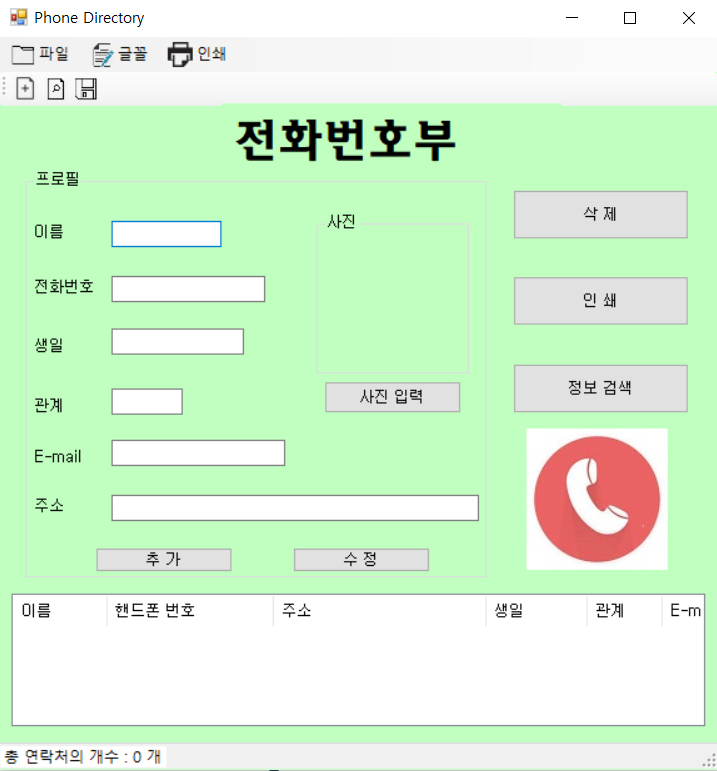
윈도우 프로그래밍

기말 프로젝트

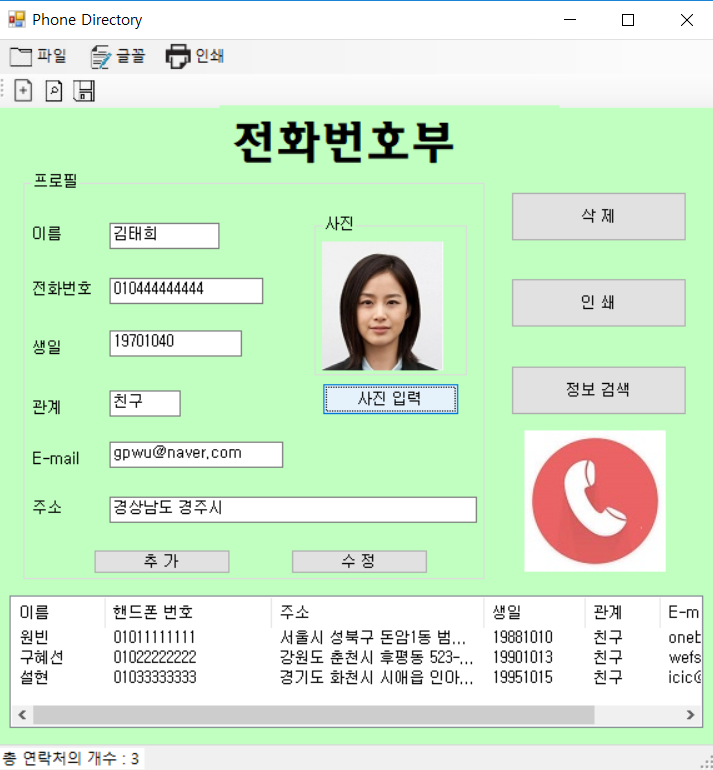
컴퓨터공학과

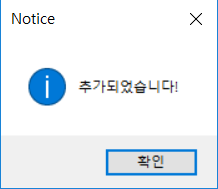
20145150

이상욱



실행하면 나오는 첫번째 폼이다. 용도는 연락처를 저장, 수정 삭제 할 수 있는 전화번호부 이다. 연락처가 많아질 경우 정보검색을 쉽게 할 수 있게 검색기능도 추가하였다. 1개의 MenuStrip, 1개의 ToolStrip, 1개의 statusStrip, button 6개, textbox 6개 , 1개의 listview ,2개의 pictureBox , 2개의 groupBox , 7개의 라벨로 구성된 메인 폼이다



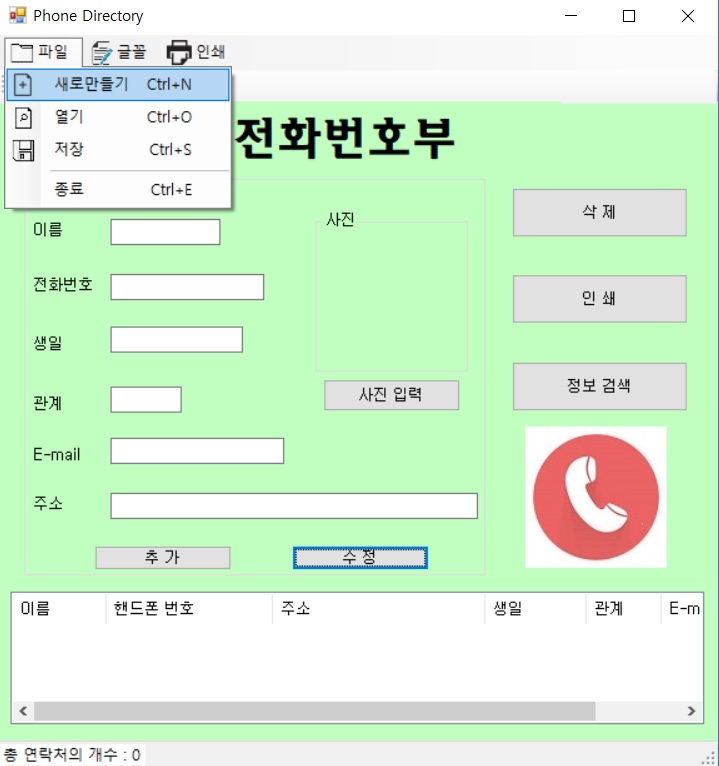
연락처 추가 버튼을 활용해 정보를 모두 입력하면 리스트뷰에 차례대로 입력이 된다. 연락처가 증가 할 때 마다 statusStrip의 연락처의 개수 가 증가한다. 사진 입력 버튼을 누르고 이미지 파일을 넣으면 pictureBox에 사진이 뜬다.



