**Lab 5: Ứng dụng Quản lý Danh bạ**

**Yêu cầu:**

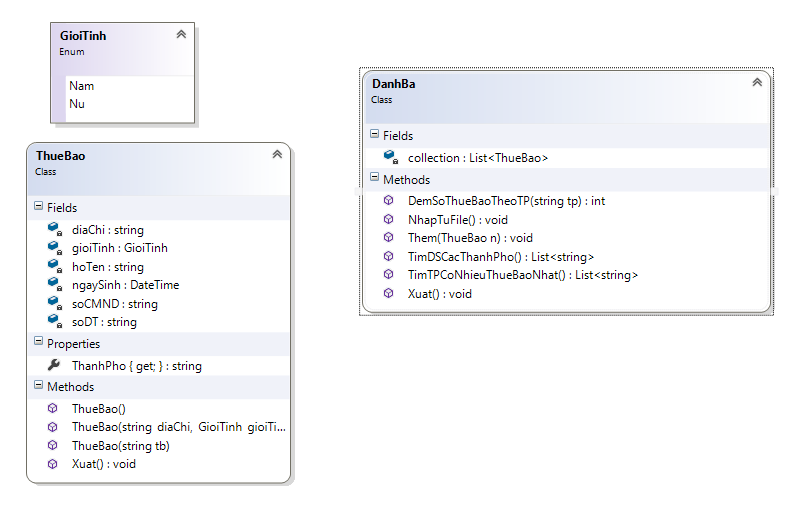
1. Sinh viên thực hiện bài thực hành và trả lời trực tiếp vào bài Lab
2. Nộp bài Lab vào cuối buổi thực hành.

**Mục tiêu**

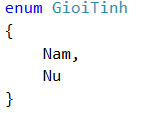
* Ôn tập tính đóng gói
* Sử dụng thuộc tính (Property)
* Sử dụng kiểu enum

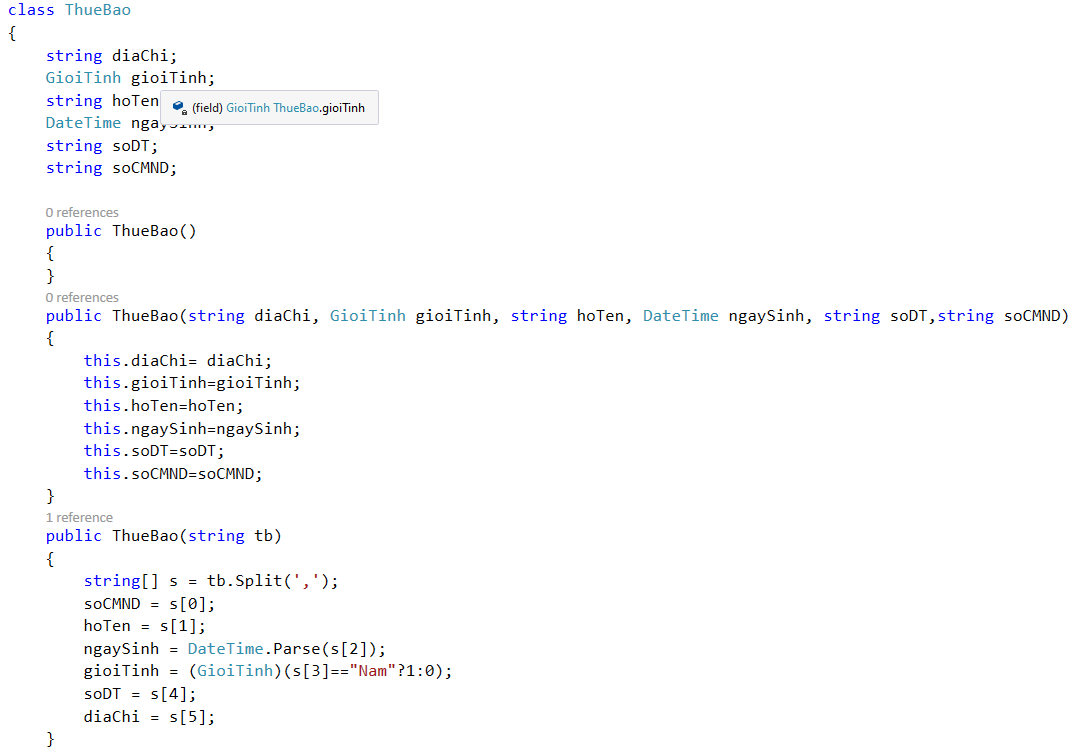
**Phần 1**

Lược đồ lớp của ứng dụng như sau:

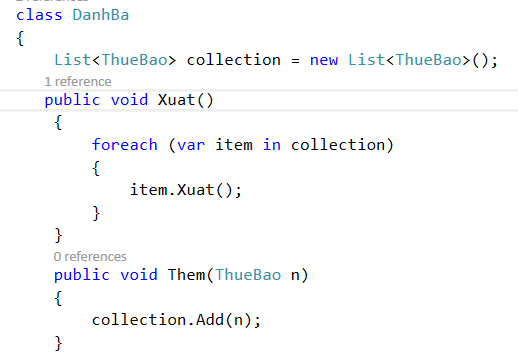


Tạo lớp ThueBao như sau:





Tạo lớp DanhBa như sau:



Tạo một tập tin data.csv và lưu vào ổ đĩa hay thư mục Debug của dự án với cấu trúc và dữ liệu như sau:

12345,Nguyen Van A,11/2/1990,Nam,123456,01 PDTV -Da Lat

12346,Tran Van B,11/3/1990,Nu,123457,2 PDTV -Da Nang

12347,Nguyen Van A,11/4/1990,Nam,123458,3 PDTV -Da Lat

12348,Tran Van B,11/5/1990,Nam,123459,4 PDTV -Nha Trang

12349,Nguyen Van A,11/6/1990,Nam,123460,2 PDTV -Da Lat

12350,Tran Van B,11/7/1990,Nam,123461,3 PDTV -Da Nang

12351,Nguyen Van A,11/8/1990,Nam,123462,4 PDTV -Quy Nhon

12352,Tran Van B,11/9/1990,Nam,123463,5 PDTV -Nha Trang

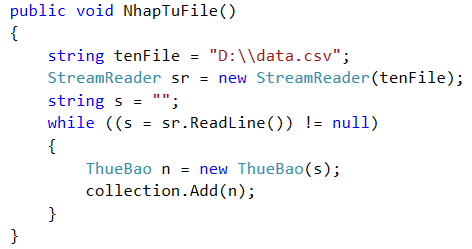
12353,Nguyen Van A,11/10/1990,Nam,123464,3 PDTV -Da Lat

12354,Tran Van B,11/11/1990,Nam,123465,4 PDTV -Da Nang

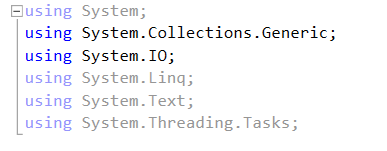
12355,Nguyen Van A,11/12/1990,Nam,123466,5 PDTV -Nha Trang

12356,Tran Van B,11/13/1990,Nam,123467,6 PDTV -Nha Trang

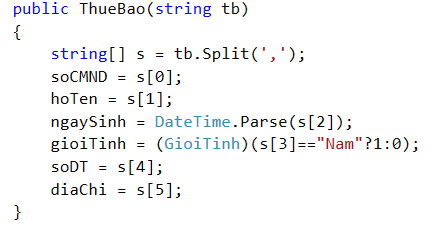
Trong lớp DanhBa tạo phương thức NhapTuFile như sau:



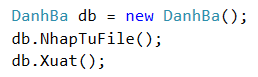
Khai báo thư viện System.IO



Khai báo phương thức tạo lập để tạo một đối tượng từ một dòng trong tập tin Data.csv



Trong hàm Main thực hiện đoạn mã sau để xem kết quả đã đọc được dữ liệu hay chưa:



Giải thích các dòng lệnh từ phần tạo phương thức NhapTuFile() ở trên và kết quả thực hiện chương trình:

/// <summary>

/// Lấy dữ liệu từ file csv và đưa các thuê bao vào mảng thuê bao

/// </summary>

public void NhapTuFile()

{

string root = "data.csv"; //Nhap ten file

StreamReader sr = new StreamReader(root); //Mo tap tin len va đọc nội dung

string s = ""; //Khai báo chuỗi rỗng dùng để nhập từng dòng trong file

while((s = sr.ReadLine()) != null) //Vòng lặp để đọc từng dòng và gán nó cho chuỗi s, nếu chuỗi rỗng (hết dữ liệu) thì dừng

