[PHẦN MỞ ĐẦU 4](#_Toc27089729)

[1. GIỚI THIỆU ĐỀ TÀI 4](#_Toc27089730)

[2. PHẠM VI NGHIÊN CỨU 4](#_Toc27089731)

[3. CÔNG CỤ THỰC HIỆN 4](#_Toc27089732)

[4. PHƯƠNG PHÁP NGHIÊN CỨU 4](#_Toc27089733)

[5. MỤC TIÊU NGHIÊN CỨU 4](#_Toc27089734)

[CHƯƠNG 1: CƠ SỞ LÝ THUYẾT 5](#_Toc27089735)

[1. Khái niệm danh sách liên kết đơn. 5](#_Toc27089736)

[2. Thao tác trên 1 phần tử. 6](#_Toc27089737)

[3. Các thao tác trên danh sách liên kết đơn. 6](#_Toc27089738)

[3.1. Tạo danh sách liên kết rỗng. 7](#_Toc27089739)

[3.2. Thêm 1 phần tử vào đầu danh sách. 7](#_Toc27089740)

[3.3. Thêm 1 phần tử vào cuối danh sách. 7](#_Toc27089741)

[3.4. Thêm 1 phần tử vào sau phần tử q. 8](#_Toc27089742)

[3.5. Xóa 1 phần tử đầu danh sách. 8](#_Toc27089743)

[3.6. Xóa 1 phần tử bất kỳ trong danh sách. 9](#_Toc27089744)

[Chương 2 : Ứng dụng danh sách liên kết đơn vào quản lý nhân khẩu 10](#_Toc27089745)

[1. Đặc tả bài toán. 10](#_Toc27089746)

[2. Phân tích thiết kế chương trình. 10](#_Toc27089747)

[2.1. Thiết kế cấu trúc dữ liệu 10](#_Toc27089748)

[2.2. Sơ đồ chức năng của chương trình : 12](#_Toc27089749)

[2.3. Thiết kế giao diện 13](#_Toc27089750)

[2.4. Xây dựng chương trình 14](#_Toc27089751)

[3. Cài đặt và kiểm thử. 20](#_Toc27089752)

[Chương 3: Mô tả phân công công việc 21](#_Toc27089753)

[Chương 4 : Kết luận 22](#_Toc27089754)

[1. Những kết quả đạt được. 22](#_Toc27089755)

[2. Những hạn chế 22](#_Toc27089756)

[3. Khó khăn khi thực hiện. 22](#_Toc27089757)

[4. Hướng phát triển của đề tài. 23](#_Toc27089758)

[5. Ưu khuyết điểm khi thực hiện 23](#_Toc27089759)

[6. Đánh giá mức độ thực hiện 23](#_Toc27089760)

[TÀI LIỆU THAM KHẢO 24](#_Toc27089761)

**Danh mục hình ảnh – sơ đồ**

[Hình 1. 1 Mô tả cấu trúc 1 phần tử trong danh sách liên kết. 5](#_Toc27089762)

[Hình 1. 2 Mô tả mối liên kết các phần tử trong danh sách liên kết. 5](#_Toc27089763)

[Hình 1. 3 Thêm 1 phần tử vào đầu danh sách. 7](#_Toc27089764)

[Hình 1. 4 Thêm 1 phần tử vào cuối danh sách. 8](#_Toc27089765)

[Hình 1. 5 Thêm 1 phần tử sau phần tử q. 8](#_Toc27089766)

[Hình 1. 6 Xóa 1 phần tử đầu danh sách. 9](#_Toc27089767)

[Hình 1. 7 Xóa 1 phần tử bất kỳ trong danh sách. 9](#_Toc27089768)

[Hình 2. 1 Màn hình tra cứu hộ khẩu 12](file:///C:\Users\tantu\Downloads\Báo-cáo-CTDL-đã-xóa-1%20(1).docx#_Toc27089769)

[Hình 2. 2 Màn hình Đăng nhập 12](file:///C:\Users\tantu\Downloads\Báo-cáo-CTDL-đã-xóa-1%20(1).docx#_Toc27089770)

[Hình 2. 3 Màn hình thủ tục Diễn biến 12](file:///C:\Users\tantu\Downloads\Báo-cáo-CTDL-đã-xóa-1%20(1).docx#_Toc27089771)

[Hình 2. 4 Màn hình Đăng ký hộ khẩu 13](file:///C:\Users\tantu\Downloads\Báo-cáo-CTDL-đã-xóa-1%20(1).docx#_Toc27089772)

[Sơ đồ 2. 1 Sơ đồ chức năng của chương trình 12](file:///C:\Users\tantu\Downloads\Báo-cáo-CTDL-đã-xóa-1.docx#_Toc27042865)

[Sơ đồ 2. 2 Sơ đồ thuật toán chức năng đăng ký nhân khẩu 15](file:///C:\Users\tantu\Downloads\Báo-cáo-CTDL-đã-xóa-1.docx#_Toc27042866)

