Cao đẳng thực hành FPT Polytechnic Hà Nội

A picture containing logo

Description automatically generated

Báo cáo môn cấu trúc dữ liệu và giải thuật

Giảng viên hướng dẫn: Chu Thị Ngân- nganct4

Lớp: IT17305

Sinh viên thực hiện: Bùi Doanh Thái

Ngày tạo ra bản đặc tả

# GIẢNG VIÊN HƯỚNG DẪN

Họ và tên: Chu Thị Ngân – Nganct4

Cơ quan công tác: Cao đẳng FPT Polytechnic

Điện thoại: 0916553410 Email: nganct4@fpt.edu.vn

Ý kiến nhận xét, đánh giá của cán bộ hướng dẫn:

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

|  |  |
| --- | --- |
| Giáo viên hướng dẫn  *(Ký và ghi rõ họ tên)* | Xác nhận của Bộ Môn  (Ký và ghi rõ họ tên) |
|  |
|  |
|  |

# LỜI CẢM ƠN

Để hoàn thành tốt đề tài này chúng em xin cảm ơn tới quý thầy cô trường CĐ FPT Polytechnic đã tạo điều kiện cho chúng em nghiên cứu và hoàn thành môn học. Tiếp đến, chúng em xin chân trọng cảm ơn giáo viên bộ môn Phân tích và thiết kế hệ thống thông tin cô Chu Thị Ngân đã tận tình giúp đỡ, chỉ bảo hướng dẫn chúng em trong suốt quá trình học. Vì thời gian và năng lực còn có hạn chế nên không thể tránh khỏi những sai sót trong khi thực hiên bài. Rất mong được sự góp ý bổ sung của cô để đề bài của chúng em ngày càng hoàn thiện hơn.

Em xin chân thành cảm ơn!

# Giới thiệu dự án

Ứng dụng các cấu trúc dữ liệu đã được học và các giải thuật tìm kiếm, sắp xếp, sinh viên viết chương trình quản lý nhân sự cho một công ty mà bạn đang làm việc, mỗi nhân viên trong công ty gồm các thông tin sau:mã số, họ, tên, ngày sinh, nơi sinh, địa chỉ, ngày công tác, lương….

# Thiết kế ứng dụng

## Các chức năng cần làm trong dự án

* Menu quản lý nhân viên
* Xuất danh sách quản lý nhân viên
* Tìm nhân viên theo tên
* Tìm nhân viên theo mã
* Sắp xếp danh sách sinh viên theo lương
* Xóa nhân viên theo mac nhân viên

## giải thích thư viện đã sử dụng trong dự án

* #include<stdio.h>: thư viện nhập xuất dữ liệu
* #include<string.h>: thu viện cắt chuỗi
* #include<stdlib.h>: thư viện cấp phát bộ nhớ

## CÁc Hàm và công Dụng

* nhap1nv(danhsach &ds) :nhập 1 nhân viên 🡺 sử dụng tham số đầu vào là một một con trỏ tới kiểu biến (danh sách)
* nhapnhieunv(danhsach &ds):nhập nhiều nhân viên 🡺con trỏ danh sách trỏ đến biến(danh sách)
* xuat1nv(nhanvien nv):xuất 1 nv
* xuatdsnv(danhsach &ds):xuất danh sách nhiều nv
* timtheoten(danhsach &ds,char\*c):
* timMANV (danhsach &ds,int d):
* sapxep(danhsach &ds):
* xoanv(danhsach &ds,int d):

2.3.1.0: Các đối tượng cần khai báo (struct Date, Struct NhanVien, danh sách)

A, Đối tượng date bao gồm : ngay, thang, nam

B, Đối tượng NhanVien bao gồm: masv, ho,ten,namsinh,noisinh,diachi,luong,ngayvao.

