



Kang ByungSu

학급 성적 계산 및 조회 프로그램



## 목차

- 1. 기획 의도
- 2. 프로젝트 순차표
- 3. 코드 입력
- 4. 알고리즘
- 5. 실행결과
- 6. 오류 수정
- 7. 보완 및 고찰



# 1. 기획 의도

- 중, 고등학교 교사들이 편하게 사용할 수 있는 프로그램으로 제작 결정
- 마우스 클릭같은 간단한 조작으로 점수를 저장하면 자동으로 평균을 산출해내도록 만들어 줌
- 상세 가이드
  - 각 학생들의 번호, 이름, 과목 별 점수(국어, 수학, 영어, 한국사) 입력
  - 점수 입력 후 저장한 뒤 조회 버튼을 클릭하면 여태까지 저장된 학생들의 점수들이 리스트 형태로 공개
  - 평균은 리스트에서 자동적으로 계산하여 출력
  - 필요시 학생들의 정보를 삭제, 점수를 수정 또는 데이터를 엑셀 파일로 추출 가능



## 2. 프로젝트 순차표

	2021-04-08	2021-04-09	2021-04-10	2021-04-11	2021-04-12	2021-04-13
프로젝트 구상						
코드 작성						
알고리즘 작성						
PPT 제작						
오류 수정						



### 3. 코드 입력

```
double average = 0;

Console.Write("학생의 수를 입력하십시오: ");
int stu_number = int.Parse(Console.ReadLine());

try
{
    for (int i = 0; i < stu_number; i++)
    {
        Console.Write("{0}번 학생의 이름: ", (i + 1));
        string name = Console.ReadLine();
        Console.Write("{0}번 학생의 국어 점수: ", (i + 1));
        int kor = int.Parse(Console.ReadLine());
        if (kor > 100 || kor < 0)
        {
            Console.WriteLine("숫자를 잘못 입력하였습니다. 다시 입력하세요");
            return;
        }
        Console.Write("{0}번 학생의 수학 점수: ", (i + 1));
        int math = int.Parse(Console.ReadLine());
        Console.Write("{0}번 학생의 영어 점수: ", (i + 1));
        int eng = int.Parse(Console.ReadLine());
        Console.Write("{0}번 학생의 한국사 점수: ", (i + 1));
        int history = int.Parse(Console.ReadLine());
        average = (kor + math + eng + history) / 4.0;
        Student scorelist = new Student(stu_number, name, kor, math, eng, history);
        list.Add(scorelist);
    }
}
```

본격적으로 WinForm으로 제작  
하기 전, 콘솔 창에서 학생들의 점수가  
정상적으로 출력이 가능한지 확인하기  
위해 간단하게 제작한 코드.

List와 Student 객체를 이용하여  
먼저 학생의 수를 입력 후 각 학생의  
점수를 입력하면 과목 변수에 저장,  
출력을 하는 형태로 제작



### 3. 코드 입력

```
private void AddScore_Click(object sender, EventArgs e)
{
    try {
        if (Stu_Num.Text == "" || Stu_Name.Text == "" || Stu_Kor.Text == ""
            || Stu_Math.Text == "" || Stu_Eng.Text == "" || Stu_Histo.Text == "")
        {
            MessageBox.Show("정보가 누락되었습니다.", "오류", MessageBoxButtons.OK, MessageBoxIcon.Error);
            return;
        }
        if (Convert.ToInt32(Stu_Kor.Text) > 100 || Convert.ToInt32(Stu_Kor.Text) < 0 ||
            Convert.ToInt32(Stu_Math.Text) > 100 || Convert.ToInt32(Stu_Math.Text) < 0 ||
            Convert.ToInt32(Stu_Eng.Text) > 100 || Convert.ToInt32(Stu_Eng.Text) < 0 ||
            Convert.ToInt32(Stu_Histo.Text) > 100 || Convert.ToInt32(Stu_Histo.Text) < 0)
        {
            MessageBox.Show("부적절한 점수가 존재합니다.", "오류", MessageBoxButtons.OK, MessageBoxIcon.Error);
            return;
        }
        Student student = new Student(Convert.ToInt32(Stu_Num.Text), Stu_Name.Text, Convert.ToInt32(Stu_Kor.Text),
                                       Convert.ToInt32(Stu_Math.Text), Convert.ToInt32(Stu_Eng.Text), Convert.ToInt32(Stu_Histo.Text));
        Model.DataManager.students.Add(student);
        MessageBox.Show("학생의 점수 정보가 저장되었습니다.");
        Close();
    }
    catch (Exception f)
    {
        MessageBox.Show("숫자가 아닌 문자를 입력하였습니다.", "오류", MessageBoxButtons.OK, MessageBoxIcon.Error);
        printLog(f.Message);
    }
}
```

점수를 정상적으로 출력이 가능한 것을  
확인, 본격적으로 WinForm 형태로  
코드 작성.

