

|  |
| --- |
| 윈도우 프로그래밍 프로젝트 과제 |
| 콩 키우기게임 |

****

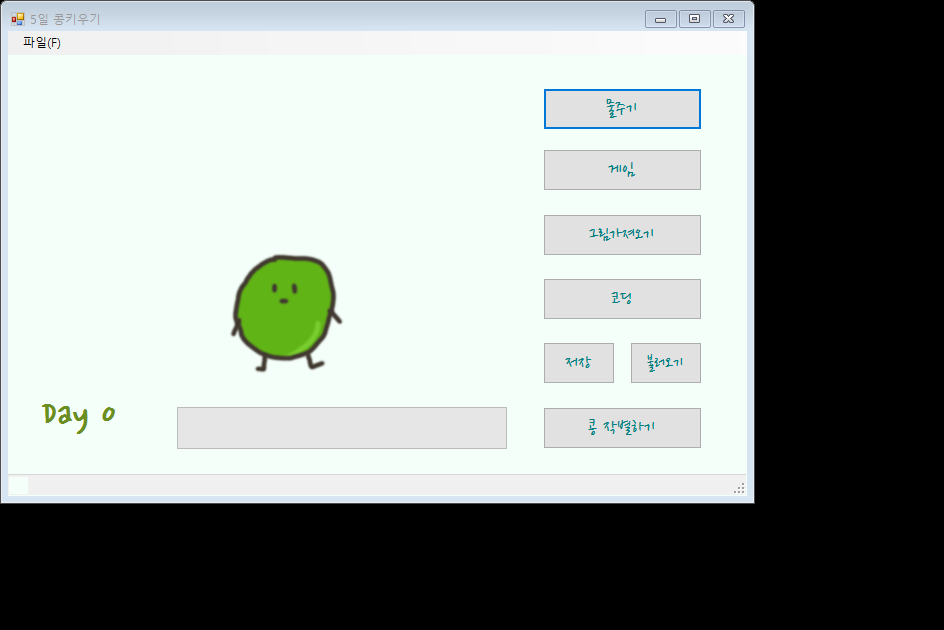


|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 제출일 | 2016. 12. 10 |  | 전공 | 화학과 |
| 과목 | 윈도우프로그래밍 |  | 학번 | 20143634 |
| 담당교수 | 김동회 |  | 이름 | 황지은 |

콩 키우기게임이란?

평소 프린세스메이커와 같은 육성시물레이션 게임을 좋아하여 사용자가 선택지를 어떻게 사용하느냐에 따라 콩의 엔딩이 다른 형식의 게임을 만들었다. 선택지 1개당 하루가 지나가고 5일을 키우면 엔딩을 보는 형식이다. 선택지로는 물주기, 게임, 그림가져오기, 코딩이 있다. 물주기는 단순 메시지박스가 뜨면서 물을 먹었다는 안내창이 뜬다. 게임은 간단한 경주 게임인데 맨 처음 라디오버튼을 이용하여 누가 이길 것인지 선택하고, 경기 후 결과를 메시지박스로 알려준다. 그림 가져오기는 원래 사용자가 그림을 그리고 저장하는 그림판 Form을 만들 예정이었지만 기한 내 구현하지 못하여 단순 그림을 가져오고 저장하는 Form으로 대체했다. 코딩은 사용자가 적은 문구를 MainForm의 StatusBar에 나타낸다. 저장, 불러오기는 변수값을 txt파일 형식으로 저장하여 불러오고 저장하는 형식이다. 5일동안 콩을 키우면 엔딩을 볼 수 있다.

엔딩은 총 5가지가 있다. 물을 4번이상 줄 경우 통통콩, 코딩을 3번이상 할 경우 공돌이콩, 게임을 3번이상 할 경우 프로게이머콩, 배경바꾸기를 3번 이상 할 경우 예술가콩, 이 중 어느 조건에도 해당되지 않으면 평범콩이 된다. 새로키우기를 누를 경우 값이 초기화 된 메인폼으로 돌아간다. 사용된 그림은 직접 그렸고, 그림판폼을 완성하지 못한 아쉬움이 남는다.