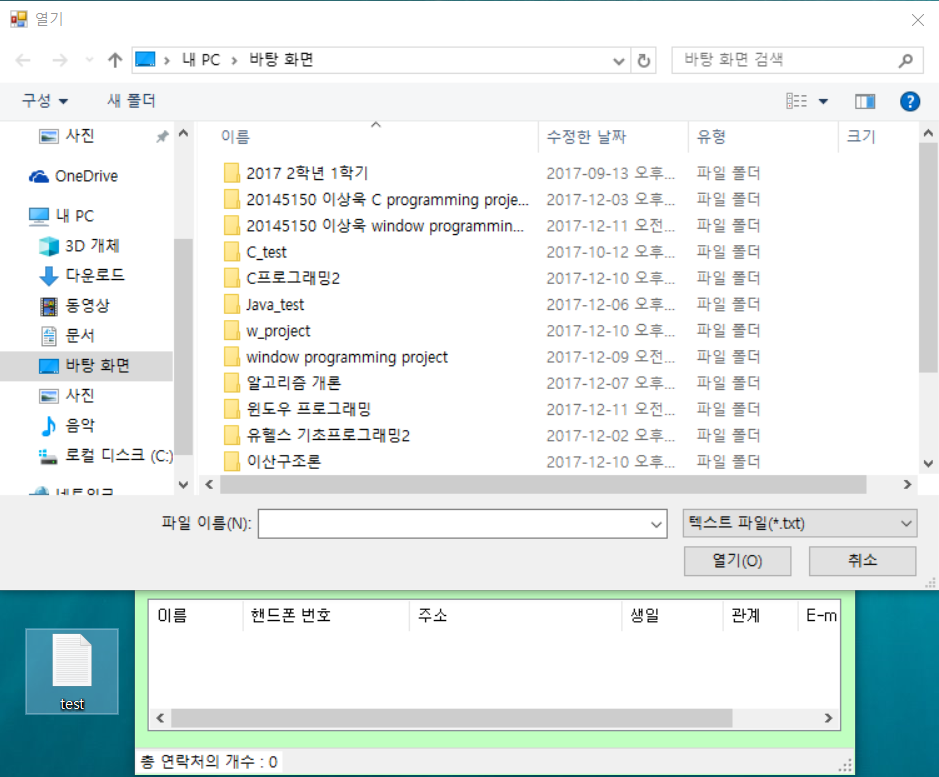
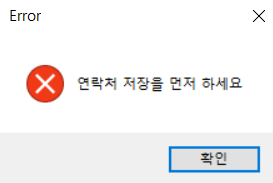
리스트뷰에서 삭제하고 싶은 연락처를 클릭 후에 삭제를 누르면 그 줄의 내용이 삭제 되고 알림 MessageBox가 뜬다 총연락처의 개수도 하나 감소 한다.