[Sơ đồ 2. 3 Sơ đồ thuật toán Tra cứu nhân khẩu 19](file:///C:\Users\tantu\Downloads\Báo-cáo-CTDL-đã-xóa-1.docx#_Toc27042867)

[Bảng 2. 1 Bảng cài đặt và kiểm thử 21](#_Toc27042878)

[Bảng 3. 1 Bảng mô tả phân công công việc 22](#_Toc27042888)

# PHẦN MỞ ĐẦU

## GIỚI THIỆU ĐỀ TÀI

**Đề tài :** Xây dựng chương trình quản lý nhân khẩu sử dụng danh sách liên kết đơn. Công nghệ thông tin là một trong những ngành phát triển vượt bậc trong những năm gần đây. Ngày nay, cùng với sự phát triển nhanh chóng của xã hội thì công nghệ thông tin được ứng dụng rộng rãi ở hầu hết các lĩnh vực và ngày càng đóng vai trò quan trọng, trở thành một phần thiết yếu của đời sống hàng ngày. Và một lĩnh vực rất quan trọng cần ứng dụng công nghệ thông tin là quản lý nhân khẩu.

## PHẠM VI NGHIÊN CỨU

* Lý thuyết về danh sách liên kết đơn (cài đặt bằng con trỏ).
* Bài toán quản lý nhân khẩu.

## CÔNG CỤ THỰC HIỆN

* Phần mềm lập trình Visual Studio 2015.
* Ngôn ngữ lập trình C#.
* Công cụ WinForm.

## PHƯƠNG PHÁP NGHIÊN CỨU

* Nghiên cứu tài liệu Cấu trúc dữ liệu và giải thuật trên sách, giáo trình, bài giảng.
* Tìm kiếm và nghiên cứu trên Internet.

## MỤC TIÊU NGHIÊN CỨU

* Xây dựng được chương trình quản lý nhân khẩu ứng dụng được danh sách liên kết.
* Giao diện dễ sử dụng.
* Thao tác nhanh gọn.
* Đa dạng các chức năng.
* Thống kê các số liệu 1 cách trực quan.
* Lưu trữ thông tin dễ dàng.

# CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

## Khái niệm danh sách liên kết đơn.

Danh sách liên kết đơn là một tập hợp các Node được phân bố động, được sắp xếp theo cách sao cho mỗi Node chứa “một giá trị”(Data) và “một con trỏ”(Next). Con trỏ sẽ trỏ đến phần tử kế tiếp của danh sách liên kết đó. Nếu con trỏ mà trỏ tới NULL, nghĩa là đó là phần tử cuối cùng của linked list. [1]



Hình 1. 1 Mô tả cấu trúc 1 phần tử trong danh sách liên kết.



Hình 1. 2 Mô tả mối liên kết các phần tử trong danh sách liên kết.

**Cú pháp:**  public class Linkedlist

{

public Kiểu\_dữ\_liệu data;

pucblic Linkedlist next=null;

public Linkedlist pHead;

public Linkedlist pTail;

}

## Thao tác trên 1 phần tử.

- Khai báo biến con trỏ p là biến con trỏ kiểu Node:

* LinkedList **p**
* Cấp phát bộ nhớ cho node của LinkedList mới – được trỏ bởi **p** :
  + LinkedList **p** = new LinkedList();
* Truy cập trường data :
  + p.data
* Truy cập trường next :
  + p.next

## Các thao tác trên danh sách liên kết đơn.

nơ

* Danh sách luôn có con trỏ đầu danh sách: con trỏ L.
* Danh sách luôn có giá trị báo kết thúc danh sách: NULL.
* Trường next của mỗi Node chỉ chứa địa chỉ của Node sau nó, trừ Node cuối.
* Trường next của Node cuối cùng chứa giá trị NULL.
* Không tách danh sách thành 2 danh sách con nếu danh sách phần sau

chưa có con trỏ tới.

* Khởi tạo danh sách rỗng: L = NULL.
* Các thao tác cơ bản trong danh sách liên kết đơn.

### Tạo danh sách liên kết rỗng.

Một danh sách rỗng khi không có phần tử nào :

pHead = pTail = null;

### Thêm 1 phần tử vào đầu danh sách.

Nếu danh sách rỗng thì

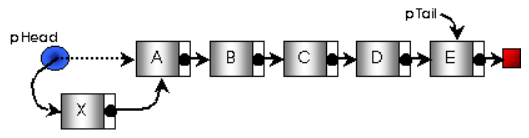
                  B11 : Head = new Linkedlist;

                  B12 : Tail = Head;

      Ngược lại

                  B21 : new Linkedlist ->pNext  = Head;

                  B22 : Head = new Linkedlist;  [2]­­



Hình 1. 3 Thêm 1 phần tử vào đầu danh sách.

