

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI TP HCM

BÁO CÁO ĐỒ ÁN
MÔN THỰC TẬP CƠ SỞ

ĐỀ TÀI: QUẢN LÝ NHÂN VIÊN

GVHD: TÔ BÁ LÂM
CÁC THÀNH VIÊN

Nguyễn Thanh Huy - 1451120025 - CN14A

Nguyễn Quý Tùng - 1451120062 - CN14A

CẤU TRÚC BÀI BÁO CÁO

- Trình bày tóm tắt nội dung đã làm.
- Trình bày các class.
- Trình bày các hàm trong class.
- Mối liên hệ.

PHẦN I : TÓM TẮT NỘI DUNG ĐÃ LÀM

Nhóm chúng em thực hiện đồ án trên ứng dụng dạng Console (Console Application), với ngôn ngữ lập trình là C#.

Theo đề bài , nhóm chúng em đã thực hiện được những chức năng sau:

1. Đọc file text danh sách các nhân viên theo định dạng (Họ và tên - Chức vụ - Ngày tháng năm sinh - Hệ số Lương).
2. Sắp xếp các nhân viên theo các tiêu chí: Ngày tháng năm sinh, Hệ số Lương và Chức vụ. Có thể sắp xếp tăng dần hoặc giảm dần.
3. Chèn một nhân viên vào danh sách theo đúng thứ tự với tiêu chí đã được xác định ở trên. Yêu cầu phải sắp xếp trước khi thực hiện chức năng này.

4. Xóa một hoặc nhiều nhân viên có thông tin liên quan đến từ khóa. Từ khóa trong hai trường là Họ tên và Năm sinh.
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. Từ khóa trong các trường sau: Họ tên, Chức vụ, Năm sinh, Ngày tháng năm sinh (dd/MM/yyyy).
6. Xuất toàn bộ danh sách nhân viên ra file text. Mỗi nhân viên là một dòng.

PHẦN II : CLASS

Source code gồm các class sau:

- Class `Node`: class đại diện cho các đối tượng (Nhân viên) của danh sách liên kết đơn. Gồm các thuộc tính:
 - ♣ `string` name, job;
 - ♣ `DateTime` birthDay;
 - ♣ `double` numOfSalary; //hệ số lương
 - ♣ `Node` next;
- Class `ListOfNode`: class danh sách liên kết đơn. Bao gồm các hàm thực hiện chức năng.

- Class `Solution`: class in ra màn hình các chức năng, yêu cầu người dùng nhập tham số để từ đó gọi các hàm trong class `ListOfNode` và in ra kết quả.
- Class `FunctionConstant`: class xây dựng các hàm ít thay đổi và dùng nhiều như kiểm tra có phải số, đọc ghi file ...

PHẦN III: CÁC HÀM TRONG CLASS

⌘ Class `ListOfNode` gồm các hàm sau:

`private void` `sendDataOfNode(Node from, Node to)`: hàm thực hiện truyền dữ liệu từ node from sang node to.

`public void` `addHead(Node node)`: hàm chèn node vào đầu danh sách.

`public void` `addTail(Node node)`: hàm chèn node vào cuối danh sách.

`private int` `countNode()`: hàm trả về tổng các nhân viên.

`private int posOfNode(Node n)`: hàm trả về vị trí của n, nếu không có trong danh sách trả về -1.

`public void insertNodeInPos(Node node, int position)`: hàm chèn node vào danh sách tại vị trí position. Nếu position > countNode() hoặc position < 0 thì sẽ không thực hiện.

`public void showList(string test = null)`: hàm in danh sách ra, tham số test có thể null, nếu có tham số test thì sẽ in ra Hiển thị danh sách theo “test”.

`public bool fileTxtIsEmpty()`: sau khi đọc file sẽ kiểm tra file có rỗng hay không, bằng cách kiểm tra Head.