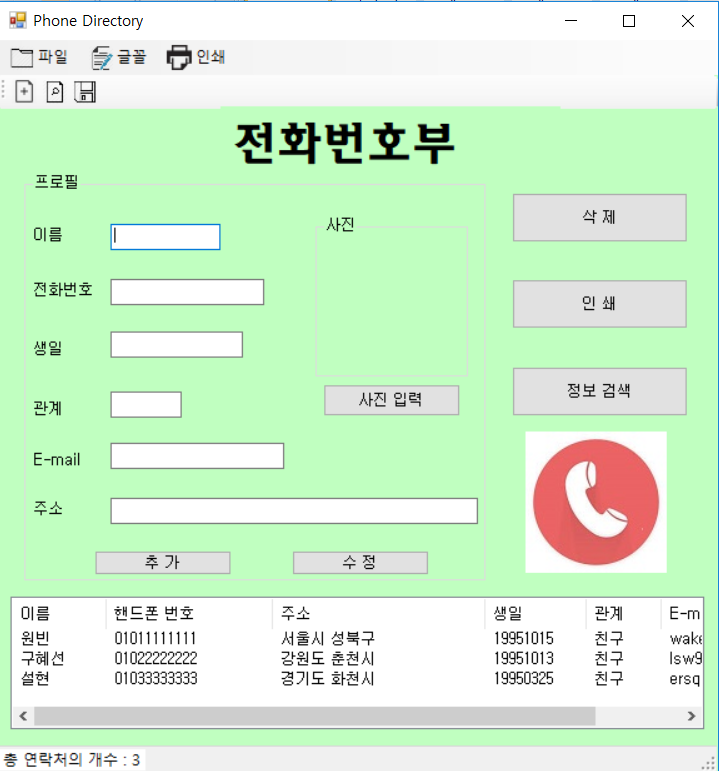


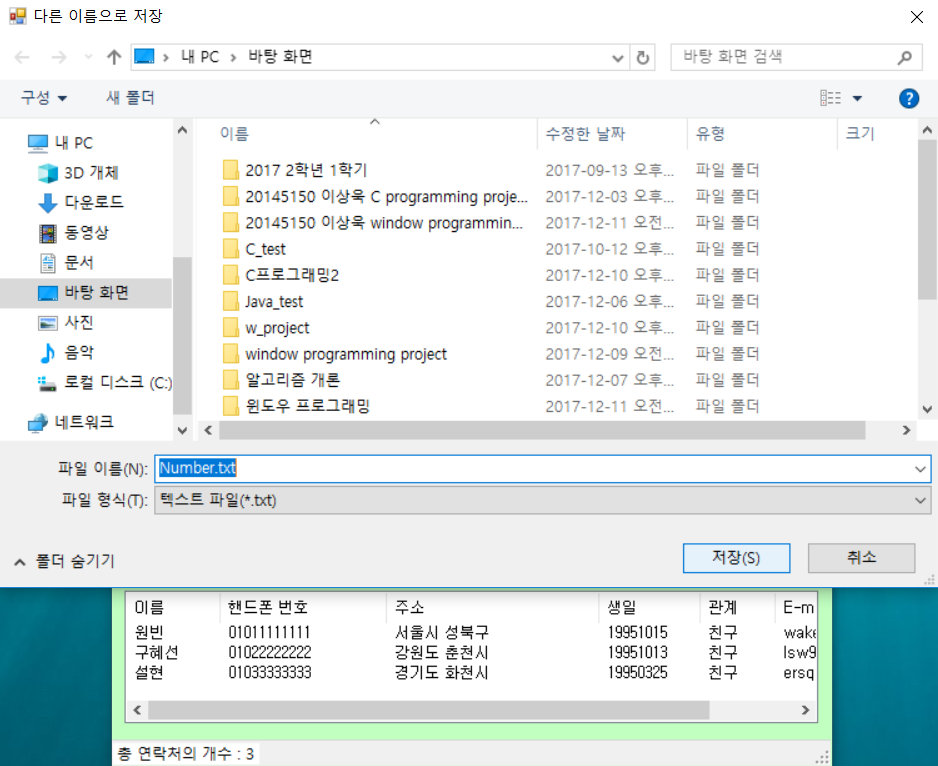
리스트뷰에서 수정하고 싶은 내용을 textBox에 먼저 입력후 리스트뷰에서 수정하고 싶은 연락처를 클릭 후 수정버튼을 누르면 내용이 수정되고 MessageBox가 뜬다.



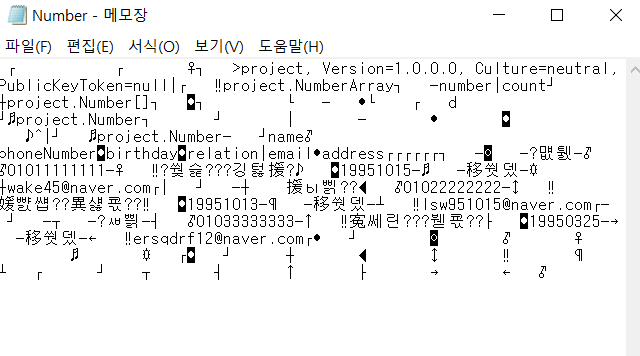
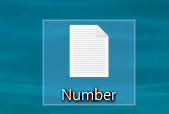
MenuStrip에 파일 목록에 새로만들기를 누르거나 ToolStrip의 새로만들기 아이콘을 클릭면 모든 textBox의 내용이 사라지고 리스트 뷰도 처음으로 초기화된다. statusStrip도 연락처의 개수가 0으로 초기화된다. 단축기는 Ctrl + N(new)로 정하였다.

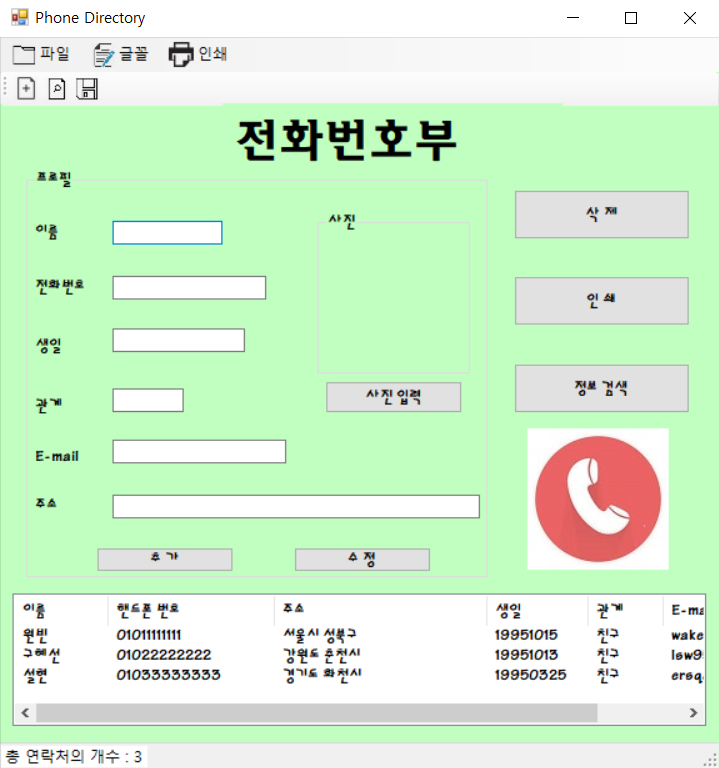
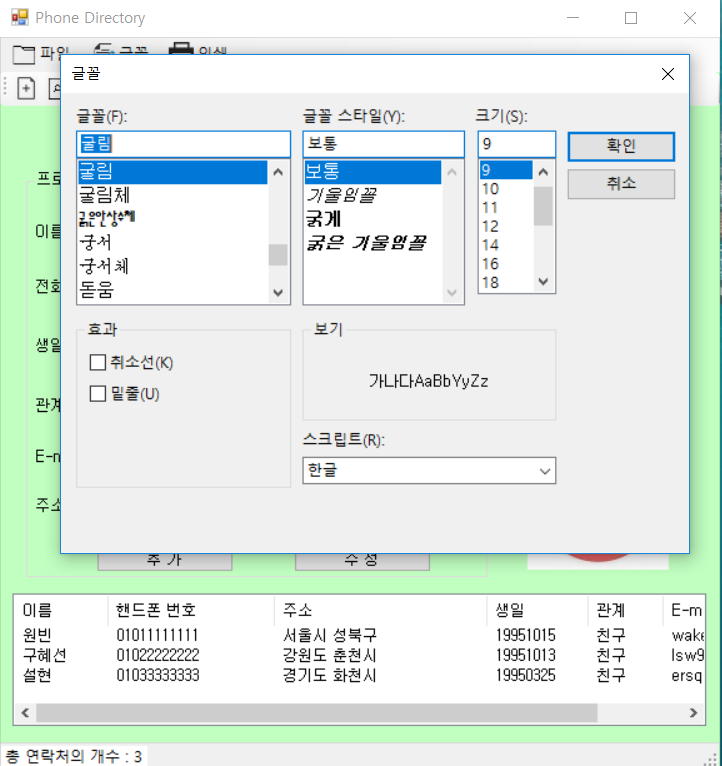


