

Cel zajęć. Celem zajęć jest zapoznanie z praktycznymi aspektami powiązania modelu obiektowego z modelem okienkowym w C#.

Wprowadzenie teoretyczne. Rozważana w ramach niniejszych zajęć tematyka jest ważna, gdyż w praktyce model obiektowy jest nieodzownie związany z modelem okienkowym. Aby ze zrozumieniem zrealizować zadania, przewidziane do wykonania w ramach zajęć laboratoryjnych, należy znać znaczenie takich pojęć jak: metody formy – Show i ShowDialog, właściwość formy DialogResult.

1. Metody Show i ShowDialog klasy Form

Wywołanie metody „Show” powoduje wyświetlenie utworzonego obiektu typu „Form”.

Wywołanie metody „ShowDialog” powoduje wyświetlenie utworzonego obiektu typu „Form” jako modalnego okna dialogowego.

2. Właściwość DialogResult

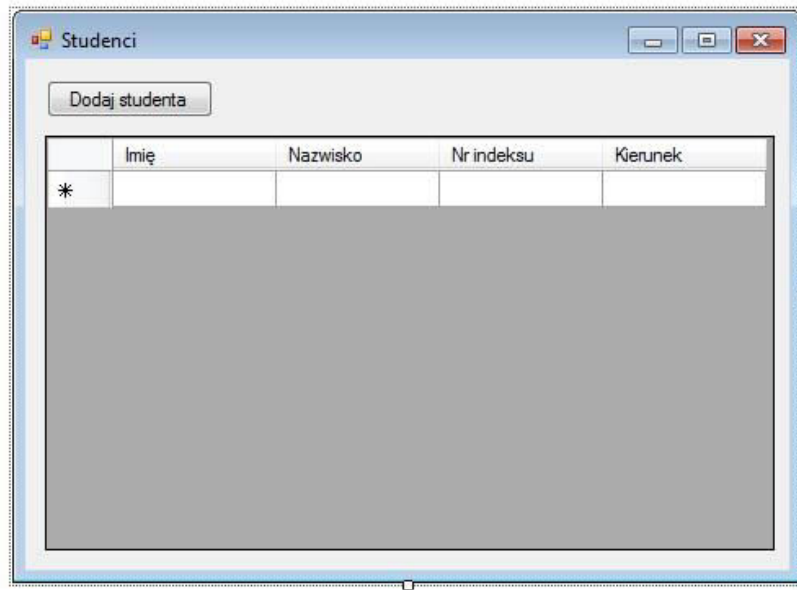
Właściwość ta określa rezultat zwracany przez metodę „ShowDialog”. Jeśli właściwości tej zostanie przypisana jakaś wartość, forma wyświetlona jako okno dialogowe zostanie ukryta, a sterowanie zostanie przekazane do formy wywołującej metodę „ShowDialog”.

Zadanie 1. Proszę zrealizować aplikację okienkową, która powinna odznaczać się następującymi cechami:

- Aplikacja ma umożliwiać dodawania określonych informacji o studentach: imienia, nazwiska, numeru indeksu, kierunku studiów.
- Aplikacja ma wyświetlać te dane w formie tabeli.

Instrukcja wykonania zadania:

- Proszę o zmianę nazwy głównej formy na „Main”.
- Proszę o dodanie do projektu nowego pliku klasy. W tym celu należy kliknąć prawym przyciskiem myszy na nazwie projektu w oknie „Solution Explorer”, a następnie wybrać opcję „Add” -> „Class...”. Następnie proszę o zmianę nazwy pliku na „Student.cs”.
- W pliku tym zostanie utworzona klasa „Student”, proszę o uczynienie jej publiczną i o jej implementację wedle wymogów zadania (musi przechowywać określone informacje o studencie).
- Następnie proszę o dostosowanie głównej formy aplikacji według poniższego rysunku:

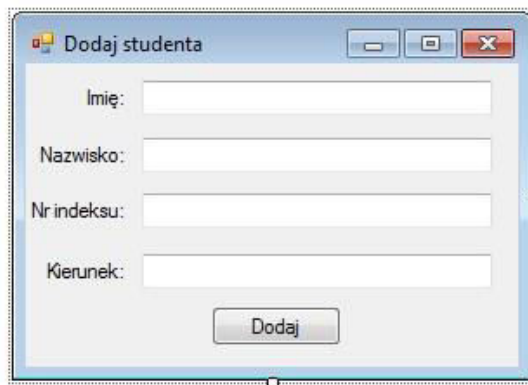


Wykorzystano tu kontrolkę typu „DataGridView”. Zarządzanie kolumnami kontrolki typu „DataGridView” odbywa się za pomocą właściwości „Columns”.

- Następnie proszę o dodanie w klasie głównej formy następującego pola:

```
private List<Student> studenci = new List<Student>();
```

- Proszę o dodanie do projektu nowej formy. W tym celu należy kliknąć prawym przyciskiem myszy na nazwie projektu w oknie „Solution Explorer”, a następnie wybrać opcję „Add” -> „Windows Form...”. Następnie proszę o zmianę nazwy pliku na „DodajStudenta.cs”.
- Następnie proszę o dostosowanie nowo dodanej formy według poniższego rysunku:



- Następnie proszę o dodanie do formy „DodajStudenta” następującego pola oraz konstruktora:

```
private Main mainForm = null;

public DodajStudenta(Main m)
{
    InitializeComponent();
    mainForm = m;
}
```