1. MainFormBeanFrom

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;

namespace BringUpBean

{

public partial class MainFormBean : Form

{

//각 버튼의 클릭 값, 날짜를 세는 변수

int day = 0;

int waterCount = 0, gameCount = 0, drawCount = 0, codingCount = 0;

//파일 저장, 불러오기에 사용되는 객체 생성

OpenFileDialog openFileDialog1 = new OpenFileDialog();

SaveFileDialog saveFileDialog1 = new SaveFileDialog();

//배경가져오기Form에 사용될 변수

Bitmap bmp;

public MainFormBean()

{

InitializeComponent();

this.day = 0;

this.waterCount = 0;

this.gameCount = 0;

this.drawCount = 0;

this.codingCount = 0;

}

//Paint에서 그린 이미지 저장하여 PictureBox에 나타나게함

//각 버튼은 5일 전 후로 안내창이 다르다.

//콩에게 물을준다(메세지창)

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

{

if( day == 5)

{

MessageBox.Show("콩은 다 자랐습니다! 콩과 작별해주세요","안녕콩",MessageBoxButtons.OK);

}else

{

MessageBox.Show("콩은 물을 맛있게 마셨다","콩에게 물주기",MessageBoxButtons.OK);

waterCount++;

day++;

lblDay.Text = "Day "+day;

ProBarDay.Value = day;

}

}

//과일경주 폼을 실행시킨다

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

{

if (day == 5)

{

MessageBox.Show("콩은 다 자랐습니다! 콩과 작별해주세요", "안녕콩", MessageBoxButtons.OK);

}

else

{

gameCount++;

day++;

lblDay.Text = "Day " + day;

ProBarDay.Value = day;

GameForm obj = new GameForm();

obj.ShowDialog();

//경기 결과를 StatusBar에 표시한다

toolbl.Text = obj.Msg;

}

}

//그림판 폼을 실행시킨다(X)

//배경을 성정하는 폼을 실행시킨다.

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

{

if (day == 5)

{

MessageBox.Show("콩은 다 자랐습니다! 콩과 작별해주세요", "안녕콩", MessageBoxButtons.OK);

}

else

{

drawCount++;

day++;

lblDay.Text = "Day " + day;

ProBarDay.Value = day;

PaintNewForm obj = new PaintNewForm();

obj.ShowDialog();

bmp = new Bitmap(obj.ImgName);

picBoxBorrow.Image = bmp;

}

}

//코딩 : 메세지창에서 입력받은 말을 Status에 뜨게 한다

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

{

if (day == 5)

{

MessageBox.Show("콩은 다 자랐습니다! 콩과 작별해주세요", "안녕콩", MessageBoxButtons.OK);

}

else

{

codingCount++;

day++;

lblDay.Text = "Day " + day;

ProBarDay.Value = day;

CodeInMainForm obj = new CodeInMainForm();

obj.ShowDialog();

toolbl.Text = obj.Sendmessage;

}

}

//새로키우기 버튼을 눌렀을 경우 모든 값 초기화

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

{

day = 0;

waterCount = 0;

gameCount = 0;

drawCount = 0;

codingCount = 0;

//시각적인 정보를 바꿔준다

ProBarDay.Value = waterCount;

lblDay.Text = "Day " + day;

MessageBox.Show("정보가 초기화 되었습니다", "새로키우기", MessageBoxButtons.OK, MessageBoxIcon.Asterisk);

}

//ToolBox에 대한 명령은 MenuBox에서 지정한 명령을 사용한다

//불러오기

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

{

btnOpen\_Click(sender, e);

}

//저장

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

{

btnSave\_Click(sender, e);

}

//끝

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

{

Close();

}

//txt파일 형식으로 변수들 정보를 저장한다

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

{

saveFileDialog1.InitialDirectory = @"C:\";

saveFileDialog1.Filter = "텍스트파일(\*.txt)|\*.txt|모든파일(\*.\*)|\*.\*";

saveFileDialog1.FilterIndex = 1;

saveFileDialog1.RestoreDirectory = true;

if (saveFileDialog1.ShowDialog() == DialogResult.OK)

{

MySave(saveFileDialog1.FileName);

}

}

//필드값 정보를 txt형식으로 저장

private void MySave(string s)

{

string str = day + "/" + waterCount + "/" + gameCount + "/" + drawCount + "/" + codingCount;