먼저 학생들의 점수 정보를 추가하는  
코드 작성. 각 TextBox마다 빈칸일  
경우, 점수가 비정상적일 경우, 문자를  
입력한 경우를 상정하여 각각의 오류가  
뜨도록 만듦.



### 3. 코드 입력

```
// ListView 초기화
private void initScoreView()
{
    //List<Student> list = new List<Student>();
    if (Model.DataManager.students.Count == 0)
    {
        MessageBox.Show("데이터가 존재하지 않습니다.", "오류", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
    }
    else
    {
        for (int i = 0; i < Model.DataManager.students.Count; i++)
        {
            double average = (Model.DataManager.students[i].Kor + Model.DataManager.students[i].Math + Model.DataManager.students[i].Eng) / 3;
            StuScore.Items.Add(new ListViewItem(
                new string[]
                {
                    Model.DataManager.students[i].Stu_num.ToString(), Model.DataManager.students[i].Stu_name.ToString(), Model.DataManager.students[i].Kor.ToString(),
                    Model.DataManager.students[i].Math.ToString(), Model.DataManager.students[i].Eng.ToString(), Model.DataManager.students[i].average.ToString()
                }
            ));
        }
        setRowColor(StuScore, Color.White, Color.WhiteSmoke);
        int index = StuScore.Items.Count - 1;
        StuScore.Items[index].Selected = true; // 초기화때 ListView의 마지막 인덱스가 자동적으로 선택되게
        StuScore.Items[index].Focused = true; // 초기화때 ListView의 마지막 인덱스쪽으로 포커스
        StuScore.EnsureVisible(index);
    }
}
```

저장한 학생들의 점수 정보들을 조회  
여태까지 저장된 List에 학생들의  
정보들을 ListView 형태로 출력을  
시켜줌.

홀수, 짝수 줄마다 배경색을 다르게  
해주어 혼란을 줄여주며 조회 버튼을  
누르면 처음에 맨 마지막 학생의 정보  
가 포커스 되도록 맞춰줌.



### 3. 코드 입력

```
private void deleteOK_Click(object sender, EventArgs e)
{
    try
    { // 텍스트 박스에 입력한 학번이 리스트에 저장된 학번들과 동일한 경우가 있는지 확인
        for (int i = 0; i < Model.DataManager.students.Count; i++)
        {
            if (Model.DataManager.students[i].Stu_num == Convert.ToInt32(StudentNumber.Text))
            {
                if (MessageBox.Show("정말로 삭제하시겠습니까?", "경고", MessageBoxButtons.YesNo, MessageBoxIcon.Warning) == DialogResult.No)
                {
                    Model.DataManager.students.RemoveAt(i);
                    MessageBox.Show("입력한 번호와 일치하는 학생의 정보를 삭제하였습니다.");
                    Close();
                }
            }
            else
            {
                MessageBox.Show("입력한 번호와 일치하는 학생이 존재하지 않습니다.", "오류", MessageBoxButtons.OK, MessageBoxIcon.Error);
                Close();
            }
        }
    }
    catch (Exception f)
    {
        MessageBox.Show("숫자가 아닌 문자를 입력하였습니다.", "오류", MessageBoxButtons.OK, MessageBoxIcon.Error);
        printLog(f.Message);
        printLog(f.StackTrace);
    }
}
```

학생 정보를 삭제를 할 때 삭제할 학생의 번호를 입력하여 삭제하는 방법으로 코드를 작성.

일치하는 번호가 존재하면 먼저 주의창을 띄워 정말로 삭제할 것인지 재차 물어본 뒤 예 버튼을 누르면 삭제가 됨. 생성 때와 마찬가지로 TextBox에 문자열 등이 들어갈 경우 경고창을 띄워 다시 번호를 입력하게 만들어 둬.