### Thêm 1 phần tử vào cuối danh sách.

Nếu Danh sách  rỗng Thì

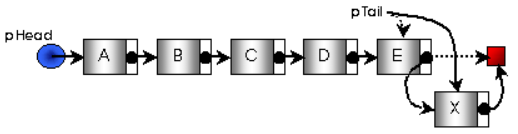
            B11 : Head = new Linkedlist;

            B12 : Tail = Head;

Ngược lại

            B21 : Tail ->pNext  = new Linkedlist;

            B22 : Tail  = new Linkedlist; [2]



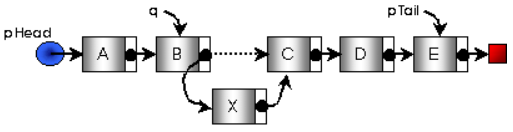
Hình 1. 4 Thêm 1 phần tử vào cuối danh sách.

### Thêm 1 phần tử vào sau phần tử q.

Nếu  ( q != NULL) thì

                  B1 :      new Linkedlist -> pNext  = q->pNext;

                  B2 :      q->pNext = new Linkedlist; [2]



Hình 1. 5 Thêm 1 phần tử sau phần tử q.

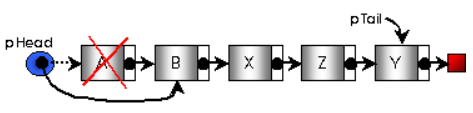
### Xóa 1 phần tử đầu danh sách.

Nếu  (Head != NULL)  thì

B1:       p = Head;                                 // p là phần tử cần hủy

B2:      Head = Head->pNext;    // tách p ra khỏi xâu

B3:       Nếu Head=NULL thì Tail = NULL; //Xâu rỗng [2]



Hình 1. 6 Xóa 1 phần tử đầu danh sách.

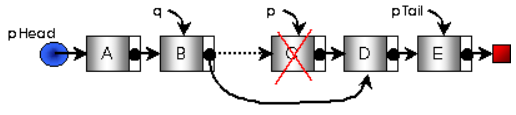
### Xóa 1 phần tử bất kỳ trong danh sách.

Nếu  (q!= NULL)  thì

B1:       p = q->Next;                             // p là phần tử cần hủy

B2:       Nếu  (p != NULL)  thì  // q không phải là cuối xâu

             B21 :  q->Next = p->Next;        // tách p ra khỏi xâu [2]



Hình 1. 7 Xóa 1 phần tử bất kỳ trong danh sách.

# Chương 2 : Ứng dụng danh sách liên kết đơn vào quản lý nhân khẩu

## Đặc tả bài toán.

**Quản lý nhân khẩu** là công việc hàng ngày của chuyên viên phụ trách quản lý nhân khẩu các cấp xã, phường,... Công việc quản lý nhân khẩu đòi hỏi sự tỉ mỉ, cẩn thận trong khâu ghi chép các thông tin cá nhân của từng nhân khẩu : Họ và tên, Giới tính, CMND,... cũng như việc thống kê các trường hợp tạm trú tạm vắng, báo tử,.... một cách rõ ràng và chính xác.

Yêu cầu bài toán là xây dựng 1 chương trình quản lý nhân khẩu một cách dễ dàng tiện lợi trên sự trợ giúp của máy tính. Mọi thao tác đều được thực hiện chung 1 cửa sổ làm việc để đảm bảo tính đồng bộ và bảo mật.[3]

Do yêu cầu tính nhanh gọn dễ sử dụng trong thao tác quản lý nhân khẩu nên thuật toán nhóm quyết định chọn **danh sách liên kết đơn** cài đặt bằng con trỏ. Với tính linh hoạt trong việc cấp phát bộ nhớ cũng như thứ tự sắp xếp các phần tử 1 cách linh động, thì thuật toán này phụ hợp với yêu cầu bài toán.

## Phân tích thiết kế chương trình.

### Thiết kế cấu trúc dữ liệu

#### Phát biểu bài toán :

Xây dựng chương trình quản lý nhân khẩu sử dụng danh sách liên kết đơn.

#### Phân tích bài **toán** :

Mỗi công dân khi ra đời sẽ được cha mẹ làm giấy khai sinh 🡪 Đăng ký nhân khẩu và hộ khẩu

Công dân có quyền chuyển nơi ở, hộ khẩu 🡪 Đăng ký tạm trú – tạm vắng, nhập – tách khẩu.

Quá trình công dân ở đâu sẽ được chính quyền nơi đó theo dõi 🡪 Hồ sơ Diễn biến.

Khi một người nào chết 🡪 Hồ sơ báo tử.

Vì số lượng nhân khẩu và hộ khẩu khá lớn 🡪 Tra cứu nhân, hộ khẩu.

Dữ liệu vào và ra tương đối lớn 🡪 Chọn lưu trữ thông tin vào file text.

#### Tổ chức dữ liệu :

##### Dữ liệu đầu vào :

Trong phân hệ quản lý nhân khẩu, thì đầu vào là :

* Hồ sơ 1 nhân khẩu: số định danh, họ và tên, này sinh, dân tộc, giới tính, quốc tịch, chứng minh nhân dân, ngày cấp, nơi cấp, trình độc học vấn, nghề nghiệp, nơi làm việc, quan hệ với chủ hộ, nơi sinh, nguyên quán, chỗ ở thường trú, nơi ở hiện tại, ghi chú.
* Hồ sơ 1 hộ khẩu : Số sổ hộ khẩu, họ và tên chủ hộ, giới tính, ngày sinh,

dân tộc, quốc tịch, ngày đăng ký, chỗ ở hiện tại.