StreamWriter sw = File.CreateText(s); //파일의 스트림은 연다

sw.WriteLine(str);//txtBox의 내용을 파일로 기록

sw.Close(); //열었던 스트림을 닫는다

}

//txt파일 형식으로 저장된 내용을 풀어 본문에 적용한다

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

{

//open창을 꾸민다

openFileDialog1.InitialDirectory = @"C:\";

openFileDialog1.Filter = "텍스트파일(\*.txt)|\*.txt|모든파일(\*.\*)|\*.\*";

openFileDialog1.FilterIndex = 1;

openFileDialog1.RestoreDirectory = true;

openFileDialog1.Multiselect = true;

if (openFileDialog1.ShowDialog() == DialogResult.OK)

{

MyOpen(openFileDialog1.FileName);

}

}

//txt파일에 저장된 형식을 MainForm필드에 적용

private void MyOpen(string s)

{

string digit = "";

StreamReader sr = File.OpenText(s); //파일의스트림을 연다

digit = sr.ReadToEnd(); //파일의 내용을 txtBox에 출력

string[] str = digit.Split('/');

//값을 int형으로 바꾸고 초기화

this.day = Convert.ToInt32(str[0]);

this.waterCount = Convert.ToInt32(str[1]);

this.gameCount = Convert.ToInt32(str[2]);

this.drawCount = Convert.ToInt32(str[3]);

this.codingCount = Convert.ToInt32(str[4]);

sr.Close();

//시각적인 정보를 바꿔준다

ProBarDay.Value = waterCount;

lblDay.Text = "Day " + day;

}

//엔딩(6일차)전엔 아직 엔딩을 볼 수 없음 메세지창, 엔딩Form

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

{

//5일 전 엔딩을 볼 수 없다

if( day == 5)

{

//값을 전달하는 Contain클래스를 만들어 전달(질문 전 방식)

Contain c = new Contain(waterCount, gameCount, drawCount, codingCount);

EndForm end = new EndForm(c);

end.ShowDialog();

//종료한 EndForm에서 값을 전달받음

day = end.Day;

waterCount = end.WaterCount;

gameCount = end.GameCount;

drawCount = end.DrawCount;

codingCount = end.CodingCount;

ProBarDay.Value = waterCount;

lblDay.Text = "Day " + day;

}

else

{

MessageBox.Show("아직 콩은 자라고 있습니다. \nDay5가 되면 엔딩을 볼 수 있습니다.","자라는콩",MessageBoxButtons.OK);

}

}

}//mainform end

//subForm에 값을 전달하기 위한 클래스(질문 전 방법)

public class Contain

{

int waterCount = 0, gameCount = 0, drawCount = 0, codingCount = 0;

public Contain() { }

public Contain(int waterCount, int gameCount, int drawCount, int codingCount)

{

this.waterCount = waterCount;

this.gameCount = gameCount;

this.drawCount = drawCount;

this.codingCount = codingCount;

}

public int WaterCount

{

get { return waterCount; }

}

public int GameCount

{

get { return gameCount; }

}

public int DrawCount

{

get { return drawCount; }

}

public int CodingCount

{

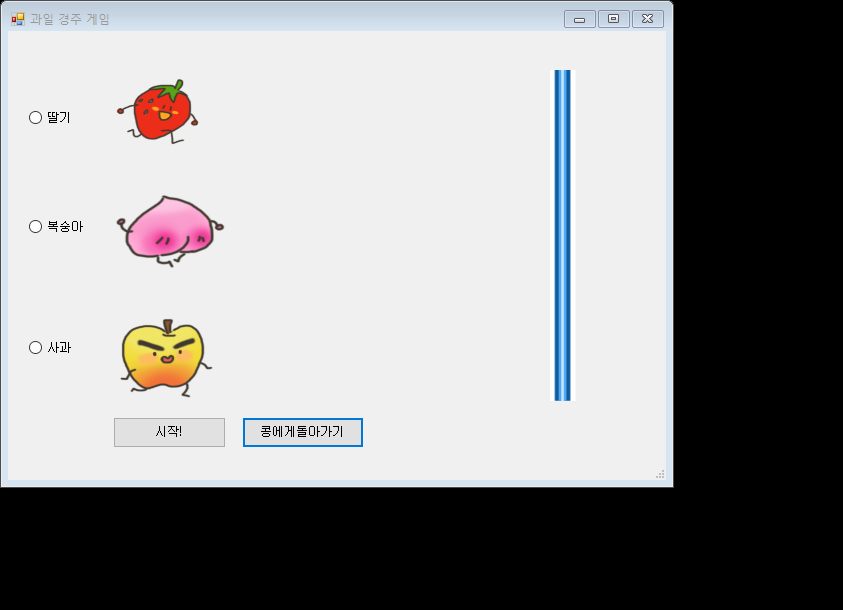
get { return codingCount; }

}

}

}

1. GameForm



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.Threading;

namespace BringUpBean

{

public partial class GameForm : Form

{

int btncount = 0;

private Fruit[] fruits;

private Random randG;

