INotifyChan ged 사용하기

2022-03-16

들어가기에 앞서

이번프로젝트에 대하여...

C#에 있는 INotifyProperty Changed를 사용해서 속성이 변경되었음을 클라이언 트에 알리는 기능을 활용해보고자 간단히 작성한 프로젝트 이다

목 차

01 Changed사용 02 학생 정보 등록 03 학생 정보 수정 04 학생 정보 삭제 05 INotifyProperty Changed 확인 06

INotifyPrope rtyChanged 사용하기

INotifyPropertyChanged 인터페이스

```
public class Student : INotifyPropertyChanged
{
    private Guid idValue = Guid.NewGuid();
    private string studentGradeValue = String.Empty;
    private string studentNoValue = String.Empty;
    private string studentNoValue = String.Empty;
    private string studentNameValue = String.Empty;
    private string studentScoreValue = String.Empty;

    public Student(string grade, string cclass, string no, string name, string score)
    {
        Grade = grade;
        Cclass = cclass;
        No = no;
        Name = name;
        Score = score;
    }

    public Student()
    {
     }
}
```

INotiFy Property Changed

INotifyPropertyChanged 이벤트

public event PropertyChangedEventHandler PropertyChanged;

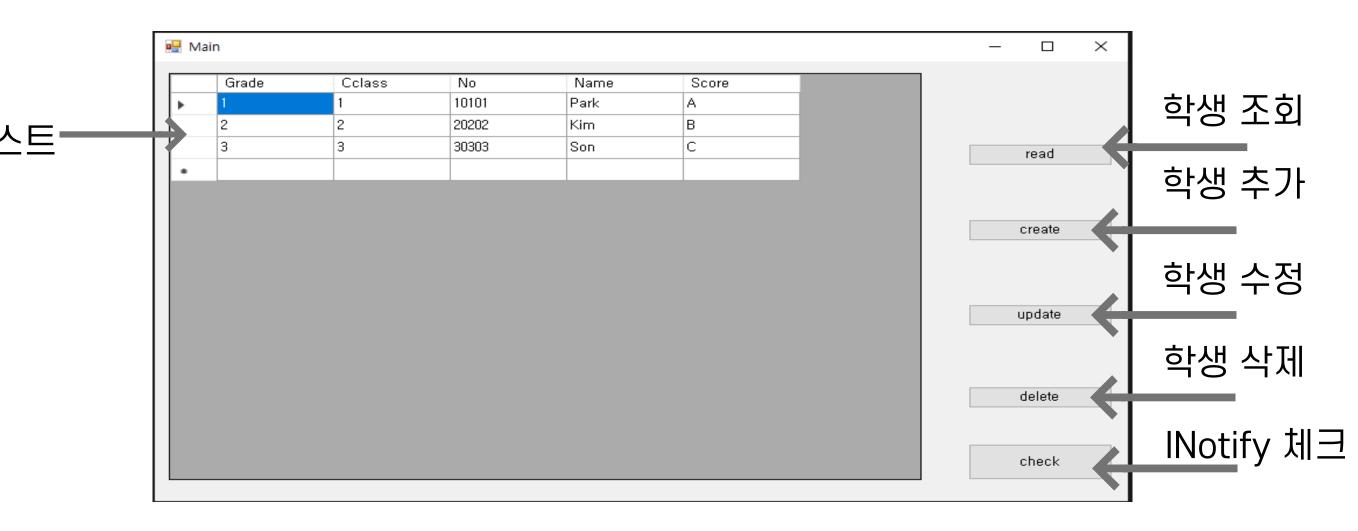
INotifyPropertyChanged 이벤트 발생시 작동됨

학생정보 등록

기본적인 화면구성.

로그인에 성공시 기본적으로 나오는 고객 멤버 관리 프로 그램입니다.

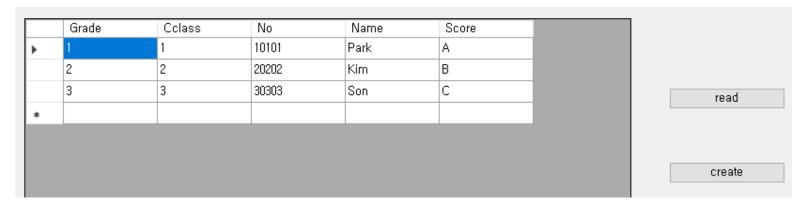
기본적으로 학생추가,학생수 정,학생조회,학생제거, 추가 **학생 리스트** 로 INorify 기능 동작 체크를 담았습니다.