* Hồ sơ Diễn biến của 1 nhân khẩu : Họ và tên, nơi đi, ngày đi, nơi đến, ngày đến, lí do, ngày lập.
* Hồ sơ Báo tử : Họ và tên người mất, ngày mất, lí do mất, người khai, ngày khai, ghi chú.
* Hồ sơ Tạm trú – tạm vắng : Họ và tên, ngày vắng, vắng tới ngày,lí do vắng, người khai báo,ngày lập, ghi chú.
* Hồ sơ Nhập – tách khẩu : Họ và tên, số sổ hộ khẩu hiện tại, số sổ hộ khẩu đã nhập,quan hệ với chủ hộ, ngày nhập.

Dữ liệu được lưu dưới dạng file text để dễ dàng trong việc lưu giữ và truy xuất nhiều lần.

1. Dữ liệu đầu ra :

* Đầu ra được thực hiện dưới dạng các báo biểu.
* Danh sách những hộ đã đăng ký.
* Danh sách nhân khẩu đã đăng ký.
* Danh sách diễn biến, tạm trú – tạm vắng, nhập – tách khẩu và báo tử.
* Dữ liệu sau khi được thao tác sẽ được cập nhật lưu lại tại file text trước.

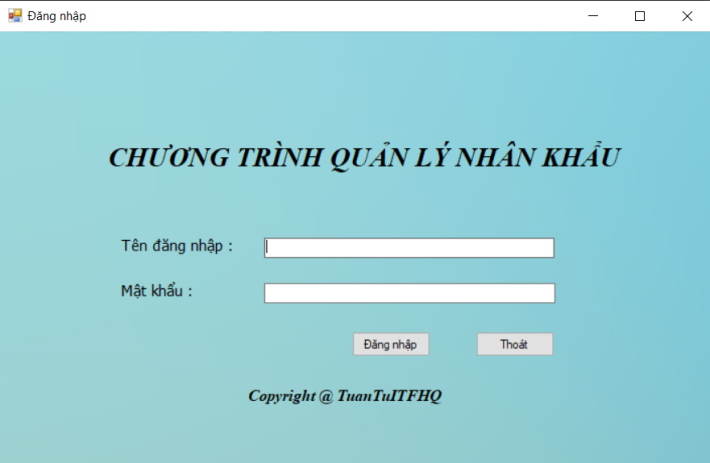
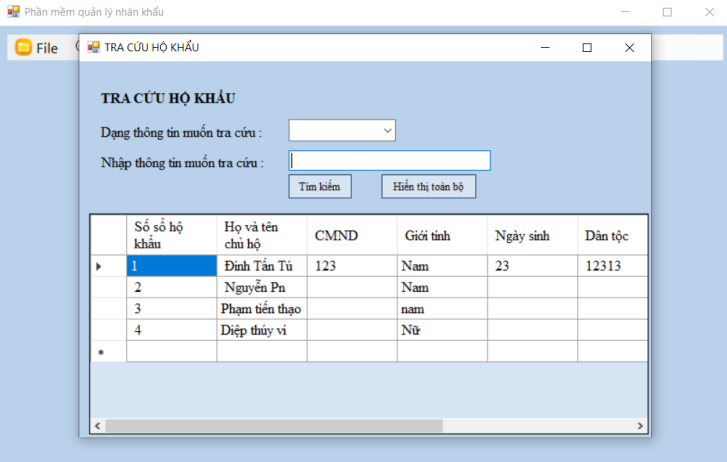
#### Xác định thuật toán

Sử dụng DSLKĐ để lưu thông tin của từng hồ sơ nhân khẩu, hộ khẩu, hồ sơ diễn biến, tạm trú - tạm vắng, nhập – tách khẩu, báo tử của từng nhân khẩu. Lúc này mỗi hồ sơ sẽ là 1 phần tử trong danh sách liên kết đơn và danh sách những hồ sơ đó cũng là một danh sách liên kết đơn. Ta sẽ ứng dụng những thao tác thêm, sửa, xóa, tìm kiếm phần tử ở trên để thực hiện với những hồ sơ đó.