MenuStrip에 파일 목록에 열기를 누르거나 ToolStrip의 파일 아이콘을 클릭하면 FileOpenDialog가 뜬다. dialog에서 전에 저장했던 파일을 클릭하고 열기를 누르면 내용 그대로 리스트뷰에 표시되고 statusStrip도 연락처의 개수만큼 초기화된다. 파일에 내용이 없을 경우 Error메세지 박스를 출력한다. 단축키는 Ctrl + O(open) 로 정하였다.

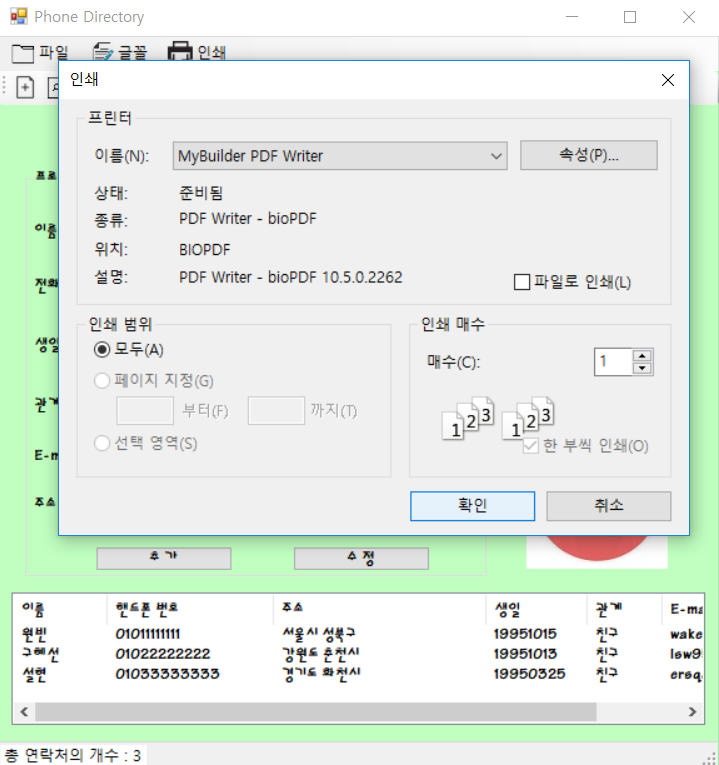


MenuStrip에 파일 목록에 저장을 누르거나 ToolStrip의 디스켓 아이콘을 클릭면 SaveFileDialog가 뜬다. dialog에서 저장 할 파일 명을 입력하고 저장을 누르면 내용이 파일에 저장이 된다. 단 이진파일이므로 txt파일로 사람이 읽기는 불가능하다. 단축키는 Ctrl + S(save)이다.

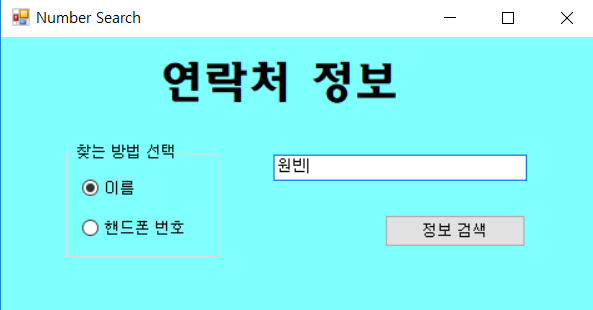


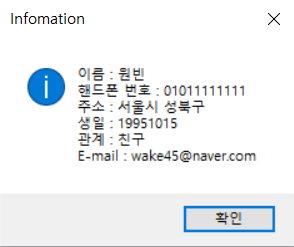
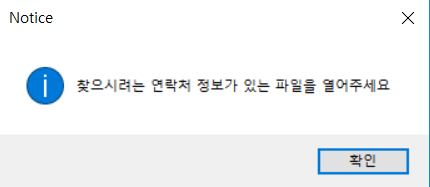


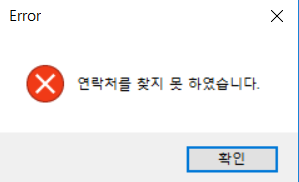
글꼴 버튼을 누르면 FontDialog 창이 뜨고 변경한 폰트,글꼴 스타일,글자크기 대로 폼의 글자가 변경된다.



인쇄버튼을 누르거나 ToolStrip의 프린트 아이콘 또는 MenuStrip의 인쇄버튼을 누르면 PrintDialog창이 뜬다.