### 3. 코드 입력

```
}  
  
// 리스트에 학생들의 점수 정보를 덮어쓰는 형태로 수정  
for (int i = 0; i < Model.DataManager.students.Count; i++)  
{  
    if (Stu_Name.Text == Model.DataManager.students[i].Stu_name)  
    {  
        Model.DataManager.students[i].Kor = Convert.ToInt32(Stu_Kor.Text);  
        Model.DataManager.students[i].Math = Convert.ToInt32(Stu_Math.Text);  
        Model.DataManager.students[i].Eng = Convert.ToInt32(Stu_Eng.Text);  
        Model.DataManager.students[i].History = Convert.ToInt32(Stu_Histo.Text);  
        MessageBox.Show("학생의 점수 정보가 수정되었습니다.");  
        break;  
    }  
    else  
    {  
        MessageBox.Show("수정할 해당 학생의 이름을 잘못 입력하였습니다.", "오류", MessageBoxButtons.OK, MessageBoxIcon.Error);  
        return;  
    }  
}  
Close();
```

학생의 점수 정보를 수정할 때에도  
일치하는 번호를 먼저 찾아준 뒤에  
기존의 점수에 새로운 점수를 덮어쓰우는  
형태로 점수 저장.



### 3. 코드 입력

```
// ListView 데이터들을 csv 확장자로 이루어진 엑셀 파일로 저장
private void button1_Click(object sender, EventArgs e)
{
    SaveFileDialog saveFile = new SaveFileDialog();
    saveFile.Filter = "csv File|*.csv";
    saveFile.Title = "Save";
    saveFile.ShowDialog();

    if (saveFile.FileName != "")
    {
        System.IO.FileStream fs = (System.IO.FileStream)saveFile.OpenFile();

        switch (saveFile.FilterIndex)
        {
            case 1:
                for (int i = 0; i < StuScore.Items.Count; i++)
                {
                    string sTmp = "";
                    for (int j = 0; j < StuScore.Items[i].SubItems.Count; j++)
                        sTmp += StuScore.Items[i].SubItems[j].Text + ", ";
                    sTmp += "\n";
                    Byte[] Bytes = Encoding.Default.GetBytes(sTmp); //인코딩 오류 있음 고쳐야 함 --> 인코딩을 Default로 변경하여 해결
                    fs.Write(Bytes, 0, Bytes.Length);
                }
                break;
        }
    }
}
```

학생의 점수 정보를 CSV 파일로  
데이터를 저장하는 코드.

학생 정보 조회 창에서 파일 저장 버튼  
을 누르면 저장할 폴더를 선택하는 창이  
나오며 저장하면 CSV 파일로 저장, 학생  
들의 정보를 엑셀에서 조회가 가능해짐