//SimpleThread tobj = new SimpleThread();

string msg = null;

public string Msg

{

get { return msg; }

}

public GameForm()

{

InitializeComponent();

}

//스레드사용 경주게임 시작

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

{

btncount++;

//ThreadStart ts = new ThreadStart(tobj.ThreadBody);

//Thread t = new Thread(ts);

//다음날이 되어도 같을 결과가 되지 않도록 메소드 안에서 객체 생성

fruits = new Fruit[3];

randG = new Random();

// 시작 위치는 28, 개의 길이는 75, 트랙의 길이는 450(PictureBox의 Length가 아니라 직접 잼)

fruits[0] = new Fruit() { FruitImg = picBerry, StartPoint = 200, LengthTrack = 300, Randomizer = randG };

fruits[1] = new Fruit() { FruitImg = picPeach, StartPoint = 200, LengthTrack = 300, Randomizer = randG };

fruits[2] = new Fruit() { FruitImg = picApple, StartPoint = 200, LengthTrack = 300, Randomizer = randG };

bool win = false;

int winFruit = -1;

//아무것도 선택하지 않으면 안내창이 뜬다.

if (raBtnApple.Checked == false && raBtnBerry.Checked == false && raBtnPeach.Checked == false)

{

MessageBox.Show("과일을 선택해 주세요", "안내", MessageBoxButtons.OK);

}

else

{//선택 시 경기시작

while (win != true)

{

for (int i = 0; i < 3 && win == false; i++) //승자가 아직 안날 경우 계속 반복한다

{

win = fruits[i].StartRun();

if (win == true)

{

winFruit = i;

}

}

//t.Start();

//Thread.Sleep(10);

System.Threading.Thread.Sleep(10);

}

//승자 == radioBtn : 이겼다!

if (winFruit == 0 && raBtnBerry.Checked == true)

{

msg = "축하합니다! 딸기가 이겼습니다!";

//MessageBox.Show(msg, "Win", MessageBoxButtons.OK);

}

else if (winFruit == 1 && raBtnPeach.Checked == true)

{

msg = "축하합니다! 복숭아가 이겼습니다!";

//MessageBox.Show(mag, "Win", MessageBoxButtons.OK);

}

else if (winFruit == 2 && raBtnApple.Checked == true)

{

msg = "축하합니다! 사과가 이겼습니다!";

//MessageBox.Show("축하합니다! 사과가 이겼습니다!", "Win", MessageBoxButtons.OK);

}

else

{

msg = "아쉽네요 다음날에 다시 도전해보세요!";

// MessageBox.Show("아쉽네요 다음날에 다시 도전해보세요!", "Lose", MessageBoxButtons.OK);

}

MessageBox.Show(msg, "Win", MessageBoxButtons.OK);

}

}

//폼을 종료하고 경기결과를 이전폼에 전달한다

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

{

//아무것도 선택하지 않으면 안내창

if(raBtnApple.Checked == false && raBtnBerry.Checked == false && raBtnPeach.Checked == false)

{

MessageBox.Show("과일을 선택해 주세요","안내",MessageBoxButtons.OK);

}

//경기를 시작하지 않았을 때 안내창

else if (btncount == 0)

{

MessageBox.Show("경기를 시작해 주세요", "안내", MessageBoxButtons.OK);

}

else

{

Close();

}

}

private void GameForm\_MouseMove(object sender, MouseEventArgs e)

{

//좌표확인 끝

//label1.Text = e.X + ", " + e.Y;

}

}//GameForm end

class Fruit

{

//시작점, 경주트랙길이, img객체의 트랙위에서의 위치

private int startPoint = 0, lengthTrack = 0, imgPosition = 0;

private PictureBox fruitImg = null; //이미지객체

public Random Randomizer = null; //랜덤함수

public bool StartRun()