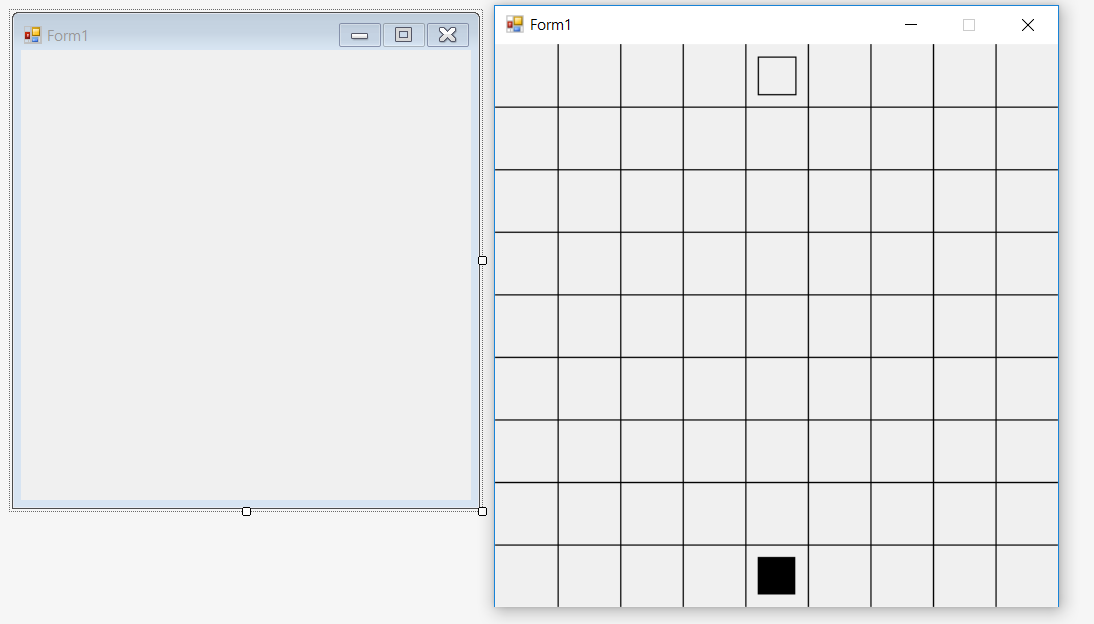


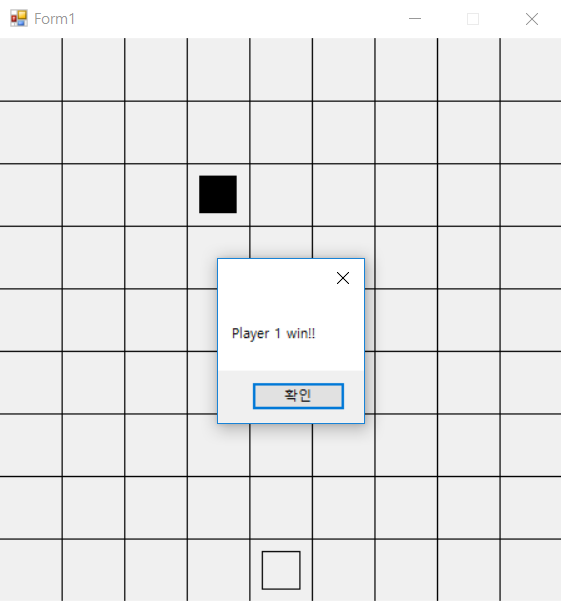
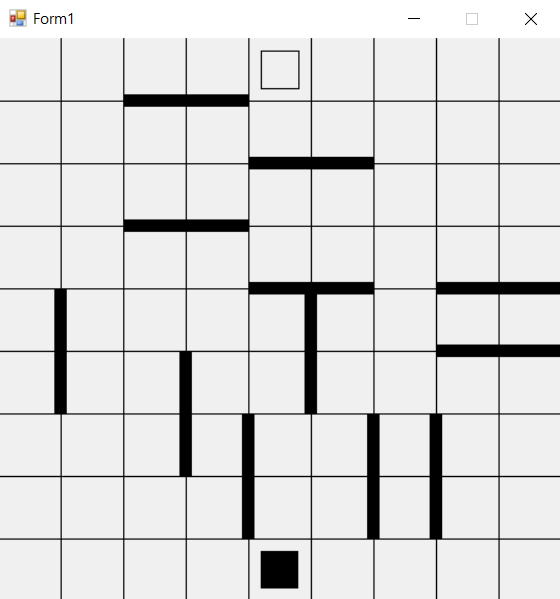


정보 검색 버튼을 누르면 이름으로 찾고 싶은지 핸드폰 번호로 찾고 싶은지 RadioButton을 체크하고 찾고자 하는 이름이나 번호를 textBox에 입력하고 정보검색을 누르면 OpenFileDialog가 띄워진다. 그 후 전화번호를 저장한 파일을 열면 그 연락처를 찾아 그 연락처가 가진 모든 정보를 MessageBox로 출력 한다. 찾는 정보가 없을경우 Error메시지박스 출력한다.

프로젝트를 마치고

메인 폼에서 사진을 입력을 받는데 사진 입력 받은 것까지 파일에 저장을 하고 불러오기를 할 수 있고 정보 검색할 때 사진도 나오게 만들었으면 좋았을 것 같았다. 하지만 아직 사진을 파일처리해서 글과 함께 저장하는 법을 안 배워서 아쉬웠다. 그리고 원래는 쿼리도 라는 게임(각자 플레이어의 말을 상대 진영으로 가지고 가면 이기는 게임이다. 벽을 세워서 진로 방해를 최대한으로 해서 적의 이동 경로를 늘려 먼저 도착해 승리를 하는 게임)을 만들려고 했었다. 교수님께 허락을 받고 거의 다 완성해가는 단계 였는데 벽을 세우는 과정에서 벽으로 사각형을 만들어서 말을 가둘 경우와 벽이 앞에 있을 경우 못 넘어가게 하는 방법을 아무리 생각해도 떠오르지가 않았다. 그러다 보니 제출기한이 얼마 남지 않아 프로젝트를 급하게 변경하게 되어서 아쉬운 점이 남았다.





메인 폼의 코드 입니다.