### Sơ đồ chức năng của chương trình :

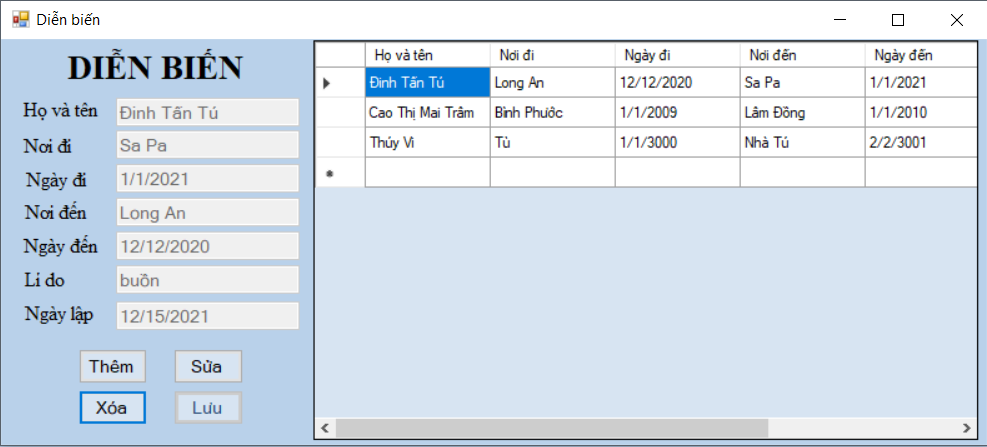
Sơ đồ 2. 1 Sơ đồ chức năng của chương trình

### Thiết kế giao diện

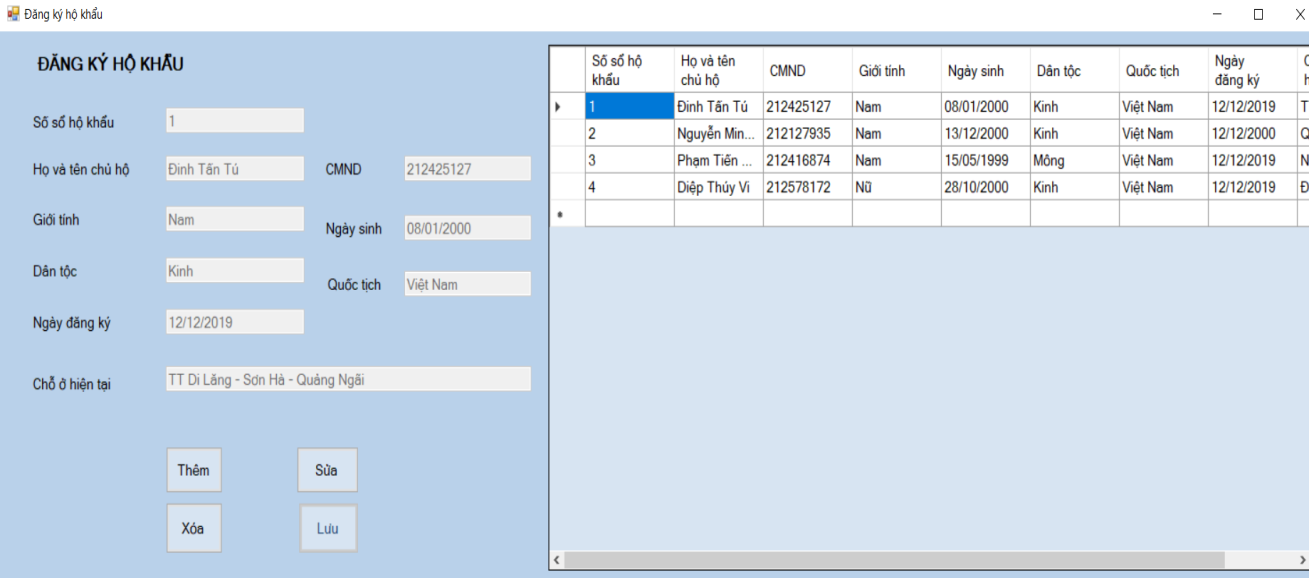


Hình 2. 1 Màn hình tra cứu hộ khẩu

Hình 2. 2 Màn hình Đăng nhập



Hình 2. 3 Màn hình thủ tục Diễn biến



Hình 2. 4 Màn hình Đăng ký hộ khẩu

### Xây dựng chương trình

* Vì các chức năng thêm, xóa, sửa, lưu cũng như các hàm đọc, ghi dữ liệu từ file text, ... của các lớp Báo tử, Nhập – tách khẩu, Nhân khẩu, ... đều chung một thuật toán và được xây dựng bên trong lớp chứa danh sách liên kết đơn nhằm thuận tiện cho việc truy xuất sau này nên dưới đây chỉ minh họa cho lớp Nhân khẩu.
* Xây dựng lớp chứa danh sách liên kết đơn cài đặt bằng con trỏ chứa thông tin hồ sơ Nhân khẩu.

public class Nhankhau1

{

public string sSosohk, sHovaten, sNgaysinh, sGioitinh, sDantoc, sQuoctich, sCmnd, sNgaycap, sNoicap, sTDhocvan, sNoilamviec;

public string sNghenghiep, sQuanhe, sTienan;

public string sNoisinh;

public string sNguyenquan;

public string sNoiott;

public string sChoohientai;

public string sGhichu;

public Nhankhau1 next = null;

}

public class ListNhankhau1

{

public Nhankhau1 pHead;

public Nhankhau1 pTail;