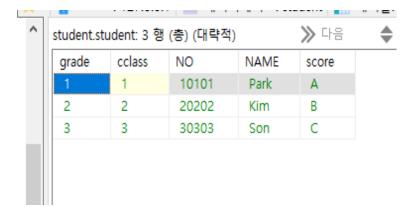


학생 등록

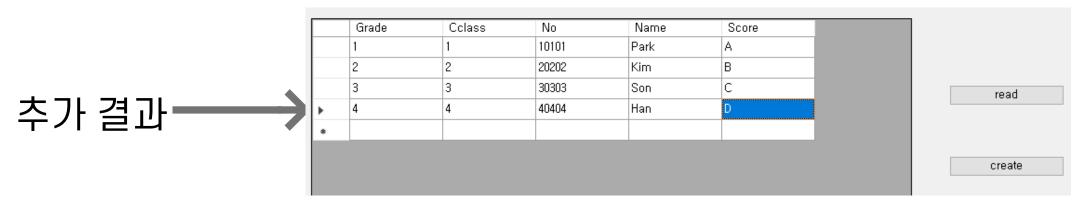
그리드뷰의 셀에 학생에 대한 정보를 입력하고 cre ate 버튼을 클릭시 DB에 학생정보가 삽입되며 해 당 정보는 BindingList에도 저장된다

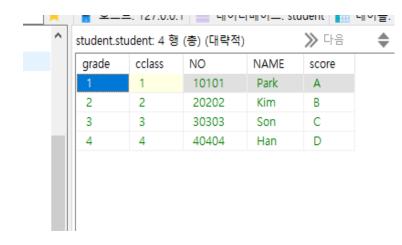
학생 추가 전





학생 추가 후





학생 등록 Code.

인스턴스로 바 인딩 소스 선언

private BindingSource StudentBindingSource = new BindingSource();

학생 클래스로 된 바인딩 리 스트를 만들고 데이터 소스를 바인딩 소스로 지정

BindingList<Student> studentList =
 this.StudentBindingSource.DataSource as BindingList<Student>;

테이블 명 및 DB정보

```
string Table = "Student";
const string MYDB = "Server=localhost; Port=3306; Database=Student; userid = root; Pwd = 0729;";
MySqlConnection MyConn = new MySqlConnection(MYDB);
```

DB 접속시도

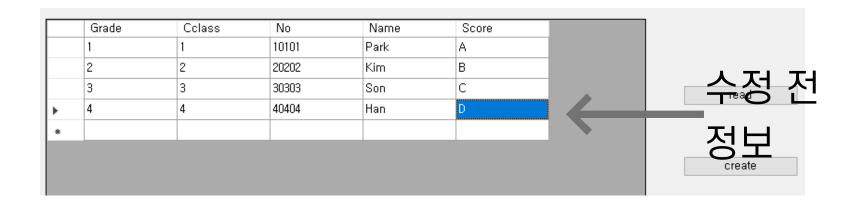
```
public void ConnectDB()
{
    try
    {
        MyConn.Open();
    }
    catch (Exception e)
    {
        Console.WriteLine("실패");
        Console.WriteLine(e.ToString());
        MyConn.Close();
}
```

선택된 셀의 위치값을 바인딩리 스트의 배열 위치 값으로 사용해 서 해당 배열 위치에 해당 셀의 내용 값을 전달한다.

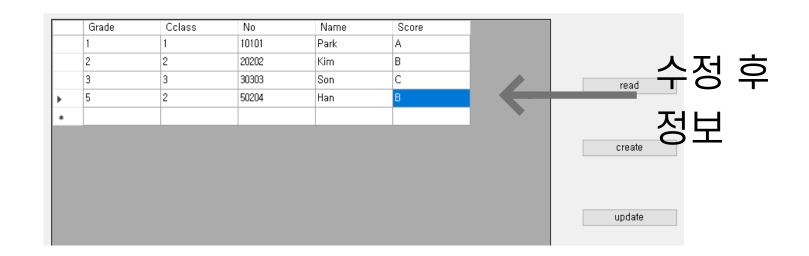
```
private void button2_Click(object sender, EventArgs e)
    BindingList<Student> studentList =
        this.StudentBindingSource.DataSource as BindingList<Student>;
    ConnectDB();
    if (this.dataGridView1.RowCount == 0)
        return;
    int rowIndex = dataGridView1.CurrentRow.Index;
    Student s = studentList[rowIndex];
    string G = s.Grade;
    string Cc = s.Cclass;
    string No = s.No;
    string Na = s.Name;
    string S = s.Score;
    string sql = $"insert into {Table} values ('{G}','{Cc}','{No}','{Na}','{S}')";
    MySqlCommand cmd = new MySqlCommand();
    cmd.Connection = MyConn;
    cmd.CommandText = sql;
    cmd.ExecuteNonQuery();
    MyConn.Close();
```