`private void showOneNodeForDelete(Node n, ref int count)`: hàm in ra node xóa, đồng thời tham số count có thể thay đổi được (ref). Tham số count thể hiện tổng số nhân viên bị xóa khỏi danh sách.

`private void swap(Node temp1, Node temp2)`: hàm hoán vị 2 node, sử dụng hàm *sendDataOfNode*.

`private void showOneNode(Node node, int pos)`: hàm hiển thị 1 node tại vị trí pos.

- Các hàm sắp xếp, chèn, xóa

`public void sortByBirthDay(int status)`: hàm sắp xếp danh sách theo ngày tháng năm sinh. <status 0 là tăng dần, 1 là giảm dần>

`public void sortByNumOfSalary(int status)`: hàm sắp xếp theo hệ số lương. <status 0 là tăng dần, 1 là giảm dần>

`public void sortByJob(int status)`: hàm sắp xếp theo chức vụ. <status 0 là tăng dần, 1 là giảm dần>

`public void insertNodeBySort(Node node, int sortByWhat, int status)`: hàm chèn node vào danh sách theo tiêu chí đã sắp xếp. <sortByWhat 1: birth, 2: numSalary, 3: job>

`public void delete_byKeyPress(string key)`: hàm xóa theo từ khóa (chỉ xét họ tên và năm sinh).

- Các hàm tìm kiếm nhân viên

`public void findStaff(string key)`: hàm tìm kiếm nhân viên theo từ khóa key. Nếu key là số thì tìm theo hệ số lương và tìm theo năm sinh. Nếu key là chuỗi thì xét có ký tự “/” thì tìm theo ngày tháng năm sinh, ngược lại tìm theo họ tên và chức vụ.

`private bool checkFindBirthDay(DateTime date)`: hàm kiểm tra có ngày tháng năm sinh nào trong danh sách trùng với tham số date.

`private bool checkFindName(string key)`: hàm kiểm tra có họ tên nào trong danh sách trùng với tham số key.

`private bool checkFindJob(string key)`: hàm kiểm tra có chức vụ nào trong danh sách trùng với tham số key.

`private bool checkFindNumSalary(int num)`: hàm kiểm tra có hệ số lương nào trong danh sách trùng với tham số num.

`private bool checkFindYearOfBD(int num)`: hàm kiểm tra có năm sinh nào trong danh sách trùng với tham số num.

`private void findByNumSalary(int num)`: hàm hiển thị các node có hệ số lương bằng tham số num.

`private void findByBirthDay(string key, int num = 0)`: nếu num mặc định (không truyền num thì num = 0) thì hiển thị các node có ngày tháng năm sinh trùng với key. Ngược lại hiển thị các node có năm sinh trùng với num.

`private void findByName(string key)`: hàm hiển thị các node có họ tên trùng với tham số key.

`private void findByJob(string key)`: hàm hiển thị các node có chức vụ trùng với tham số key.

⌘ Class *FunctionConstant* gồm các hàm sau:

`public static bool IsNumber(string pValue)`: hàm kiểm tra pValue có phải là số hay không.

`public static string titleShowList(int sortByWhat, int status)`: hàm trả về chuỗi “=>Theo...”, đây là biến test trong hàm showList ở class ListOfNode.

`public static void dataDefault(ListOfNode list)`: hàm lưu dữ liệu mặc định vào danh sách liên kết (list) khi file bị lỗi hoặc file trống.

`public static bool` readFileText(`ListOfNode` list):
hàm đọc file text và lưu vào danh sách liên kết
(list).

`public static void` writeFileText(`ListOfNode` list):
hàm ghi danh sách liên kết (list) xuống file text.

⌘ Class *Solution* gồm các hàm sau:

`public void` showList_0(): chức năng số 0, hiển thị
danh sách.

`public void` read_1(): chức năng số 1, đọc file text
và lưu vào danh sách.

`public void` sort_2(): chức năng số 2, sắp xếp danh
sách theo tiêu chí.

`public void` insert_3(): chức năng số 3, chèn vào
danh sách 1 node theo tiêu chí đã sắp xếp.