{

int digitRand; //랜덤함수 저장위치

digitRand = Randomizer.Next(10) + 1; //1~10칸 랜덤 이동

//img객체의 x좌표에 랜덤값을 넣는다

Point p = fruitImg.Location; //좌표에 현재 img객체의 위치값 넣는다

p.X += digitRand; //랜덤값을 X좌표에 넣어 이동시긴다

fruitImg.Location = p; //바뀐 좌표값을 다시 객체의 좌표값에 넣어준다

imgPosition = p.X;

if(imgPosition >= (startPoint + lengthTrack))

{

return true; //도착지점에 도착해 경기가 끝남

}

else

{

return false; //아직 도착하지 않았으면 false로 반환

}

}

public int StartPoint

{

get { return startPoint; }

set { this.startPoint = value; }

}

public int LengthTrack

{

get { return lengthTrack; }

set { this.lengthTrack = value; }

}

public int ImgPosition

{

get { return imgPosition; }

set { this.imgPosition = value; }

}

public PictureBox FruitImg

{

get { return fruitImg; }

set { this.fruitImg = value; }

}

}//FruitClass end

/\*

class SimpleThread

{

public void ThreadBody()

{

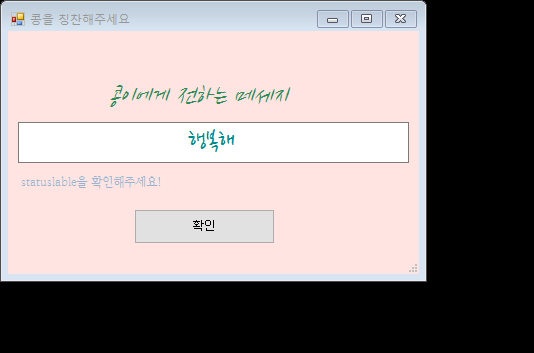
Thread.Sleep(10);

}

}\*/

}

1. CodeInMainForm



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;

namespace BringUpBean

{

public partial class CodeInMainForm : Form

{

string sendmessage = "";

public CodeInMainForm()

{

InitializeComponent();

}

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

{

//메세지를 변수에 저장하여 이 폼의 종료지점으로 가서 MainForm에 적용

sendmessage = txtSendBean.Text;

Close();

}

public string Sendmessage

{

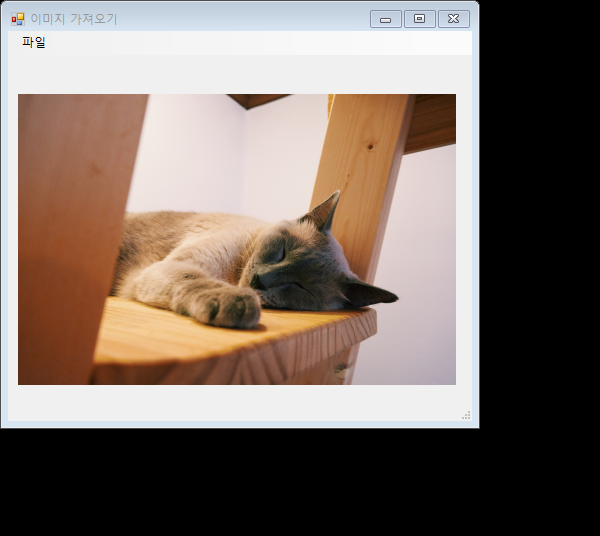
get { return sendmessage; }

}

}

}

1. PaintNewForm



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;

namespace BringUpBean

{

public partial class PaintNewForm : Form

{

SaveFileDialog save;

OpenFileDialog open;

//파일 경로를 전달하기위한 변수

private string imgName = "";

Image img;

Bitmap bmp;

public string ImgName

{

get { return imgName; }

}

public PaintNewForm()

{

InitializeComponent();

save = new SaveFileDialog();

open = new OpenFileDialog();

}

//이미지 저장

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

{

save.Filter = "이미지파일(\*.jpg)|\*.jpg|이미지파일(\*.png)|\*.png|비트맵(\*.bmp)|\*.bmp";

if(save.ShowDialog() == DialogResult.OK)

{

imgName = save.FileName;

//파일 경로로 비트맵 저장

bmp.Save(imgName);