{

ThueBao n = new ThueBao(s); //Gán thuê bao s mới cho n

thueBaos.Add(n); //Thêm thuê bao mới đó vào mảng ThueBao

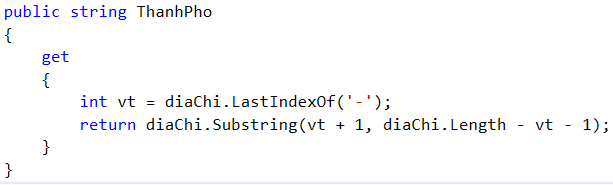
}

}

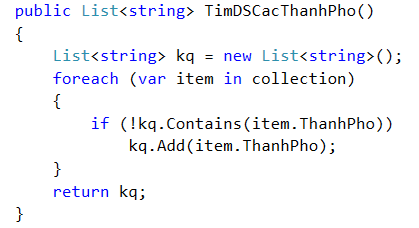
**Phần 2**

1. Trong lớp DanhBa viết phương thức Tìm thành phố có nhiều thuê bao nhất?

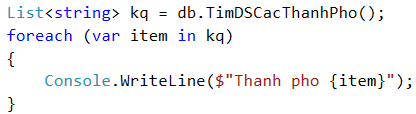
Trong lớp ThueBao bổ sung thuộc tính ThanhPho trích dữ liệu từ trường (Field) diaChi:



Trong lớp DanhBa, bổ sung phương thức sau:



Trong hàm Main, thực hiện đoạn mã sau và kiểm tra kết quả của thực hiện phương thức:



Giải thích các dòng lệnh ở trên và kết quả thực hiện chương trình:

/// <summary>

/// Thuộc tính chỉ đọc ThanhPho, trích xuất Thành phố từ diaChi

/// </summary>

public string ThanhPho

{

get

{

int vt = diaChi.LastIndexOf("-"); //lưu Vị trí xuất hiện cuối cùng của dấu '-' vào vt

return diaChi.Substring(vt + 1, diaChi.Length - 1 - vt); //Lấy một phần của diaChi từ dấu '-' đến hết chuỗi

}

}

/// <summary>

/// Lấy danh sách các thành phố

/// </summary>

/// <returns>Danh sách các thành phố</returns>

public List<string> TimDSCacThanhPho()

{

List<string> list = new List<string>(); //Khai báo mảng lưu các thành phố

foreach (var i in thueBaos) //Vòng lặp duyệt qua mảng chứa các thuê bao

{

if (!list.Contains(i.ThanhPho)) //Kiểm tra nếu thành phố đó không nằm trong mảng lưu các thành phố

{

list.Add(i.ThanhPho); //Thêm thành phố vào mảng

}

}

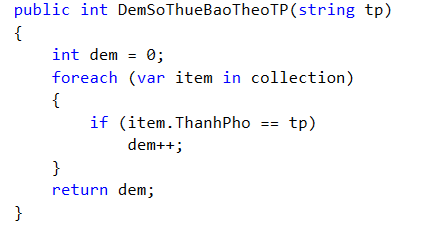
return list; //Trả về mảng gồm danh sách các thành phố

}

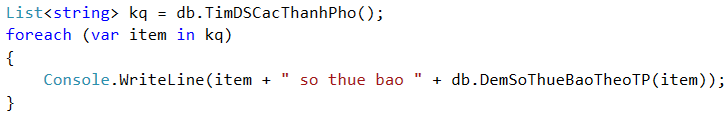
Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình, thuật in máy

Nội dung do AI tạo ra có thể không chính xác.

Trong lớp DanhBa, bổ sung phương thức sau:



Trong hàm Main, thực hiện đoạn mã sau và kiểm tra kết quả của thực hiện phương thức:



Giải thích các dòng lệnh ở trên và kết quả thực hiện chương trình:

/// <summary>

/// Đếm các thuê bao của một thành phố

/// </summary>

/// <param name="tp">Tên thành phố cần đếm</param>

/// <returns>Số thuê bao của thành phố đó</returns>

public int DemSoThueBaoTheoTP(string tp)

{

int dem = 0; //Khai báo biến đếm

foreach(var i in thueBaos) //Vòng lặp duyệt qua mảng chứa các thuê bao

{

if (i.ThanhPho == tp) //Nếu thành phố của thuê bao trùng với thành phố đang cần đếm