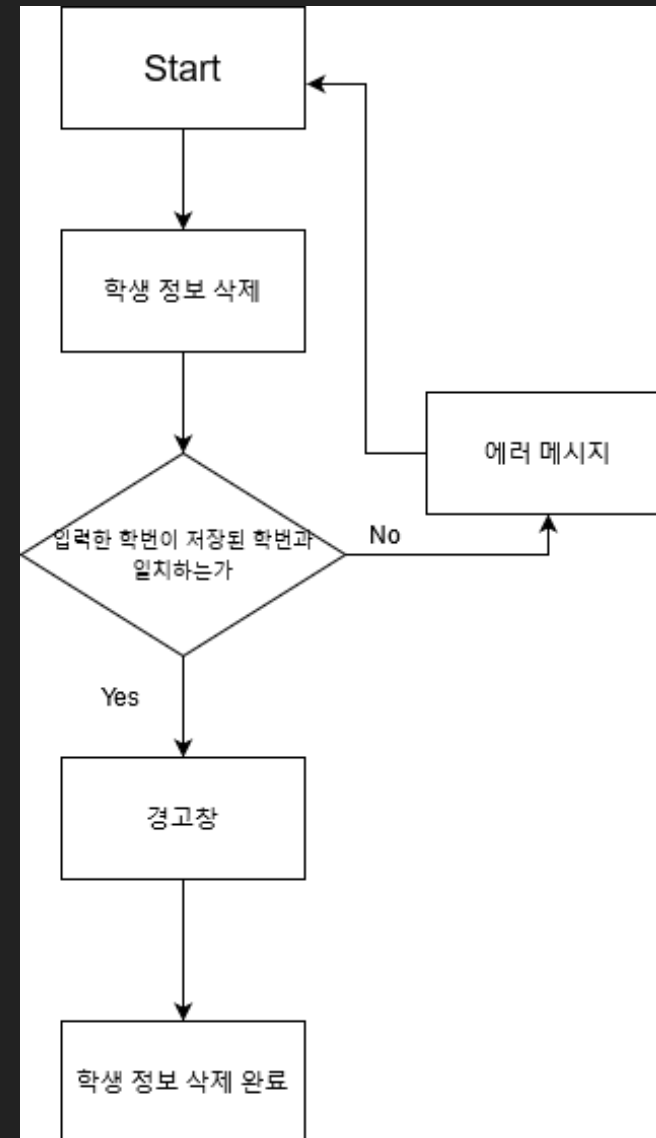
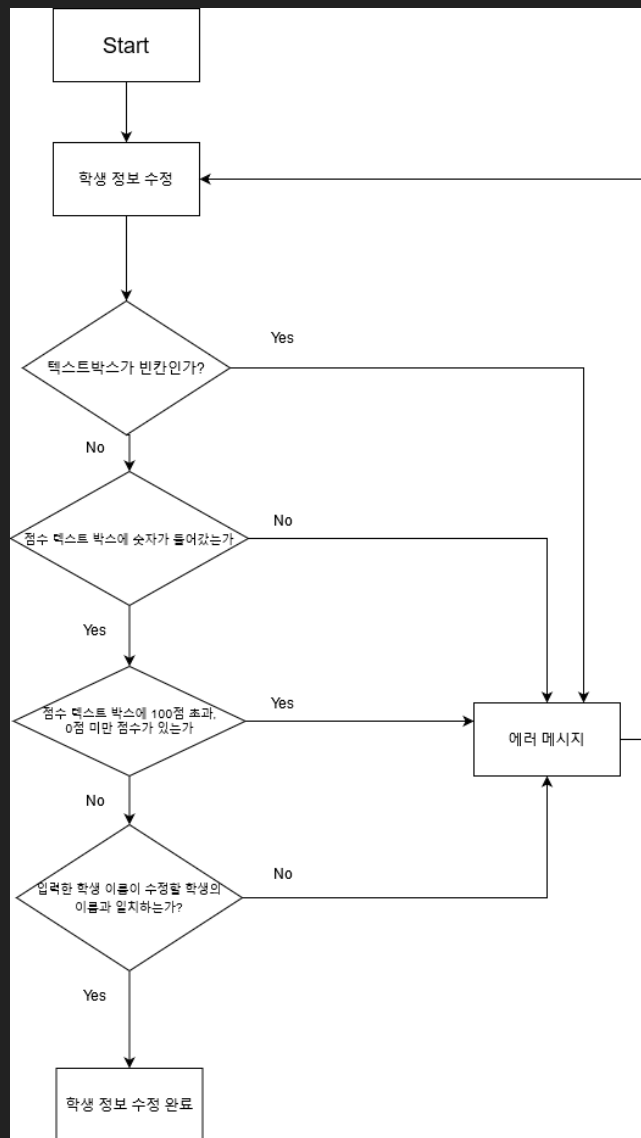
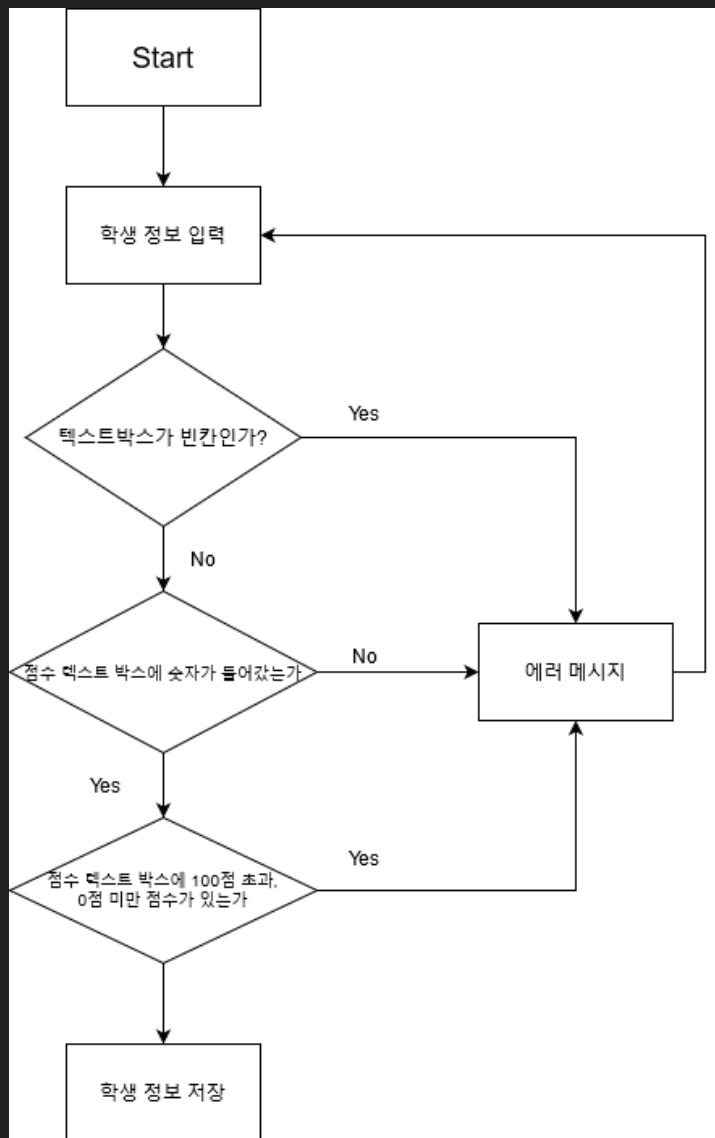
### 3. 코드 입력

```
private void printLog(string contents)
{
    DirectoryInfo di = new DirectoryInfo("Trace");
    if (di.Exists == false)
    {
        di.Create();
    }
    using (StreamWriter writer = new StreamWriter("Trace" + "\\\" + "Log.txt"))
    {
        writer.WriteLine
            ($"[{DateTime.Now.ToString("yyyy-MM-dd hh:mm:ss")}]{{contents}}{".txt"}");
    }
}
```