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.IO;

using System.Drawing.Printing;

using System.Runtime.Serialization;

using System.Runtime.Serialization.Formatters.Binary;

namespace project

{

public partial class Form1 : Form

{

NumberArray numberArray = new NumberArray(); //Number클래스를 가지고 있는 NumberArray클래스 객체 생성

Number Num; //Number클래스 변수

int count = 0; //저장되어 있는 연락처의 개수를 셀 변수

public Form1()

{

InitializeComponent();

}

private void NewToolStripMenuItem\_Click(object sender, EventArgs e) //toolstrip메뉴의 새로만들기 버튼

{ //리스트뷰와 텍스트 박스에 있는 모든 내용을 초기화

NametextBox.Text = "";

PhoneNumbertextBox.Text = "";

BirthDaytextBox.Text = "";

RelationtextBox.Text = "";

EmailtextBox.Text = "";

AddresstextBox.Text = "";

listView1.Items.Clear();

count = 0;

toolStripStatusLabel.Text = "총 연락처의 개수 : " + count;

}

private void NewStripButton\_Click(object sender, EventArgs e) //메뉴의 새로만들기 버튼

{ //리스트뷰와 텍스트 박스에 있는 모든 내용을 초기화

NametextBox.Text = "";

PhoneNumbertextBox.Text = "";

BirthDaytextBox.Text = "";

RelationtextBox.Text = "";

EmailtextBox.Text = "";

AddresstextBox.Text = "";

listView1.Items.Clear();

count = 0;

toolStripStatusLabel.Text = "총 연락처의 개수 : " + count;

}

private void OpentoolStripButton\_Click(object sender, EventArgs e) //toolstrip메뉴의 새로만들이 버튼

{

OpenFileDialog OFD = new OpenFileDialog(); //파일오픈에 필요한 dialog객체 생성

OFD.Filter = "텍스트 파일(\*.txt)|\*.txt|모든 파일(\*.\*)|\*.\*"; //볼 파일만 나오게함

OFD.FilterIndex = 1; //필터 인덱스 설정

OFD.RestoreDirectory = true; //이전의 선택한 디렉터리로 복원가능

OFD.Multiselect = true; //여러대상 선택가능

DialogResult dr = OFD.ShowDialog(); //파일오픈 dialog창 띄우기

try

{

if (dr == DialogResult.OK) //파일이 선택되면

{

listView1.Items.Clear(); //리스트뷰 초기화

FileStream f = new FileStream(OFD.FileName, FileMode.Open); // open형식으로 파일 열기

BinaryFormatter bf = new BinaryFormatter();

numberArray = (NumberArray)bf.Deserialize(f); //deserialize

for (int i = 0; i < numberArray.Count; i++) //numberArray에 들어가있는 배열 크기만큼 반복

{

ListViewItem item = new ListViewItem(numberArray[i].Name); //파일에 저장되어있는 정보를 numberArray 각각의 자리에 대입

item.SubItems.Add(numberArray[i].PhoneNumber);

item.SubItems.Add(numberArray[i].Address);

item.SubItems.Add(numberArray[i].Birthday);

item.SubItems.Add(numberArray[i].Relation);

item.SubItems.Add(numberArray[i].Email);

count++;

listView1.Items.Add(item); //리스트 뷰에 표시

}

f.Close(); // 파일닫기

}

}

catch (SerializationException se) //연락처 저장이 안되어 있을 경우

{

MessageBox.Show("연락처 저장을 먼저 하세요","Error",MessageBoxButtons.OK,MessageBoxIcon.Error);

}

toolStripStatusLabel.Text = "총 연락처의 개수 : " + count;

}

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

{

OpenFileDialog OFD = new OpenFileDialog(); //파일오픈에 필요한 dialog객체 생성

OFD.Filter = "텍스트 파일(\*.txt)|\*.txt|모든 파일(\*.\*)|\*.\*"; //볼 파일만 나오게함

OFD.FilterIndex = 1; //필터 인덱스 설정

OFD.RestoreDirectory = true; //이전의 선택한 디렉터리로 복원가능

OFD.Multiselect = true; //여러대상 선택가능

DialogResult dr = OFD.ShowDialog(); //파일오픈 dialog창 띄우기

try

{

if (dr == DialogResult.OK) //파일이 선택되면

{

listView1.Items.Clear(); //리스트뷰 초기화

FileStream f = new FileStream(OFD.FileName, FileMode.Open); // open형식으로 파일 열기

BinaryFormatter bf = new BinaryFormatter();

numberArray = (NumberArray)bf.Deserialize(f); //deserialize

for (int i = 0; i < numberArray.Count; i++) //numberArray에 들어가있는 배열 크기만큼 반복

{

ListViewItem item = new ListViewItem(numberArray[i].Name); //파일에 저장되어있는 정보를 numberArray 각각의 자리에 대입

item.SubItems.Add(numberArray[i].PhoneNumber);

item.SubItems.Add(numberArray[i].Birthday);

item.SubItems.Add(numberArray[i].Relation);

item.SubItems.Add(numberArray[i].Email);

item.SubItems.Add(numberArray[i].Address);

count++;

listView1.Items.Add(item); //리스트 뷰에 표시

}

f.Close(); // 파일닫기

}

}

catch (SerializationException se) //연락처 저장이 안되어 있을 경우

{

MessageBox.Show("연락처 저장을 먼저 하세요", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

toolStripStatusLabel.Text = "총 연락처의 개수 : " + count;

}

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

{

try

{

SaveFileDialog SFD = new SaveFileDialog(); //파일저장 dialog 객체생성

for (int i = 0; i < listView1.Items.Count; i++) //numberArray에 들어가있는 배열 크기만큼 반복

{

Num = new Number(listView1.Items[i].SubItems[0].Text, listView1.Items[i].SubItems[1].Text, //리스트 뷰의 내용을 num변수에 저장

listView1.Items[i].SubItems[2].Text, listView1.Items[i].SubItems[3].Text,

listView1.Items[i].SubItems[4].Text, listView1.Items[i].SubItems[5].Text);

numberArray[numberArray.Count++] = Num; //numberarray배열에 인덱스를 증가시키면서 num저장

}

SFD.Filter = "텍스트 파일(\*.txt)|\*.txt|모든 파일(\*.\*)|\*.\*"; //볼 파일 필터

SFD.FilterIndex = 1; //필터 인덱스 저장

SFD.RestoreDirectory = true; //이전의 선택한 디렉터리로 복원가능

DialogResult dr = SFD.ShowDialog(); //파일저장 dialog 창 띄우기

if (dr == DialogResult.OK) //저장 버튼 눌렀을 경우

{

FileStream fs = new FileStream(SFD.FileName, FileMode.Create); //파일저장 바이트단위 접근

BinaryFormatter bf = new BinaryFormatter();

