CODE THAM KHẢO LAB 5

Bài 5.1

//file Phone.cs

namespace Lab5

{

abstract class Phone

{

private string num;//số điện thoại

public Phone()

{

num = "0911111122";

}

public Phone (string p)

{

num = p;

}

public string Num { get => num; set => num = value; }

public abstract string SearchPhone(string name1);

}

//lớp Contact

class Contact:Phone

{

string name;

public string Name { get => name; set => name = value; }

//phương thức thiết lập

public Contact ():base()

{

name = "A";

}

public Contact(string n, string p):base(p)

{

name = n;

}

public Contact (Contact c)

{

name = c.name;

Num = c.Num;

}

//override phương thức trừu tượng

public override string SearchPhone(string s)

{

string n = "";

if (name.CompareTo(s) == 0) //String.Compare(name,s)==0

n = Num;//trả về số điện thoại của người tên s

return n;

}

//phương thức in thông tin

public void Output()

{

Console.WriteLine("{0}\t{1}", name, Num);

}

//cách khác override phương thức ToString()

public override string ToString()

{

return name+"\t"+Num;

}

//toán tử so sánh theo thuộc tính name

public static bool operator>(Contact c1,Contact c2)

{

if (String.Compare(c1.Name, c2.Name) == 1)

return true;

else return false;

}

public static bool operator <(Contact c1, Contact c2)

{

if (String.Compare(c1.Name, c2.Name) ==- 1) return true;

else return false;

}

}

//lớp PhoneBook

class PhoneBook

{

int n;

Contact[] ds;

public PhoneBook()

{

n = 0;

}

public PhoneBook (int n1)

{

n = n1;

ds = new Contact[n+1];

//nhập n Contact

for(int i = 0; i < n; i++)

{

Console.Write("nhap name, phone\_num:");

string name = Console.ReadLine();

string phone\_num = Console.ReadLine();

ds[i] = new Contact(name, phone\_num);

}

}

//thêm đối tượng vào vị trí pos

public void InSert(Contact c, byte pos)

{

int i = n;

ds[n] = new Contact();

for (i = n; i > pos; i--)

ds[i] = ds[i-1];

ds[i] = c;

n++;

}

//xóa đối tượng Contact đầu tiên có name truyền vào

public void Remove(string name)

{

for(int i=0;i<n;i++)

if (String.Compare (ds[i].Name,name )==0)

{

for (int j = i; j < n-1; j++)

ds[j] = ds[j + 1];

n--;

break;

}

}

//Cập nhật

public void Update(string name, string phonenew)

{

for (int i = 0; i < n; i++)

if (String.Compare(ds[i].Name, name) == 0)

{

ds[i].Num = phonenew;

}

}

//sắp xếp bằng thuật toán bubble sort

public void Sort()

{

Contact c= new Contact();

for (int i=0;i<n;i++)

for(int j=i+1;j<n;j++)

if (ds[i]>ds[j])

{

Console.WriteLine("T");

c = ds[i];

ds[i] = ds[j];

ds[j] = c;

}

}

//xuất danh bạ

public void Output()

{

for(int i=0;i<n;i++)

Console.WriteLine(ds[i].ToString());

// ds[i].Output();

}

}

}

//file Program.cs

namespace Lab5

{

class Program

{

static void Main(string[] args)

{

#region BT5\_1

Console.WriteLine("nhap danh ba dien thoai:");

PhoneBook ls = new PhoneBook(2);

//nhập thông tin đối tượng cần thêm

//Console.Write("nhap name, phone can them:");

//string name = Console.ReadLine();

//string phone\_num = Console.ReadLine();

//Contact c = new Contact(name, phone\_num);

//Console.Write("nhap vi tri can them:");

//byte i = byte.Parse(Console.ReadLine());//SV tự kiểm tra ràng buộc giá trị

//ls.InSert(c, i);

////xóa phần tử có name đầu tiên

//Console.Write("nhap name can xoa:");

//string name1 = Console.ReadLine();

//ls.Remove(name1);

////nhập thông tin đối tượng cần cập nhật

//Console.Write("nhap name, phone can cap nhat:");

//string name = Console.ReadLine();

//string phone\_new = Console.ReadLine();

//ls.Update(name, phone\_new);

ls.Sort();

ls.Output();

Console.ReadKey();

}

}

}

Bài 5.2

//file Phone1.cs

namespace Lab5

{

//lớp Contact sử dụng các lớp String, List, FileStream

class Contact1 : Phone, IComparable <Contact1>

{

string name;

public string Name { get => name; set => name = value; }

//phương thức thiết lập

public Contact1() : base()

{

name = "A";

}

public Contact1(string n, string p) : base(p)

{

name = n;

}

public Contact1(Contact c)

{

name = c.Name;

Num = c.Num;