C, Đối tượng danh sách bao gồm: con trỏ : danh sachnhanvien sẽ trỏ đến đối tượng nhân viên , và số lượng nhân viên (int soLuong);

#### void nhap1nv(danhsach &ds , nhanvien &nv)

Giải thích: con trỏ ds trỏ đến biến danh sách

* Kiểm tra mã nhân viên đã tồn tại hay chưa
* Nếu đã tồn tại bắt nhập cho đến khi phù hợp điều kiện mã chưa tồn tại
* Sau đó nhập dữ liệu cho nhân viên

*// nếu mã nhân viên đã tồn tại bắt nhập lại*

  do

  {

    printf("\nNhap ma so nhan vien: ");

    scanf("%d", &nv.msnv);

    if (timMSNV(ds, nv.msnv) != -1)

    {

      printf("Ma so nhan vien da ton tai, vui long nhap lai!\n");

    }

  } while (timMSNV(ds, nv.msnv) != -1);

  fflush(stdin);

  printf("\nNhap ho nhan vien: ");

  gets(nv.ho); *// sử dụng gets thay vì scanf vì scanf không đọc được khoảng trắng*

  printf("\nNhap ten nhan vien: ");

  gets(nv.ten);

  printf("\nNhap nam sinh: ");

  nhapngaythang(nv.namsinh);

  fflush(stdin);

  printf("\nNhap noi sinh: ");

  gets(nv.noisinh);

  printf("\nNhap dia chi: ");

  gets(nv.diachi);

  printf("\nNhap luong: ");

  scanf("%f", &nv.luong);

  fflush(stdin); *// sử dụng fflush(stdin) để xóa bộ đệm trước khi nhập chuỗi*

  printf("\nNhap ngay vao lam: ");

  nhapngaythang(nv.ngayvao);

#### 2.3.1. nhapdanhsach (danhsach &ds)

- Kiểm tra số lượng nhân viên :

+, Nếu số lượng nhân viên nhỏ hơn 0 thì bắt nhập lại

+, Nếu số lượng nhân viên lớn số lượng mặc định là 100 thì bắt nhập lại

* Đưa điều kiện nhập số nhân viên vào trong vòng lặp
* Nếu nhập sai bắt nhập lại cho đến khi nhập đúng yêu cầu
* Sau đó cho nhập từng nhân viên
* Dùng vòng lặp “for” và dùng hàm nhap1nv(ds, ds.a[i])

void nhapdanhsach(danhsach *&*ds)

{

*// nếu số lượng nhân viên không hợp lệ thì bắt nhập lại*

  do

  {

    printf("\nNhap so luong nhan vien: ");

    scanf("%d", &ds.n);

    if (ds.n < 0 || ds.n > sl)

    {

      printf("So luong nhan vien khong hop le, vui long nhap lai!\n");

    }

  } while (ds.n < 0 || ds.n > sl);

  for (int i = 0; i < ds.n; i++)

  {

    printf("\nNhap thong tin cho nhan vien thu %d:\n", i + 1);

    nhap1nv(ds, ds.a[i]);

  }

}

#### xuat1nv(Nhanvien nv)

Giai thích:

-%s: in ra một chuỗi

%20s : dùng để in ra chuỗi có 20 kí tự

void xuat1nv(nhanvien nv)

{

  printf("\n%5s %20s %20s %20s %20s %20s %15s %15s",

         "MSNV", "Ho", "Ten", "Ngay sinh", "Noi sinh", "Dia chi", "Luong", "Ngay vao");

  printf("\n%5d %20s %20s ", nv.msnv, nv.ho, nv.ten);

  xuatngaythang(nv.namsinh);

  printf("%20s %20s %15.2f ", nv.noisinh, nv.diachi, nv.luong);

  xuatngaythang(nv.ngayvao);

}

#### xuatdanhsach(danhsach ds)

Giai thích:

* Tương tự như 2.3.1.2 giải thích về %s
* Dùng vòng lặp “for” in ra từng thông tin của nhân viên

void xuatdanhsach(danhsach ds)

{

  printf("\nDanh sach nhan vien:");

  printf("\n%5s %20s %20s %20s %20s %20s %20s %20s",

         "MSNV", "Ho", "Ten", "Ngay sinh", "Noi sinh", "Dia chi", "Luong", "Ngay vao");

  for (int i = 0; i < ds.n; i++)

  {

    printf("\n%5d %20s %20s ", ds.a[i].msnv, ds.a[i].ho, ds.a[i].ten);

    xuatngaythang(ds.a[i].namsinh);

    printf("%20s %20s %15.2f ", ds.a[i].noisinh, ds.a[i].diachi, ds.a[i].luong);

    printf("\t");

    xuatngaythang(ds.a[i].ngayvao);

  }

}

2.3.2.4 timten(danhsach ds, char \*c)

Giai thích: khai báo int i

Và duyệt 1 vòng lặp for🡪 cho I bắt đầu =0-> i chạy từ 0🡪 hết danh sách🡪 và tăng lên 1

-srtcasecmp có phân biệt hoa chữ thường .