}

}

//이미지 불러오기

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

{

//처음에 C드라이브로

open.InitialDirectory = @"C:\";

open.Filter = "이미지파일(\*.jpg)|\*.jpg|이미지파일(\*.png)|\*.png|모든파일(\*.\*)|\*.\*";

if(open.ShowDialog() == DialogResult.OK)

{

//경로저장

imgName = open.FileName;

//img에 저장 후 bmp에 저장

img = Image.FromFile(imgName);

picBox.Image = img;

//불러온 이미지 정보로 비트맵 생성

bmp = new Bitmap(img);

}

}

//종료

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

{

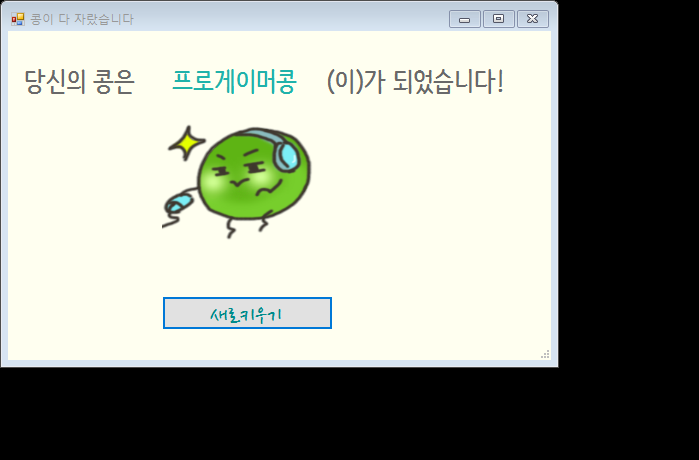
Close();

}

}

}

1. EndForm



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;

namespace BringUpBean

{

public partial class EndForm : Form

{

//전달받은 값을 저장하기 위한 변수

int waterCount = 0, gameCount = 0, drawCount = 0, codingCount = 0;

int day = 0;

string result = "result";

public EndForm()

{

InitializeComponent();

}

public EndForm(Contain c)

{

InitializeComponent();

//생성될 때 저장받은 값으로 result에 저장

this.waterCount = c.WaterCount;

this.gameCount = c.GameCount;

this.drawCount = c.DrawCount;

this.codingCount = c.CodingCount;

if (waterCount >= 4) //4번이상 물을 준 경우

{

result = "Big";

}else if (gameCount >= 3) //3번이상 게임을 한 경우

{

result = "Progamer";

}else if (codingCount >= 3) //3번이상 코딩을 한 경우

{

result = "Programmer";

}else if(drawCount >= 3) //3번이상 배경가져오기(그리기)를 한 경우

{

result = "Painter";

}else //아무것도 아닐 때

{

result = "Plain";

}

}

//새로시작하기 버튼을 누를 경우

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

{

//초기화시킨 값을 MainForm에 전달

MainFormBean obj = new MainFormBean();

waterCount = 0; gameCount = 0; drawCount = 0; codingCount = 0;

Close();

}

//엔딩에따라 다른 이미지를 나타내기

private void EndForm\_Load(object sender, EventArgs e)

{

if (result == "Plain")

{

lblFinalBean.Text = "행복한 평범콩";

picBoxEnd.Image = imgListBeanEnding.Images[0];

}else if (result == "Big")

{

lblFinalBean.Text = "통통콩";

picBoxEnd.Image = imgListBeanEnding.Images[1];

}

else if (result == "Progamer")

{

lblFinalBean.Text = "프로게이머콩";

picBoxEnd.Image = imgListBeanEnding.Images[3];

}

else if (result == "Programmer")

{

lblFinalBean.Text = "공돌이콩";

picBoxEnd.Image = imgListBeanEnding.Images[4];

}

else if (result == "Painter")

{

lblFinalBean.Text = "예술가콩";

picBoxEnd.Image = imgListBeanEnding.Images[2];

}

}

//waterCount = 0, gameCount = 0, drawCount = 0, codingCount = 0;

public int WaterCount

{

get { return waterCount; }

}

public int GameCount

{

get { return gameCount; }

}

public int DrawCount

{

get { return drawCount; }

}

public int CodingCount

{

get { return codingCount; }

}

public int Day

{

get { return day; }

}

}

}