}

//override phương thức trừu tượng

public override string SearchPhone(string s)

{

string n = "";

if (name.CompareTo(s) == 0)

n = name;

return n;

}

//phương thức in thông tin

public void Output()

{

Console.WriteLine("{0}\t{1}", name, Num);

}

//cách khác override phương thức ToString()

public override string ToString()

{

return name + "\t" + Num;

}

//phương thức so sánh

public int CompareTo(Contact1 c)

{

if (name.CompareTo (c.name )>1) return 1;

else if (name.CompareTo(c.name) ==0) return 0;

else return -1;

}

}

//lớp PhoneBook

class PhoneBook1

{

int n;

List <Contact1> ls;

public PhoneBook1()

{

ls = new List<Contact1>();

try

{

FileStream f = new FileStream("E:\\PhoneBook.txt", FileMode.Open);

StreamReader rd = new StreamReader(f, Encoding.UTF8);

string line;

n = int.Parse(rd.ReadLine());//đọc từng dòng dữ liệu từ file

// doc du lieu tu file

while ((line = rd.ReadLine()) != null)

{

string name = line;

string phone\_num = rd.ReadLine();

Contact1 c = new Contact1(name, phone\_num);

ls.Add(c);

}

}

catch

{ Console.WriteLine("Khong the doc file da cho: "); }

}

//thêm đối tượng vào vị trí pos

public void InSert(Contact1 c, byte pos)

{

ls.Insert(pos, c);

}

//xóa đối tượng Contact đầu tiên có name truyền vào

public void Remove(string name)

{

int i = 0;

for (i = 0; i < n; i++)

if (String.Compare(ls[i].Name, name) == 0)

{

break;

}

if (i < ls.Count)//có người tên name trong danh bạ

ls.RemoveAt(i);

}

//Cập nhật

public void Update(string name, string phonenew)

{

for (int i = 0; i < n; i++)

if (String.Compare(ls[i].Name, name) == 0)

{

ls[i].Num = phonenew;

}

}

//sắp xếp qua phương thức của interface

public void Sort()

{

ls.Sort();

}

//xuất danh bạ

public void Output()

{

for (int i = 0; i < n; i++)

Console.WriteLine(ls[i].ToString());

// c.Output();

}

}

//lớp PhoneBook sử dụng filestream

class PhoneBook2

{

int n;

List<Contact1> ls;

public PhoneBook2()

{

ls = new List<Contact1>();

//đọc dữ liệu từ file đưa vào ls, mỗi giá trị của thuộc tính lưu trên 1 dòng trong file

try

{

FileStream f = new FileStream("E:\\PhoneBook.txt", FileMode.Open);

StreamReader rd = new StreamReader(f, Encoding.UTF8);

string line;

n=int.Parse (rd.ReadLine());//đọc từng dòng dữ liệu từ file

// doc du lieu tu file

while ((line = rd.ReadLine()) != null)

{

string name = line ;

string phone\_num = rd.ReadLine();

Contact1 c = new Contact1(name, phone\_num);

ls.Add(c);

}

}

catch

{ Console.WriteLine("Khong the doc file da cho: "); }

}

//thêm đối tượng vào vị trí pos

public void InSert(Contact1 c, byte pos)

{

ls.Insert(pos, c);

}

//xóa đối tượng Contact đầu tiên có name truyền vào

public void Remove(string name)

{

int i = 0;

for (i = 0; i < n; i++)

if (String.Compare(ls[i].Name, name) == 0)

{

break;

}

if (i < ls.Count)

ls.RemoveAt(i);

}

//Cập nhật

public void Update(string name, string phonenew)

{

for (int i = 0; i < n; i++)

if (String.Compare(ls[i].Name, name) == 0)

{

ls[i].Num = phonenew;

}

}

//sắp xếp qua phương thức của interface

public void Sort()

{

ls.Sort();

}

//xuất danh bạ

public void Output()

{

for (int i = 0; i < n; i++)

Console.WriteLine(ls[i].ToString());

// ls[i].Output();

}

}

}

//chương trình chính tương tự BT5.1

Bài 5.3

//file News.cs

namespace Lab5

{

interface INews

{

int Id { get; set; }

string Title { get; set; }

string Author { get; set; }

void Display();

}

class News:INews

{

int id;

string title, author;

DateTime publishDate;

string content;

List<float> avgRate;

public int Id { get => id; set => id = value; }

public string Title { get => title; set => title = value; }

public string Author { get => author; set => author = value; }

//viết cụ thể phương thức Display()

public void Display()

{

//sv tự viết các lệnh theo yêu cầu đề bài

}

//các phương thức còn lại sv tự viết

}

//class danh sách các tin tức

class ListNews

{

//SV tự viết

}

}

//chương trình chính SV tự viết