- Nếu tìm thấy trả về vị trí của nhân viên tìm thấy

- Không tìm thấy trả về -1

\* Giải thích vì sao trả về -1 vì danh sach(list) hay ở dây là mảng đều có vị trí (index) lớn hơn hoặc bằng 0 nên trả về -1 hay bất kì 1 số âm đầu đều được

int timten(danhsach ds, char *\**c)

{

  for (int i = 0; i < ds.n; i++)

  {

    if (strcasecmp(ds.a[i].ten, c) == 0) *// so sanh 2 chuoi khong phan biet hoa thuong (strcasecmp)*

    {

      return i;

    }

  }

  printf("Khong tim thay nhan vien co ten %s trong danh sach!\n", c);

  return -1;

}

#### timMSNV(danhsach ds, int d)

Giai thích: đầu vào danhSach, int mã nhân viên cầm tìm (d)

-hàm duyệt từng phần tử danh sách ( dùng vòng lặp for)

+khởi tạo biên i=0🡪 duyệt từng phần tử ds bắt đầu ở vị trí i

-khi tìm nv trùng khớp ma🡪 thi trả giá vế biên i

- duyệt qua toàn bộ mà k tìm thấy mã cần tìm🡪trả về gtri-1 🡪k tìm thấy

int timMSNV(danhsach ds, int d)

{

  for (int i = 0; i < ds.n; i++)

  {

    if (ds.a[i].msnv == d)

    {

      return i;

    }

  }

  return -1;

}

#### sapxep(danhsach &ds)

(sắp xếp nổi bọt): hàm được sử dụng sắp xếp các ptu mảng theo thứ tự tăng hoặc giảm dần🡪tìm ptu nhỏ nhất 🡪vị trí đầu tiên -> tiếp tục các ptu tiếp theo.

Giai thich: hàm này nhận 1 đối tượng danhsach🡪 là danh sách nhanVien của các đối tượng-> mỗi nv đối tương có 1 thuộc tính🡪 luong

-hàm thực hiện vòng lặp 🡪kép🡪 so sánh cặp nv trong ds

🡪 nếu nv hiện tại thấp hơn nv tiếp theo🡪 2 nv hoán đổi vị trí ds

* và cuối cùng hàm trả về ds sắp xếp
* void sapxep(danhsach *&*ds) *// & dung de thay doi gia tri cua ds trong ham sapxep ma khong can return ds sau khi sap xep*
* {
* int i, j;
* for (i = 0; i < ds.n - 1; i++)
* {
* for (j = 0; j < ds.n - i - 1; j++) *// -i-1 de bo qua phan tu cuoi cung da sap xep*
* {
* if (ds.a[j + 1].luong > ds.a[j].luong)
* {
* *// hoan doi vi tri hai nhan vien*
* nhanvien temp = ds.a[j];
* ds.a[j] = ds.a[j + 1]; *// gán giá trị của ds.a[j + 1] cho ds.a[j]*
* ds.a[j + 1] = temp; *// gán giá trị của biến tạm cho ds.a[j + 1]*
* }
* }
* }
* }

#### xoanv(danhsach &ds,int d)

-Nhận 1 một trỏ 🡪 cấu trúc danh sách và ma con trỏ là số nguyên

🡪 hàm tk theo mã🡪 tìm kiếm 1 ptu trong mảng🡪 gtri được lưu trữ trong biến pos

* Đầu tiên trả về vị trí của mã cần tìm
* Nếu không tìm thấy thông báo cho người dùng
* Nếu tìm thấy thì cho vào vòng lặp for

void xoanv(danhsach *&*ds, int d)

{

  int i = timMSNV(ds, d);

  if (i == -1)

  {

    printf("Khong tim thay nhan vien co ma so %d de xoa.\n", d);

    return;

  }

  for (int j = i; j < ds.n - 1; j++)

  {

    ds.a[j] = ds.a[j + 1];

  }

  ds.n--;

  printf("Da xoa nhan vien co ma so %d.\n", d);

}