만약 프로그램 실행 도중에 오류가 발생  
하였을 때 응용 프로그램이 있는 자리에  
자동으로 로그를 txt 파일로 저장하게  
함으로써 어디에서 오류가 났는지 신속  
하게 확인을 하여 수정을 할 수 있게  
만들어 줌.

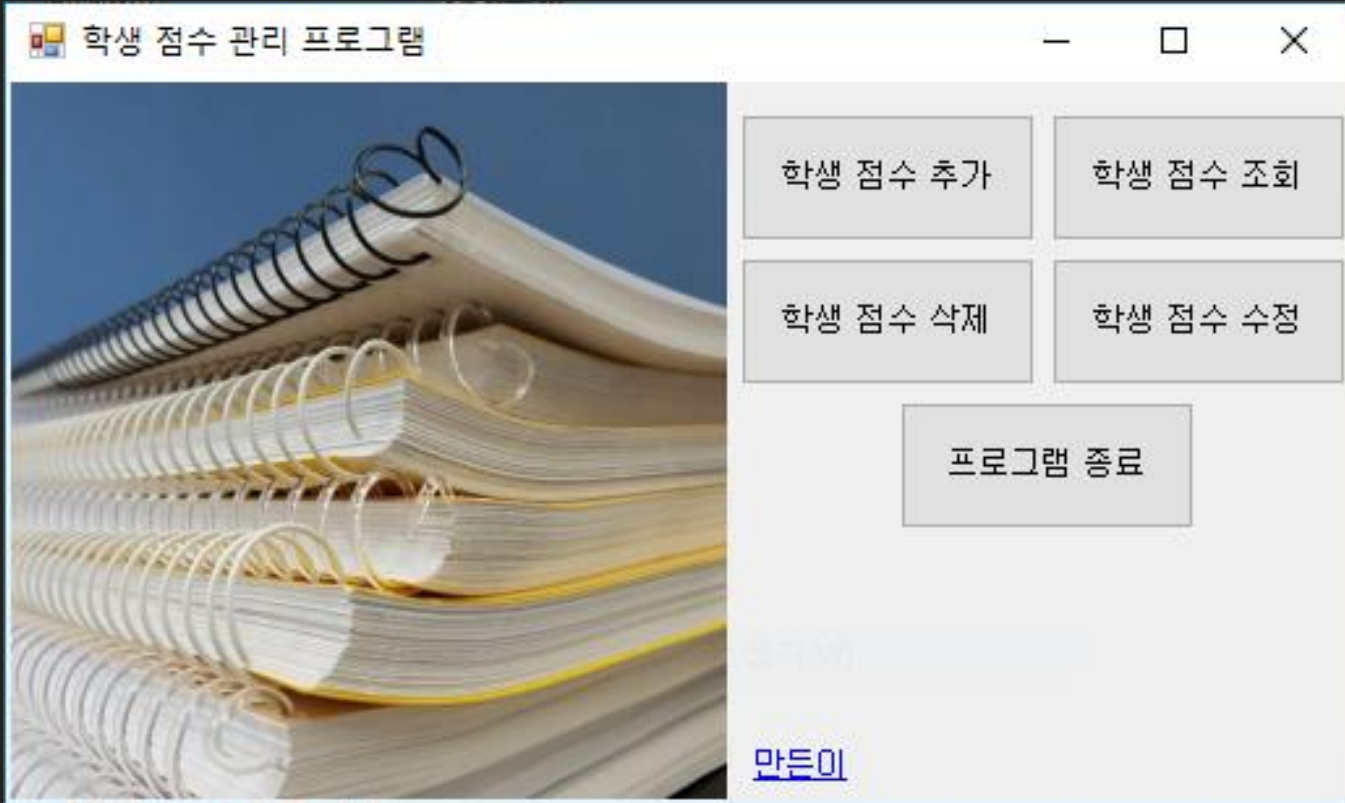


## 4. 알고리즘

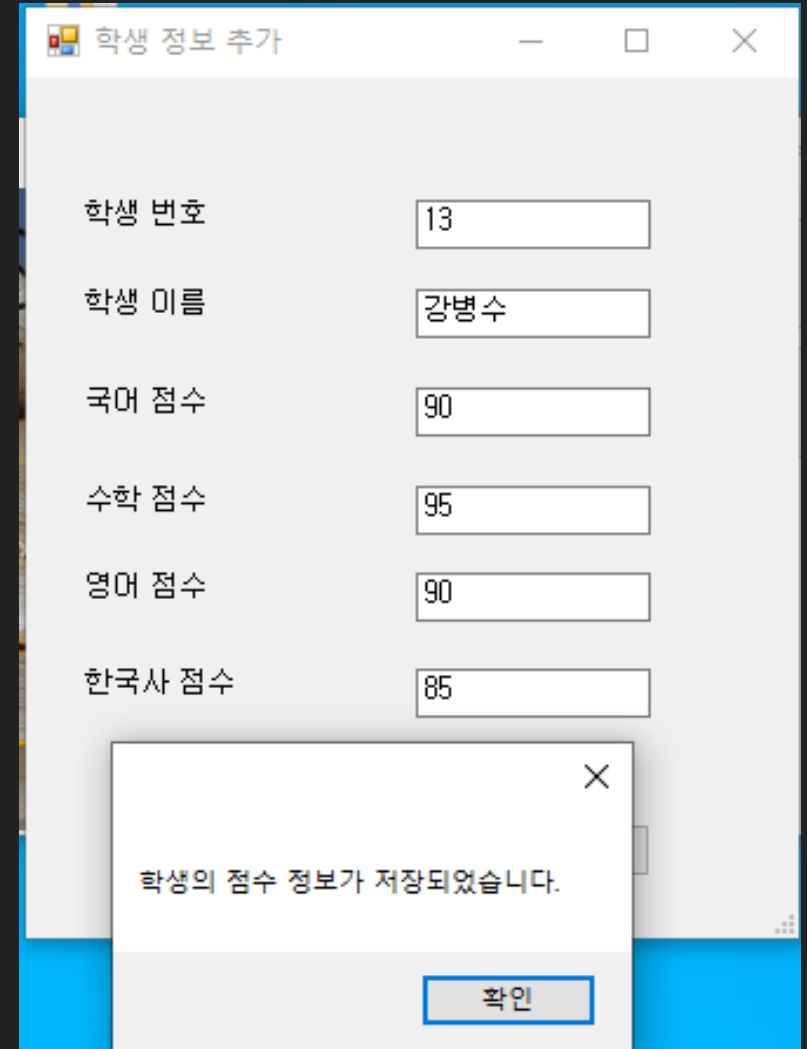




## 5. 실행 결과



<프로그램을 처음 실행하였을 때의 화면>



학생 정보 추가 창에서 알맞은 값을 입력하면 정보가 정상적으로 저장된다.