bf.Serialize(fs, numberArray);

fs.Flush();

fs.Close(); //파일닫기

}

}

catch (NullReferenceException ne) //연락처 추가가 안되어 있을 경우

{

}

}

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

{

try

{

SaveFileDialog SFD = new SaveFileDialog(); //파일저장 dialog 객체생성

for (int i = 0; i < listView1.Items.Count; i++) //numberArray에 들어가있는 배열 크기만큼 반복

{

Num = new Number(listView1.Items[i].SubItems[0].Text, listView1.Items[i].SubItems[1].Text, //리스트 뷰의 내용을 num변수에 저장

listView1.Items[i].SubItems[2].Text, listView1.Items[i].SubItems[3].Text,

listView1.Items[i].SubItems[4].Text, listView1.Items[i].SubItems[5].Text);

numberArray[numberArray.Count++] = Num; //numberarray배열에 인덱스를 증가시키면서 num저장

}

SFD.Filter = "텍스트 파일(\*.txt)|\*.txt|모든 파일(\*.\*)|\*.\*"; //볼 파일 필터

SFD.FilterIndex = 1; //필터 인덱스 저장

SFD.RestoreDirectory = true; //이전의 선택한 디렉터리로 복원가능

DialogResult dr = SFD.ShowDialog(); //파일저장 dialog 창 띄우기

if (dr == DialogResult.OK) //저장 버튼 눌렀을 경우

{

FileStream fs = new FileStream(SFD.FileName, FileMode.Create); //파일저장 바이트단위 접근

BinaryFormatter bf = new BinaryFormatter();

bf.Serialize(fs, numberArray);

fs.Flush();

fs.Close(); //파일닫기

}

}

catch (NullReferenceException ne) //연락처 추가가 안되어 있을 경우

{

}

}

private void ExitToolStripMenuItem\_Click(object sender, EventArgs e) //종료버튼 실행

{

this.Close(); //실행 종료

}

private void FontToolStripMenuItem\_Click(object sender, EventArgs e) //폰트 dialog사용해서 폰트 변경

{

FontDialog FD = new FontDialog();

FD.ShowDialog();

this.Font = FD.Font;

}

private void PrintToolStripMenuItem\_Click(object sender, EventArgs e) //인쇄 dialog사용해서 인쇄

{

PrintDialog PD = new PrintDialog();

PrinterSettings printer = new PrinterSettings();

PrintDocument pd = new PrintDocument();

PD.PrinterSettings = printer;

PD.Document = pd;

pd.PrintPage += new PrintPageEventHandler(pd\_PrintPage);

DialogResult dr = PD.ShowDialog();

if (dr == DialogResult.OK)

{

pd.Print();

}

}

private void pd\_PrintPage(object sender, PrintPageEventArgs e)

{ // Print키

string txt = "test.txt";

Font printFont = new Font("Arial", 10, FontStyle.Regular);

e.Graphics.DrawString(txt, printFont, Brushes.Black, 10, 10);

}

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

{

OpenFileDialog OFD = new OpenFileDialog();

OFD.ShowDialog();

pictureBox1.Image = Image.FromFile(OFD.FileName);

}

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

{

foreach (ListViewItem item in listView1.SelectedItems)

{

ListViewItem.ListViewSubItemCollection subItem = item.SubItems; //입력된 정보로 리스트뷰 정보 수정

subItem[0].Text = NametextBox.Text;

subItem[1].Text = PhoneNumbertextBox.Text;

subItem[2].Text = AddresstextBox.Text;

subItem[3].Text = BirthDaytextBox.Text;

subItem[4].Text = RelationtextBox.Text;

subItem[5].Text = EmailtextBox.Text;

MessageBox.Show("수정되었습니다!","Notice",MessageBoxButtons.OK,MessageBoxIcon.Asterisk);

NametextBox.Text = "";

PhoneNumbertextBox.Text = "";

BirthDaytextBox.Text = "";

RelationtextBox.Text = "";

EmailtextBox.Text = "";

AddresstextBox.Text = "";

}

}

private void button3\_Click(object sender, EventArgs e) //입력된 정보 리스트뷰에 추가

{

String name = NametextBox.Text;

String phoneNumber = PhoneNumbertextBox.Text;

String birthDay = BirthDaytextBox.Text;

String group = RelationtextBox.Text;

String Email = EmailtextBox.Text;

String homeNumber = AddresstextBox.Text;

System.Windows.Forms.ListViewItem listViewItem = new System.Windows.Forms.ListViewItem(new String[] { name, phoneNumber, homeNumber, birthDay, group, Email });

listView1.Items.AddRange(new System.Windows.Forms.ListViewItem[]{

listViewItem});

NametextBox.Text = "";

PhoneNumbertextBox.Text = "";

BirthDaytextBox.Text = "";

RelationtextBox.Text = "";

EmailtextBox.Text = "";

AddresstextBox.Text = "";

count++;

toolStripStatusLabel.Text = "총 연락처의 개수 : " + count; //연락처의 개수 증가

MessageBox.Show("추가되었습니다!", "Notice", MessageBoxButtons.OK, MessageBoxIcon.Asterisk);

}

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

{

}

private void button4\_Click(object sender, EventArgs e)

{

ListView.SelectedListViewItemCollection selItems = listView1.SelectedItems;

foreach (ListViewItem item in selItems)

{

listView1.Items.Remove(item);

count--;

toolStripStatusLabel.Text = "총 연락처의 개수 : " + count; //연락처의 개수 감소

MessageBox.Show("삭제되었습니다!", "Notice", MessageBoxButtons.OK, MessageBoxIcon.Asterisk);

}

}

private void button5\_Click(object sender, EventArgs e) //인쇄 버튼 실행

{

PrintDialog PD = new PrintDialog();

PrinterSettings printer = new PrinterSettings();

PrintDocument pd = new PrintDocument();

PD.PrinterSettings = printer;

PD.Document = pd;

pd.PrintPage += new PrintPageEventHandler(pd\_PrintPage);

DialogResult dr = PD.ShowDialog();

if (dr == DialogResult.OK)

{

pd.Print();

}

}

private void button6\_Click(object sender, EventArgs e) //정보검색 버튼실행

{

Form2 f2 = new Form2();

f2.Show();

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

}

}

}

Form2의 코드입니다.