}

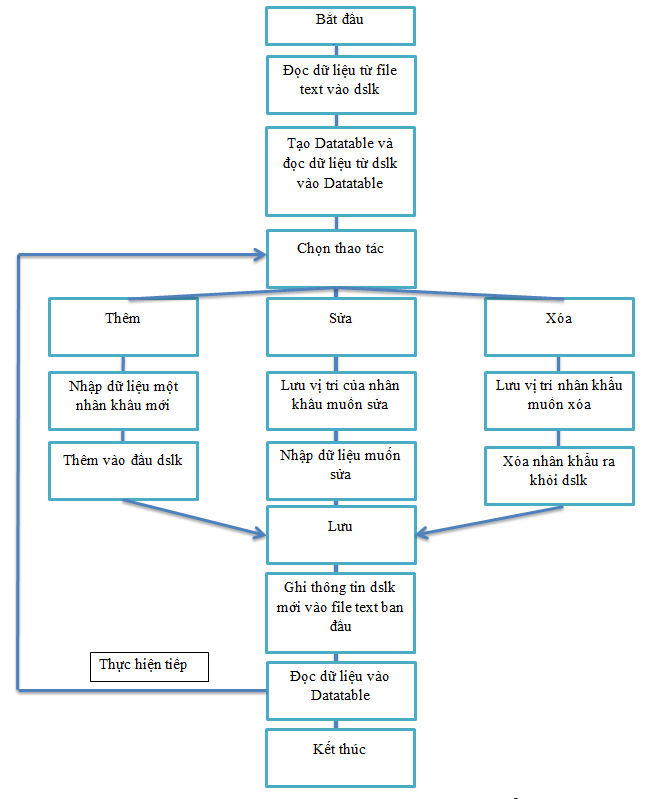
* Hàm khởi tạo danh sách liên kết đơn chứa thông tin nhân khẩu.

public ListNhankhau1()

{

pHead = pTail = null;

}

* Đăng ký nhân khẩu :

Sơ đồ 2. 2 Sơ đồ thuật toán chức năng đăng ký nhân khẩu.

**Các hàm được sử dụng trong thuật toán trên.**

* Hàm thêm 1 nhân khẩu mới vào đầu danh sách.

public void Addhead(Nhankhau1 p)

{

if (pHead == null) //list rỗng

{

pHead = p;

pTail = p;

}

else

{

p.next = pHead;

pHead = p;

}

}

* Hàm thêm 1 nhân khẩu mới vào cuối danh sách.

public void Addtail(Nhankhau1 p)

{

if (pHead == null) //list rỗng

{

pHead = p;

pTail = p;

}

else

{

pTail.next = p;

pTail = p;

}

}

* Hàm xóa 1 nhân khẩu bất kỳ trong danh sách.

public void RemoveNode(Nhankhau1 p, int x)

{

Nhankhau1 g = new Nhankhau1();

Nhankhau1 k = pHead;

int i = 0;

if (x == 0) pHead = pHead.next;

else

{

while (i < x)

{

g = k;

k = k.next;

i++;

}

g.next = k.next;

}

}

* Hàm event khi bấm Xóa.

private void buttonXoa\_Click(object sender, EventArgs e)

{

flag = "delete";

xoa = dsnk.pHead;

if (MessageBox.Show("Bạn có thật sự muốn xóa hộ khẩu này này?", "Thông Báo", MessageBoxButtons.YesNo) == System.Windows.Forms.DialogResult.Yes)

{

int temp = dataGridView1.CurrentCell.RowIndex;

dsnk.RemoveNode(xoa, temp);

DataGridViewRow index = dataGridView1.CurrentRow;

dataGridView1.Rows.Remove(index);

}

Block();

buttonLuu.Enabled = true;

dataGridView1.RefreshEdit();

}

* Hàm event khi bấm Sửa.

private void buttonSua\_Click(object sender, EventArgs e)

{

flag = "edit";

sua = dsnk.pHead;

while (sua.sHovaten != textBoxHovaten.Text)

{

sua = sua.next;

}

Unlock();

buttonThem.Enabled = buttonSua.Enabled = buttonXoa.Enabled = false;

}

* Hàm event khi bấm Lưu.

private void buttonLuu\_Click(object sender, EventArgs e)

{

buttonThem.Enabled = buttonSua.Enabled = buttonXoa.Enabled = true;

if (flag == "add")

{

Nhankhau1 p = new Nhankhau1();

p.sSosohk = comboBox1.Text;

....

p.sGhichu = textBoxGhichu.Text;

dsnk.Addtail(p);

dtbt.Rows.Add(p.sSosohk,....,p.sGhichu);

dataGridView1.DataSource = dtbt;

dataGridView1.RefreshEdit();

}

if (flag == "edit")

{

Nhankhau1 dssua = dsnk.pHead;

dsnk.pHead = sua;

dsnk.pHead.sSosohk = comboBox1.Text;

...

sua.sGhichu = textBoxGhichu.Text;

dsnk.pHead = dssua;

int index = dataGridView1.CurrentCell.RowIndex;

DataTable dt = (DataTable)dataGridView1.DataSource;

if (dt.Rows.Count > 0)

{

dataGridView1.Rows[index].Cells[0].Value = comboBox1.Text;

...