## 5. 실행 결과

The screenshot displays two overlapping windows from a program titled '학생 정보 수정' (Student Information Modification).

The background window prompts the user to enter the student number to be modified: '수정할 학생의 번호를 입력하십시오' (Enter the number of the student to be modified). A text box contains the number '13'.

The foreground window is a form for entering scores. It includes the following fields and labels:

- 국어 점수 (Korean Score)
- 수학 점수 (Math Score)
- 영어 점수 (English Score)
- 한국사 점수 (Korean History Score)
- A confirmation field: '확인을 위하여 수정할 해당 학생의 이름을 입력하십시오' (Enter the name of the student to be modified for confirmation).

At the bottom of the foreground window are two buttons: '저장' (Save) and '취소' (Cancel).

학생의 정보를 수정하고자 한다면 먼저 수정하려는 학생의 번호를 입력을 한다.

그 다음에 점수를 새로 입력을 한 뒤에 마지막 확인 절차로 수정하려는 학생의 이름을 적어준 뒤 일치한다면 수정 완료, 불일치하면 오류 창이 띄워진다.



## 5. 실행 결과

학생 정보


번호	이름	국어	수학	영어	한국사	평균
358413	강병수	90	95	90	85	90

데이터 저장      확인

학생 정보를 저장한 다음에 조회 버튼을 클릭하면 나오는 창.

List에 저장된 번호, 이름, 각 과목의 점수들이 출력되며 평균은 각 과목의 점수들을 계산하여 자동으로 출력하게 만들어 줌.

데이터 저장을 누르면 ListView에 저장되어 있는 데이터들을 CSV 파일로 저장이 됨.



이름이 한글일 경우

**학생의 이름을 영어로 입력을 할 때에는 문제 없이 CSV 파일로 저장되며 글자가  
출력이 되지만 학생의 이름을 한글로 입력하여 저장하면 글자가 깨지는 문제와  
일부 점수가 셀에서 이탈하는 문제가 발생함**





## 6. 오류 수정

```
{
    string sTmp = "";
    for (int j = 0; j < StuScore.Items[i].SubItems
        sTmp += StuScore.Items[i].SubItems[j].Text
    sTmp += "\n";
    Byte[] Bytes = Encoding.UTF8.GetBytes(sTmp); /
    fs.Write(Bytes, 0, Bytes.Length);
}
break;
```



```
{
    string sTmp = "";
    for (int j = 0; j < StuScore.Items[i].SubItems.
        sTmp += StuScore.Items[i].SubItems[j].Text
    sTmp += "\n";
    Byte[] Bytes = Encoding.Default.GetBytes(sTmp);
    fs.Write(Bytes, 0, Bytes.Length);
}
break;
```

이 문제는 한글을 인코딩하고 디코딩  
할 때 다른 규격을 사용해서 생기는  
문제였다.

엑셀에서 CSV 파일을 열 때 ANSI  
코드로 디코딩 하도록 되어있기 때문이다.

그렇기 때문에 저장할 때 UTF-8로  
인코딩을 하지 않고 기본값(Default)으로  
변경해주면 해결이 된다.



## 7. 보완점 및 고찰

- ListView에 저장된 데이터들을 CSV 파일로 저장되는데 반대로 CSV 파일을 불러와 ListView에 불러오는 방법이 필요함
- ListView에서 학생의 학번 번호(첫번째 Column)이 가운데 정렬이 안되는 문제가 있다. 이 부분은 Visual Studio에서 강제하는 것이라 GridView나 외부 라이브러리를 이용하는 등 다른 방법이 필요할 것 같다.
- 파일을 저장하고자 사용한 코드를 이해하고 짜는데 생각보다 시간이 오래 걸렸다. FileStream 에 대한 공부를 더 해야할 것 같다.



Kang ByungSu

THANK YOU FOR YOUR ATTENTION