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Windows.Forms;

using System.IO;

using System.Drawing.Printing;

using System.Runtime.Serialization;

using System.Runtime.Serialization.Formatters.Binary;

namespace project

{

public partial class Form2 : Form

{

NumberArray numberArray = new NumberArray();

public Form2()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e) //찾는 방법을 이름 전화번호 둘중에 골라 정보를 텍스트 박스에 입력하고 전화번호부 파일을 열어 원하는 정보를 메세지 박스로 출력받음

{

string temp = textBox1.Text;

int runOK = 0;

MessageBox.Show("찾으시려는 연락처 정보가 있는 파일을 열어주세요","Notice",MessageBoxButtons.OK,MessageBoxIcon.Asterisk);

if (radioButton1.Checked) //이름으로 찾을경우

{

OpenFileDialog OFD = new OpenFileDialog(); //파일 열기 form1의 파일열기와 기능은 같음

OFD.Filter = "텍스트 파일(\*.txt)|\*.txt|모든 파일(\*.\*)|\*.\*";

OFD.FilterIndex = 1;

OFD.RestoreDirectory = true;

OFD.Multiselect = true;

DialogResult dr = OFD.ShowDialog();

try

{

if (dr == DialogResult.OK)

{

FileStream f = new FileStream(OFD.FileName, FileMode.Open); // open형식으로 파일 열기

BinaryFormatter bf = new BinaryFormatter();

numberArray = (NumberArray)bf.Deserialize(f); //deserialize

for (int i = 0; i < numberArray.Count; i++)

{

if(numberArray[i].Name == temp) //텍스트 박스 내용과 비교하여 맞으면 그 인덱스의 정보 모두 출력

{

MessageBox.Show("이름 : " + numberArray[i].Name + "\n핸드폰 번호 : " + numberArray[i].PhoneNumber +

"\n주소 : " + numberArray[i].Address + "\n생일 : " + numberArray[i].Birthday + "\n관계 : " + numberArray[i].Relation

+ "\nE-mail : " + numberArray[i].Email,"Infomation", MessageBoxButtons.OK, MessageBoxIcon.Asterisk);

runOK = 999;

break;

}

}

if(runOK != 999) //위의 if문이 실행 되지 않았을 경우 실행

{

MessageBox.Show("연락처를 찾지 못 하였습니다.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

f.Close(); // 파일닫기

}

}

catch (SerializationException se)

{

MessageBox.Show("연락처 저장을 먼저 하세요");

}

}

else //전화번호로 찾을경우

{

OpenFileDialog OFD = new OpenFileDialog();

OFD.Filter = "텍스트 파일(\*.txt)|\*.txt|모든 파일(\*.\*)|\*.\*";

OFD.FilterIndex = 1;

OFD.RestoreDirectory = true;

OFD.Multiselect = true;

DialogResult dr = OFD.ShowDialog();

try

{

if (dr == DialogResult.OK)

{

FileStream f = new FileStream(OFD.FileName, FileMode.Open); // open형식으로 파일 열기

BinaryFormatter bf = new BinaryFormatter();

numberArray = (NumberArray)bf.Deserialize(f); //deserialize

for (int i = 0; i < numberArray.Count; i++)

{

if (numberArray[i].PhoneNumber == temp)

{

MessageBox.Show("이름 : " + numberArray[i].Name + " 핸드폰 번호 : " + numberArray[i].PhoneNumber +

" 주소 : " + numberArray[i].Address + " 생일 : " + numberArray[i].Birthday + " 관계 : " + numberArray[i].Relation

+ " E-mail : " + numberArray[i].Email,"Infomation",MessageBoxButtons.OK,MessageBoxIcon.Asterisk);

runOK = 999;

break;

}

}

if (runOK != 999) //위의 if문이 실행 되지 않았을 경우 실행

{

MessageBox.Show("연락처를 찾지 못 하였습니다.","Error",MessageBoxButtons.OK,MessageBoxIcon.Error);

}

f.Close(); // 파일닫기

}

}

catch (SerializationException se)

{

MessageBox.Show("연락처 저장을 먼저 하세요", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

}

}

Number클래스의 코드입니다.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace project

{

[Serializable] // 직렬화

public class Number

{

string name; // 이름

string phoneNumber; // 핸드폰 번호

string birthday; // 생일

string relation; // 관계

string email; // 메일주소

string address; // 집주소

public Number(string name, string phoneNumber, string address, string birthday, string relation, string email) // Number 클래스의 생성자

{

this.name = name;

this.phoneNumber = phoneNumber;

this.birthday = birthday;

this.relation = relation;

this.email = email;

this.address = address;

}

public string Name //name의 프로퍼티

{

get

{

return name;

}

set

{

name = value;

}

}

public string PhoneNumber //phonenumber의 프로퍼티

{

get

{

return phoneNumber;

}

set

{

phoneNumber = value;

}

}

public string Birthday //birthday의 프로퍼티

{

get

{

return birthday;

}

set

{

birthday = value;

}

}

public string Relation //relation의 프로퍼티

{

get

{

return relation;

}

set

{

relation = value;

}

}

public string Email //email의 프로퍼티

{

get

{

return email;

}

set

{

email = value;

}

}

public string Address //address의 프로퍼티

{

get

{

return address;

}

set

{

address = value;

}

}

}

}

NumberArray클래스의 코드입니다.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace project

{

[Serializable]

class NumberArray

{

private Number[] number = new Number[100]; // Number클래스 100개짜리 배열

private int count = 0; // 카운트 0

public Number this[int index]

{

get

{

return number[index];

}

set

{

number[index] = value;

}

}

public int Count

{

get

{

return count;

}

set

{

count = value;

}

}

}

}