}

dataGridView1.RefreshEdit();

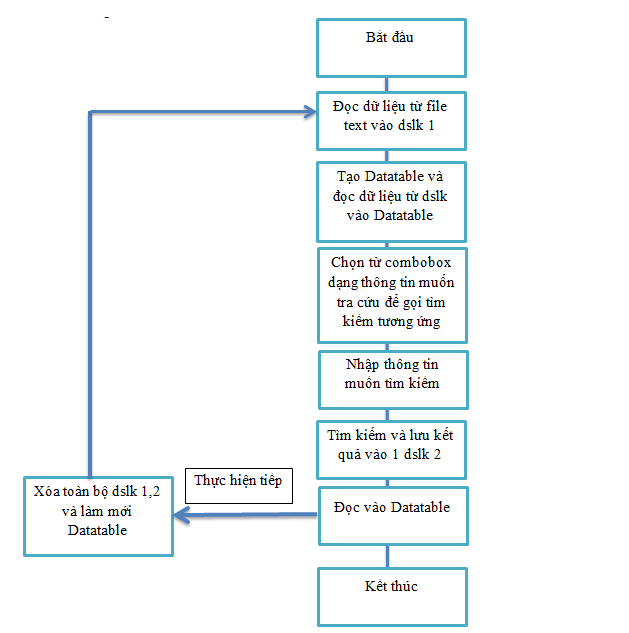
}

Unlock();

GhiNhankhau();

Block();

}

* Tra cứu nhân khẩu :

Sơ đồ 2. 3 Sơ đồ thuật toán Tra cứu nhân khẩu.

* Hàm tìm kiếm 1 nhân khẩu theo 1 thông tin (minh họa tìm kiếm theo tên).

public ListNhankhau1 Search(string x)

{

ListNhankhau1 dstam = new ListNhankhau1();

Nhankhau1 tem = new Nhankhau1();

int dem = 0;

for (Nhankhau1 p = pHead; p != null; p = p.next)

{

if (p.sSosohk == x)

{

tem = p;

if (dem == 0)

{

dstam.Addhead(tem);

}

else

{

dstam.Addtail(tem);

}

dem++;

}

}

tem.next = null;

return dstam;

}

* Hàm event khi bấm Tìm kiếm.

private void button1\_Click(object sender, EventArgs e)

{

switch (comboBox1.Text)

{

case "Số sổ hộ khẩu":

{

dstam = dsnk.SearchSosohk(textBox1.Text);

break;

}

.......

}

Khoitaodstam();

BlockTimkiem();

}

* Hàm event khi bấm Hiển thị toàn bộ.

private void button2\_Click(object sender, EventArgs e)

{

dtbttam.Clear();

dsnk.DeleteList();

dstam.DeleteList();

DocNhanKhau();

dataGridView1.DataSource = dtbt;

dataGridView1.Refresh();

UnlockTimkiem();

textBox1.Clear();

}

Bảng 2. 1 Bảng cài đặt và kiểm thử

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Chức năng gặp lỗi** | **Mô tả lỗi** | **Nguyên nhân** | **Người phát hiện** | **Cách khắc phục** | **Tình trạng khắc phục** |
| Tra cứu | Nếu có nhiều nhân khẩu có cùng tên thì chỉ xuất ra được nhân khẩu đầu tiên. | Hàm tìm kiếm cũ : sau mỗi lần tìm kiếm sẽ đổi Next của phần tử đó thành null 🡪 các phần tử được tìm thấy không liên kết với nhau. | Nguyễn Minh Tuấn | Chỉ đổi Next của phần tử cuối bằng null. | Đãkhắc phục. |
|  | Kết quả tìm kiếm lần 2 trở đi sẽ ghi vào sau danh sách tìm kiếm lần 1. | Sau khi tìm kiếm lần 1 chưa có lệnh xóa toàn bộ phần tử trong danh sách tìm kiếm và lệnh Refresh Datatable. | Đinh Tấn Tú | Thêm lệnh xóa toàn bộ danh sách và refresh Datatable. | Đã khắc phục. |
| Đăng ký | Nếu trong 1 danh sách có phần tử đầu hoặc cuối giống với phần tử bất kỳ muốn xóa trong danh sách, thì hàm xóa chỉ xóa phần từ đầu. | Chia ra 3 trường hợp : Xóa đầu, xóa cuối, xóa giữa. Chỉ thực hiện 1 trong 3 cách xóa mà không thực hiện đồng thời cả 3. | Đinh Tấn Tú | Viết lại hàm xóa mới dùng chỉ số của Curentcell.row để đặt số vòng lặp while để xác định vị trí phần tử cần xóa. | Đã khắc phục |