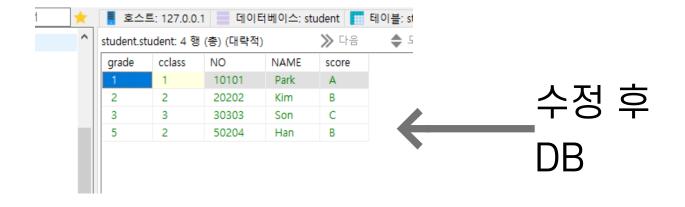

학생 수정

그리드뷰에 저장된 정보를 클릭후 원하는 값으로 바꾼뒤에 이후 원하는 정보를 수정한뒤 update 등을 클릭시 다음과 같이 수정되서 적용되는걸 볼수있다.









학생 수정 Code.

```
ivate void button3_Click(object sender, EventArgs e)
                                                                             바인딩 리스트를 가져옴
 BindingList<Student> studentList =
    this.StudentBindingSource.DataSource as BindingList<Student>;
                                                              선택된 셀의 위치깂
ConnectDB();
                                                              을 가져온뒤 위치깂
if (this.dataGridView1.RowCount == 0)
    return;
                                                              을 바인딩 리스트의
                                                              위치 값으로 지정
int rowIndex = dataGridView1.CurrentRow.Index;
Student s = studentList[rowIndex];
string G = s.Grade;
                                          해당 리스트의 값을 클래스를 통해
string Cc = s.Cclass;
string No = s.No;
                                          get; set; 을 하여 받아서 지정함
string \overline{Na} = s.Name;
string S = s.Score;
string sql = $"update {Table} set Grade = '{G}', Cclass = '{Cc}', No = '{No}', Score = '{S}' where Name={Na}";
MySqlCommand cmd = new MySqlCommand();
cmd.Connection = MyConn;
cmd.CommandText = sql;
cmd.ExecuteNonQuery();
MyConn.Close();
```

학생 수정 Code.

```
public string Grade
{
    get
    {
        return this.studentGradeValue;
    }

    set
    {
        if (value != this.studentGradeValue)
        {
            this.studentGradeValue = value;
            NotifyPropertyChanged();
        }
    }
}
```

```
public string Cclass
{
    get
    {
        return this.studentCclassValue;
    }
    set
    {
        if (value != this.studentCclassValue)
        {
            this.studentCclassValue = value;
            NotifyPropertyChanged();
        }
    }
}
```

```
public string No
{
    get
    {
        return this.studentNoValue;
    }
    set
    {
        if (value != this.studentNoValue)
        {
            this.studentNoValue = value;
            NotifyPropertyChanged();
        }
    }
}
```

```
public string Name
{
    get
    {
        return this.studentNameValue;
    }

    set
    {
        if (value != this.studentNameValue)
        {
            this.studentNameValue = value;
            NotifyPropertyChanged();
        }
    }
}
```

```
public string Score
{
    get
    {
        return this.studentScoreValue;
    }

    set
    {
        if (value != this.studentScoreValue)
        {
            this.studentScoreValue = value;
            NotifyPropertyChanged();
        }
    }
}
```

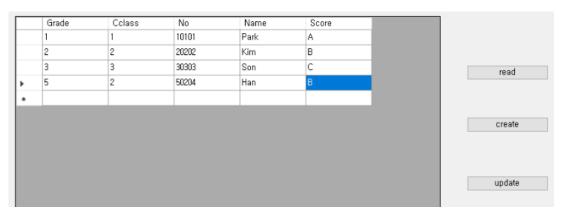
학생에 대한 학년,반,번호,이름,성적등의 정보를 바꾸거나 넘겨주는 Student 클래스 해당 값이 바뀔 경우 INotify PropertyChanged 이벤트 를 발생시키게했다

수 학생 정보 사제.

