

K B 경 북 산 업 직 업 전 문 학 교

INotifyChan ged 사용하기

2022-03-16

권 용 규

들어가기에
앞서

이번 프로젝트
에 대하여...

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

목 차

01 INotifyPropertyChanged
사용

02 학생 정보 등록

03 학생 정보 수정

04 학생 정보 삭제

05 INotifyPropertyChanged
확인

06 마치며....

01. • INotifyPropertyChanged 사용하기

INotifyPropertyChanged 사용

INotifyPropertyChanged
인터페이스

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

INotifyPropertyChanged
이벤트

```
public event PropertyChangedEventHandler PropertyChanged;
```

INotifyPropertyChanged
이벤트 발생시 작동됨

```
private void NotifyPropertyChanged([CallerMemberName] String propertyName = "")
{
    Console.WriteLine("바뀜");
    Console.WriteLine(propertyName.ToString());
    PropertyChanged?.Invoke(this, new PropertyChangedEventArgs(propertyName));
}
```

02.

학생 정보 등록

기본적인 화면구성.

로그인에 성공시 기본적으로 나오는 고객 멤버 관리 프로그램입니다.
기본적으로 학생추가,학생수정,학생조회,학생제거, 추가로 INotify 기능 동작 체크를 담았습니다.

학생 리스트

Main

	Grade	Cclass	No	Name	Score
▶	1	1	10101	Park	A
	2	2	20202	Kim	B
	3	3	30303	Son	C
*					

read

create

update

delete

check

학생 조회

학생 추가

학생 수정

학생 삭제

INotify 체크

학생 등록

그리드뷰의 셀에 학생에 대한 정보를 입력하고 create 버튼을 클릭시 DB에 학생정보가 삽입되며 해당 정보는 BindingList에도 저장된다

학생 추가 전

	Grade	Cclass	No	Name	Score
▶	1	1	10101	Park	A
	2	2	20202	Kim	B
	3	3	30303	Son	C
*					

read

create

student.student: 3 행 (총) (대략적) >> 다음					
grade	cclass	NO	NAME	score	
1	1	10101	Park	A	
2	2	20202	Kim	B	
3	3	30303	Son	C	

학생 추가 후

추가 결과 →

	Grade	Cclass	No	Name	Score
	1	1	10101	Park	A
	2	2	20202	Kim	B
	3	3	30303	Son	C
▶	4	4	40404	Han	D
*					

read

create

student.student: 4 행 (총) (대략적) >> 다음					
grade	cclass	NO	NAME	score	
1	1	10101	Park	A	
2	2	20202	Kim	B	
3	3	30303	Son	C	
4	4	40404	Han	D	

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();  
}
```

03.

학생 정보 수정

학생 수정

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

	Grade	Cclass	No	Name	Score
	1	1	10101	Park	A
	2	2	20202	Kim	B
	3	3	30303	Son	C
▶	4	4	40404	Han	D
•					

수정 전

정보

create

grade	cclass	NO	NAME	score
1	1	10101	Park	A
2	2	20202	Kim	B
3	3	30303	Son	C
4	4	40404	Han	D

수정 전

DB

	Grade	Cclass	No	Name	Score
	1	1	10101	Park	A
	2	2	20202	Kim	B
	3	3	30303	Son	C
▶	5	2	50204	Han	B
•					

수정 후

정보

create

update

grade	cclass	NO	NAME	score
1	1	10101	Park	A
2	2	20202	Kim	B
3	3	30303	Son	C
5	2	50204	Han	B

수정 후

DB

학생 수정 Code.

```
private void button3_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 = $"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();
}
```

바인딩 리스트를 가져옴

선택된 셀의 위치값
을 가져온뒤 위치값
을 바인딩 리스트의
위치 값으로 지정

해당 리스트의 값을 클래스를 통해
get; set; 을 하여 받아서 지정함

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 INotifyPropertyChanged 이벤트를 발생시키게했다

04. 학생 정보 삭제.

학생 삭제

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

	Grade	Cclass	No	Name	Score
	1	1	10101	Park	A
	2	2	20202	Kim	B
	3	3	30303	Son	C
▶	5	2	50204	Han	B
•					

read

create

update

delete 클릭시 데이터
가 삭제된다.

	Grade	Cclass	No	Name	Score
▶	1	1	10101	Park	A
	2	2	20202	Kim	B
	3	3	30303	Son	C
•					

삭제 전

삭제 전 DB

호스트: 127.0.0.1 데이터베이스: student 테이블: st					
student.student: 4 행 (총) (대략적) >> 다음 << 5					
grade	cclass	NO	NAME	score	
1	1	10101	Park	A	
2	2	20202	Kim	B	
3	3	30303	Son	C	
5	2	50204	Han	B	

DB 결과

student.student: 3 행 (총) (대략적) >> 다음 << 5					
grade	cclass	NO	NAME	score	
1	1	10101	Park	A	
2	2	20202	Kim	B	
3	3	30303	Son	C	

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

05.

INotifyPropertyChanged

확인

INotifyPropertyChanged

변화 확인

바인딩된 리스트와 바인딩 클래스에서 일어나는 변화를 확인한다.

처음 실행시 값을 읽어들이기 때문에 생기는 변화

```
C:\Windows\system32\cmd.exe
Grade
바뀜
Cclass
바뀜
No
바뀜
Name
바뀜
Score
바뀜
Grade
바뀜
Cclass
바뀜
No
바뀜
Name
바뀜
Score
바뀜
Grade
바뀜
Cclass
바뀜
No
바뀜
Name
바뀜
Score
```

값을 바꾸고 넘어가는 순간 바인딩된 값이 바뀌었을 것 같고 이벤트 호출

바뀜	4	3	30303	Son	C
Grade					

바뀐 값에 대해 확인 할때 바뀐 값이 제대로 바인딩 리스트에 적용된걸 볼수 있다

Grade
1
1
10101
Park
A
2
2
20202
Kim
B
4
3
30303
Son
C
바뀜

Main

	Grade	Cclass	No	Name	Score
	1	1	10101	Park	A
	2	2	20202	Kim	B
▶	4	3	30303	Son	C
•					

06. THANK YOU

질문 있으신 분은 손을 들고 질문해 주세요.