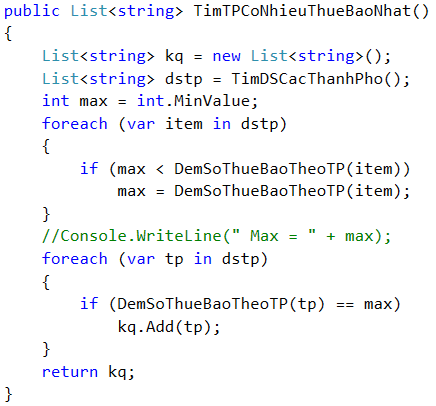
dem++; //Tăng biến đếm lên 1

}

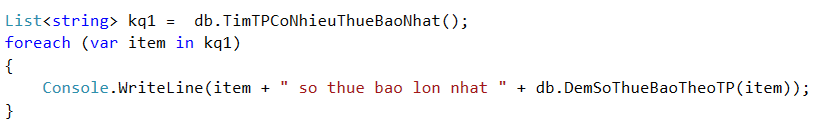
return dem; //Trả về số thuê bao của thành phố đó

}

Trong lớp DanhBa, bổ sung phương thức sau:



Trong hàm Main, thực hiện đoạn mã sau và kiểm tra kết quả của thực hiện phương thức:



Giải thích các dòng lệnh ở trên và kết quả thực hiện chương trình:

/// <summary>

/// Tìm thành phố có nhiều thuê bao nhất

/// </summary>

/// <returns>Danh sáchtên thành phố</returns>

public List<string> TimTPCoNhieuThueBaoNhat()

{

List<string> kq = new List<string>(); //Khai báo mảng chứa tên các thành phố có nhiều thuê bao nhất

List<string> dstp = TimDSCacThanhPho(); //Khai báo danh sách chứa tên các thành phố

int max = int.MinValue; //Khai báo biến max là số thuê bao nhiều nhất của một thành phố

foreach (var i in dstp) //Vòng lặp duyệt qua dstp

{

if (max < DemSoThueBaoTheoTP(i)) //Nếu max nhỏ hơn số thuê bao của thành phố đó

max = DemSoThueBaoTheoTP(i); //Gán số thuê bao của thành phố đó cho max

}

foreach (var i in dstp) //Duyệt qua dstp

{

if(DemSoThueBaoTheoTP(i) == max) //Nếu số thuê bao của thành phố đó = max

kq.Add(i); //Thêm tên tp đó vào mảng chứa tên các tp có nhiều thuê bao nhất

}

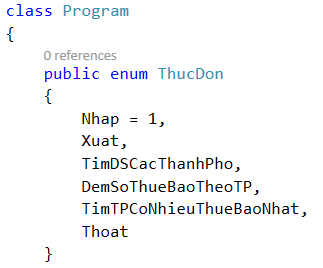
return kq; //Trả về mảng gồm tên các thành phố có nhiều thuê bao nhất

}

**Phần 3**

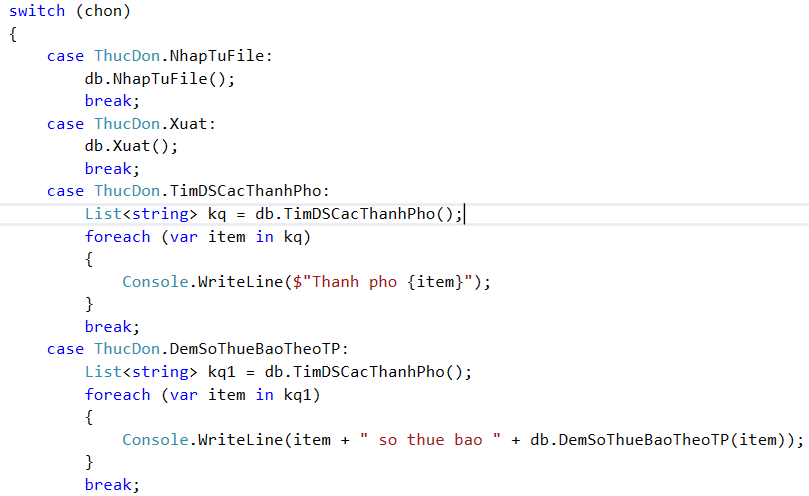
Sử dụng enum để tạo thực đơn vào gọi các chức năng của hệ thống:

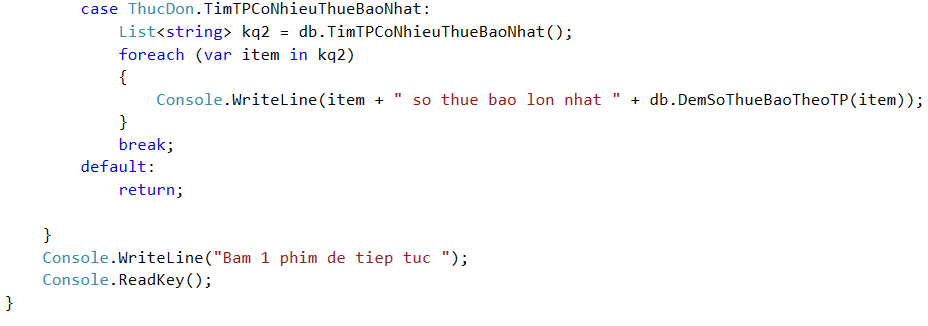
Trong lớp Program tạo kiểu enum như sau:



Xóa hết hàm Main củ và nhập nội dung sau:







Giải thích các dòng lệnh ở trên và kết quả thực hiện chương trình:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, màu đen

Nội dung do AI tạo ra có thể không chính xác.

**Phần 4**

1. Thêm một thuê bao bằng tay.
2. Tìm tất cả thành phố có số thuê bao là x.
3. Tìm thành phố có ít thuê bao nhất.
4. Tìm thuê bao sở hữu ít số điện thoại nhất.
5. Sắp xếp khách hàng tăng giảm theo họ tên, số lượng số điện thoại sở hữu.
6. Hiển thị danh sách các thành phố theo chiều tăng, giảm của số lượng thuê bao.
7. Hiển thị danh bạ theo chiều tăng, giảm của các trường trong thuê bao.
8. Bổ sung thuộc tính ngày đăng ký thuê bao,tìm tháng không có thuê bao nào đăng ký.
9. Tìm tất cả các khách hàng theo giới tính.
10. Xóa tất cả khách hàng thuộc một tỉnh nào đó.
11. Tất cả khách hàng nào sinh tháng 1 thì được tặng thêm một số điện thoại mới có số là cmnd.
12. Tìm ngày có nhiều khách hàng đăng ký nhất, ít người đăng ký nhất.
13. Thống kê và hiển thị dữ liệu theo từng tỉnh và mỗi tỉnh hiển thị theo thành phố theo mẫu sau:

Tỉnh: Lâm Đồng (tổng số thuê bao: 4)

Thành Phố: Dalat (tổng số thuê bao: 2)

1. 001, nguyen van a, 01 PDTV, Dalat, Lam Dong, 123
2. 002, nguyen van b, 01 PDTV, Dalat, Lam Dong, 123

Thành phố bảo lộc: (Tổng số thuê bao: 2)

---Hiển thị danh sách thuê bao ở thành phố bảo lộc

Tỉnh Khánh Hòa (Tổng số….)

Thành Phố: Nha Trang (tổng số thuê bao:……)

---Danh sách thuê bao

**Yêu cầu thêm:**

Ứng dụng cần quản lý thêm thuê bao cố định và thuê bao di động. Đối với thuê bao cố định quản lý thêm thuộc tính ngày lắp đặt, đối với thuê báo di động quản lý thêm thuộc tính nhà dịch vụ (ví dụ: VNPT, Viettel…)

Bổ sung các thuộc tính cần thiết và thực hiện các yêu cầu sau:

1. Tìm thành phố có nhiều thuê bao cố định nhất, có ít thuê bao nhất di động nhất.
2. Tìm thuê bao sở hữu ít số điện thoại cố định nhất.
3. Tìm tháng không có thuê bao nào đăng ký số cố định, di động.
4. Tìm tất cả các thuê bao di động theo giới tính.
5. Xóa tất cả thuê bao theo ngày lắp đặt.
6. Tìm khách hàng di động theo nhà cung cấp dịch vụ
7. Hiển thị số lượng thuê bao của từng loại hình thuê bao

Hiển thị số lượng thuê bao cố định theo từng nhà cung cấp dịch vụ