Thao tác xuất file



- Thao tác làm mới danh sách