## Cài đặt và kiểm thử.

Bảng 3. 1 Bảng mô tả phân công công việc

|  |  |  |  |
| --- | --- | --- | --- |
| **KẾ HOẠCH THỰC HIỆN ĐỒ ÁN CUỐI KỲ MÔN CẤU TRÚC DỮ LIỆU & GIẢI THUẬT - 2019** | | | |
|  | **ĐỀ TÀI: TẠO CHƯƠNG TRÌNH QUẢN LÝ NHÂN KHẨU** | | |
| **DANH SÁCH CÔNG VIỆC** | **NỘI DUNG CHI TIẾT CÔNG VIỆC** | **Đinh Tấn Tú** | **Nguyễn Minh Tuấn** |
| Tìm hiểu |  |  |  |
|  | Tìm hiểu về cách thức hoạt động và sử dụng của danh sách liên kết | X | X |
|  | Tìm hiểu về đồ họa trong C# và đề ra mục tiêu về giao diện chương trình. | X | X |
|  | Tìm hiểu về những yêu cầu của người dùng đối với chương trình cần tạo ra. | X | X |
| Phân tích |  |  |  |
|  | Phân tích thiết kế thuật toán sẽ sử dụng | X | X |
|  | Thiết kế giao diện phù hợp với yêu cầu và mục tiêu đề ra. | X | X |
| Thực hiện |  |  |  |
|  | Viết code chức năng Đăng ký hộ khẩu, nhân khẩu | X |  |
|  | Viết code Thủ tục Diễn biến, Báo tử,Nhập tách khẩu, Tạm trú tạm vắng | X |  |
|  | Viết code Tra cứu Hộ khẩu, Nhân khẩu |  | X |
|  | Kiểm duyệt, rút gọn code | X | X |
| Khắc phục | Nhìn nhận những ưu nhược điểm của chương trình đã tạo ra và đề ra hướng khắc phục cho những đồ án sau này. | X | X |

# Chương 3: Mô tả phân công công việc

# Chương 4 : Kết luận

## Những kết quả đạt được.

Qua quá trình xây dựng chương trình, em đã thực hiện được những yêu cầu đề ra của đề tài là :

* Xây dựng được chương trình quản lý nhân khẩu với thuật toán đơn giản.
* Giao diện dễ sử dụng.
* Thao tác nhanh gọn.
* Dễ lưu trữ dữ liệu.

## Những hạn chế

Bên cạnh những vấn đề đã làm được, nhóm tác giả nhận thấy vẫn còn nhiều vấn đề còn hạn chế :

* Giao diện chưa đẹp.
* Chương trình còn khá ít tính năng sử dụng.
* Chưa thống kê dữ liệu.
* Chức năng tra cứu còn thô sơ, một vài trường hợp không thể tra cứu được.
* Tính bảo mật thông tin chưa cao.

## Khó khăn khi thực hiện.

* Tiếp cận với ngôn ngữ mới và kiến thức về môn học mới.
* Làm quen với sử dụng công cụ mới : Winform.
* Khả năng debug để phát hiện lỗi chưa tốt.

## Hướng phát triển của đề tài.

* Phát triển đề tài từ quy mô quản lý nhân khẩu cấp nhỏ đến quản lý nhân khẩu các cấp lớn hơn.
* Phát triển nhiều tính năng để quản lý.
* Xây dựng lại giao diện đẹp mắt và dễ sử dụng hơn.
* Khắc phục tính bảo mật bằng cách sử dụng Mysql để lưu trữ thông tin.

## Ưu khuyết điểm khi thực hiện

* Ưu điểm :
  + Khả năng làm việc nhóm và phân chia công việc hiệu quả.
  + Có sáng tạo khi thực hiện không hoàn toàn theo khuôn mẫu.
* Khuyết điểm :
  + Một số phần còn phải dựa vào tài liệu tham khảo.
  + Chưa thể 100% tự fix các lỗi xảy ra khi chạy thử.
  + Làm việc còn chậm so với tiến độ đề ra

## Đánh giá mức độ thực hiện

Thực hiện được khoảng 70% so với mục tiêu đề ra.

# TÀI LIỆU THAM KHẢO

**[1]** [**https://nguyenvanhieu.vn/danh-sach-lien-ket-don/**](https://nguyenvanhieu.vn/danh-sach-lien-ket-don/)

**[2]** [**https://voer.edu.vn/c/danh-sach-lien-ket-don/7c75a38d/62e2fd7a**](https://voer.edu.vn/c/danh-sach-lien-ket-don/7c75a38d/62e2fd7a)

**[3]** [**http://doc.edu.vn/tai-lieu/de-tai-phan-mem-quan-li-nhan-khau-54272/**](http://doc.edu.vn/tai-lieu/de-tai-phan-mem-quan-li-nhan-khau-54272/)

[**https://tailieu.vn/doc/do-an-tot-nghiep-phan-tich-thiet-ke-he-thong-quan-ly-nhan-khau-cho-mot-xa-1223743.html**](https://tailieu.vn/doc/do-an-tot-nghiep-phan-tich-thiet-ke-he-thong-quan-ly-nhan-khau-cho-mot-xa-1223743.html)

[**https://www.youtube.com/watch?v=h7OYMHCa2WE&list=PL33lvabfss1xnPhBJHjM0A8TEBBcGCTsf&index=3**](https://www.youtube.com/watch?v=h7OYMHCa2WE&list=PL33lvabfss1xnPhBJHjM0A8TEBBcGCTsf&index=3)