`public void` delete_4(): chức năng số 4, xóa những
nhân viên (node) theo từ khóa.

`public void find_5()`: chức năng số 5, tìm những nhân viên (node) theo từ khóa.

`public void writeFile_6()`: chức năng số 6, ghi danh sách xuống file text.

`private Node inputDataNode_3()`: hàm in ra màn hình yêu cầu người dùng nhập dữ liệu nhân viên (node) cần chèn của chức năng số 3.

`private int show_choseSort_2(string str)`: hàm in ra màn hình yêu cầu người dùng chọn sắp xếp tiêu chí theo thứ tự tăng dần hay giảm dần của chức năng số 2.

PHẦN IV: MỐI QUAN HỆ

*Khi người dùng chọn chức năng **đọc** file text vào danh sách nhân viên:*

Chương trình gọi hàm `read_1()` trong class `Solution`.

Hàm `read_1()` liên kết với các hàm:

- Class `FunctionConstant`: hàm đọc file text (`readFileText`), hàm chèn bộ dữ liệu có sẵn (`dataDefault`)
- Class `ListOfNode`: hàm kiểm tra danh sách có rỗng không (`fileTxtIsEmpty`), hàm hiển thị danh sách (`showList`)

Khi người dùng chọn chức năng sắp xếp danh sách nhân viên:

Chương trình gọi hàm `sort_2()` trong class `Solution`.

Hàm `sort_2()` liên kết với các hàm:

- Class `FunctionConstant`: hàm kiểm tra ký tự có phải số (`IsNumber`)
- Class `ListOfNode`: các hàm sắp xếp theo ngày sinh, chức vụ, hệ số lương (`sortByBirthDay`, `sortByJob`, `sortByNumOfSalary`), hàm hiển thị danh sách (`showList`)
- Class `Solution`: hàm chọn sắp xếp tăng dần hay giảm dần (`show_choseSort_2`)

Khi người dùng chọn chức năng chèn nhân viên theo tiêu chí đã sắp xếp ở câu trên:

Chương trình gọi hàm `insert_3()` trong class `Solution`.

Hàm `insert_3()` liên kết với các hàm:

- Class `ListOfNode`: hàm chèn nhân viên theo tiêu chí (`insertNodeBySort`), hàm hiển thị danh sách (`showList`)
- Class `Solution`: hàm nhập thông tin nhân viên (`inputDataNode_3`)

Khi người dùng chọn chức năng xóa nhân viên liên quan đến từ khóa:

Chương trình gọi hàm delete_4() trong class Solution.

Hàm delete_4() liên kết với các hàm:

- Class ListNode: hàm xóa nhân viên theo từ khóa (delete_byKeyPress)

Khi người dùng chọn chức năng tìm kiếm nhân viên theo từ khóa:

Chương trình gọi hàm find_5() trong class Solution.

Hàm find_5() liên kết với các hàm:

- Class ListNode: hàm tìm kiếm nhân viên (findStaff)

Khi người dùng chọn chức năng ghi danh sách nhân viên xuống file text:

Chương trình gọi hàm writeFile_6() trong class Solution.

Hàm writeFile_6() liên kết với các hàm:

- Class FunctionConstant: hàm ghi file (writeFileText)

Ngoài ra, các hàm chức năng trong class `ListOfNode` (hàm được gọi trong class `Solution`) còn liên kết với các hàm phụ trợ trong class `ListOfNode` (các hàm private)

Hàm tìm kiếm nhân viên liên kết với:

- Các hàm kiểm tra có nhân viên nào trong danh sách theo từ khóa (họ tên, chức vụ, ngày tháng năm sinh hay năm sinh).
- Các hàm hiển thị nhân viên đó theo họ tên, chức vụ, ngày tháng năm sinh, năm sinh.
- Hàm hiển thị không có nhân viên nào trùng với từ khóa.
- Hàm hiển thị từng nhân viên tại vị trí.