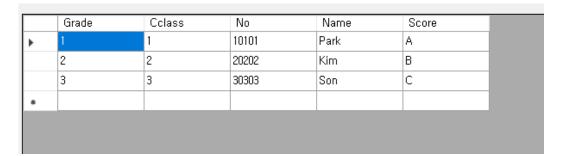
학생 삭제

그리드뷰에 저장된 정보중 원하는 셀을 클릭한 후 Delete 버튼을 누르면 DB와 바인딩리스트에 서 지워지고 바인딩 리스트에서 삭제되었기 때문에 그리드뷰에서 데이터가 삭제된다.

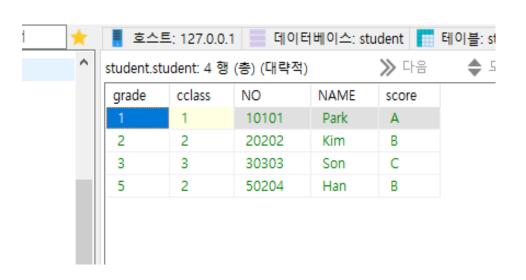
삭제 전



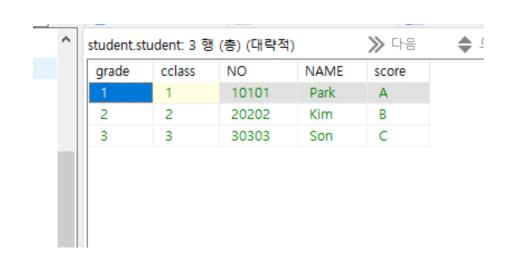
delete 클릭시 데이터 가 삭제된다.



삭제 전 DB



DB 결과



학생 삭제 Code.

```
private void button4_Click(object sender, EventArgs e)
   BindingList<Student> studentList =
        this.StudentBindingSource.DataSource as BindingList<Student>;
   ConnectDB();
   int rowIndex = dataGridView1.CurrentRow.Index;
   Student s = studentList[rowIndex];
   studentList.Remove(studentList[rowIndex]);
   string Na = s.Name;
   Console.WriteLine(Na);
   //studentList.Remove(studentList[rowIndex]);
   string sql = $"delete from {Table} where name = '{Na}'";
   MySqlCommand cmd = new MySqlCommand();
   cmd.Connection = MyConn;
   cmd.CommandText = sql;
   cmd.ExecuteNonQuery();
   MyConn.Close();
```

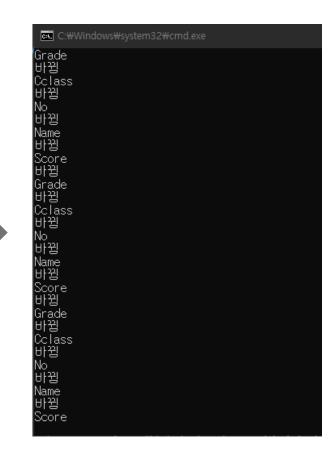
선택된 셀의 위치값에 따라 바인딩 리스트에서 삭제할 값 을 지정해서 삭제해준다

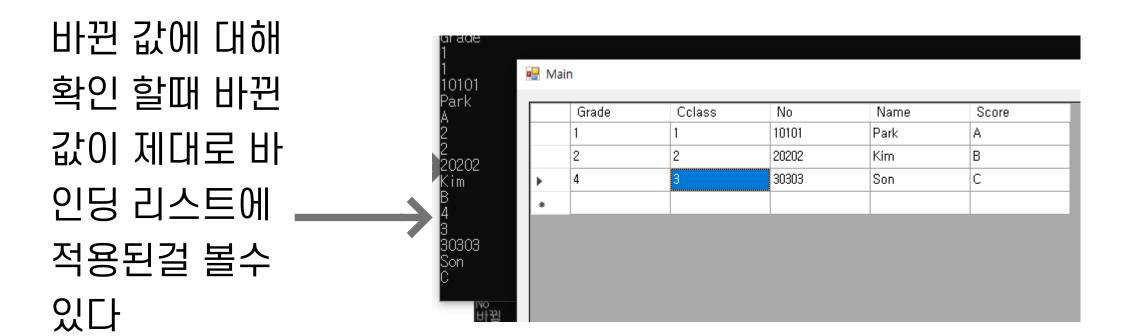
INotifyPrope rtyChanged 확인

INotifyPropertyChanged 변화 확인

바인딩된 리스트와 바인딩 클래스에서 일어나는 변화를 확인한다.

처음 실행시 값 을 읽어들이기 때문에 생기는 변화





질문 있으신 분은 손을 들고 질문해 주세요.