- Następnie proszę o obsługę zdarzenia „Click” przycisku „Dodaj studenta” na głównej formie. Ma ono powodować wyświetlenie okna umożliwiającego dodanie nowego studenta. W tym celu można wykorzystać następujący fragment kodu:

```
DodajStudenta addStudentForm = new DodajStudenta(this);
DialogResult dr = addStudentForm.ShowDialog();
if (dr == DialogResult.OK)
{
    MessageBox.Show("Dodano studenta!");
}
```

- Następnie proszę o obsługę zdarzenia „Click” przycisku „Dodaj” na formie „DodajStudenta”. Ma ono sprawdzać, czy wypełniono wszystkie pola, następnie dodać nowego studenta do listy studentów zawartej w klasie głównej formy oraz do kontrolki „DataGridView” w celu ich wyświetlenia. W tym celu można wykorzystać następujący fragment kodu:

```
if (textBoxImie.Text == "" || textBoxNazwisko.Text == "" ||
    textBoxNrIndeksu.Text == "" || textBoxKierunek.Text == "")
{
    MessageBox.Show("Uzupełnij wszystkie pola!");
}
else
{
    try
    {
        Student s = new Student(textBoxImie.Text, textBoxNazwisko.Text,
            Convert.ToInt32(textBoxNrIndeksu.Text), textBoxKierunek.Text);
        mainForm.DodajStudenta(s);
        mainForm.DodajStudentaDoDataGridView(s);
        this.DialogResult = DialogResult.OK;
    }
    catch (Exception ex)
    {
        MessageBox.Show("Błąd: " + ex.Message);
    }
}
```

- W klasie głównej formy należy zaimplementować metody „DodajStudenta” oraz „DodajStudentaDoDataGridView”.

Wskazówka:

Dodawanie wiersza do kontrolki typu „DataGridView” można zrealizować w następujący sposób:

```
dataGridView1.Rows.Add(imie, nazwisko, nrIndeksu, kierunek);
```

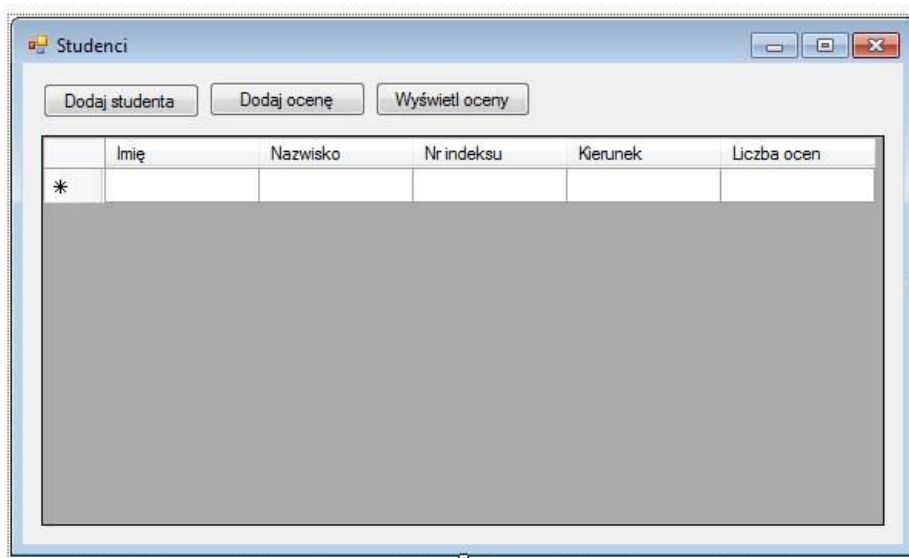
Zadanie 2. Proszę zrealizować aplikację okienkową, która powinna odznaczać się następującymi cechami:

- Aplikacja stanowi rozszerzenie aplikacji z zadania 1.

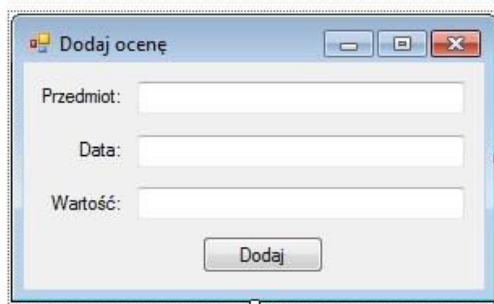
- Aplikacja ma umożliwiać dodawania informacji o ocenach danego studenta i wyświetlanie tych informacji.

Instrukcja wykonania zadania:

- Proszę o dodanie do projektu klasy „Ocena” zawierającej informacje o nazwie przedmiotu, dacie wystawienia oraz wartości oceny.
- Proszę o modyfikację klasy „Student”, polegającej na umożliwieniu jej przechowywania listy ocen.
- Proszę o modyfikację głównej formy aplikacji według rysunku:



- Proszę o ustawienie właściwości „SelectionMode” kontrolki „DataGridView” jako „FullRowSelect”. Spowoduje to możliwość zaznaczenia jedynie całego wiersza w kontrolce, bez możliwości zaznaczenia pojedynczej komórki.
- Proszę o dodanie do projektu formy „DodajOcene” i dostosowanie jej według rysunku:



- Proszę o dodanie do klasy formy „DodajOcene” następującego pola oraz konstruktora:

```
private Student student = null;
```

```
public DodajOcene(Student s)
{
    InitializeComponent();
    student = s;
}
```

- Proszę o obsługę zdarzenia „Click” przycisku „Dodaj ocenę” na głównej formie. Zdarzenie to powinno powodować wyszukanie obiektu studenta, którego dane zostały zaznaczone w kontrolce „DataGridView”. Następnie powinno zostać utworzone i wyświetlone okno „DodajOcene”, przyjmujące w konstruktorze jako parametr obiekt studenta.

Wskazówka:

Do pierwszego spośród zaznaczonych wierszy odwołać można się za pomocą konstrukcji:

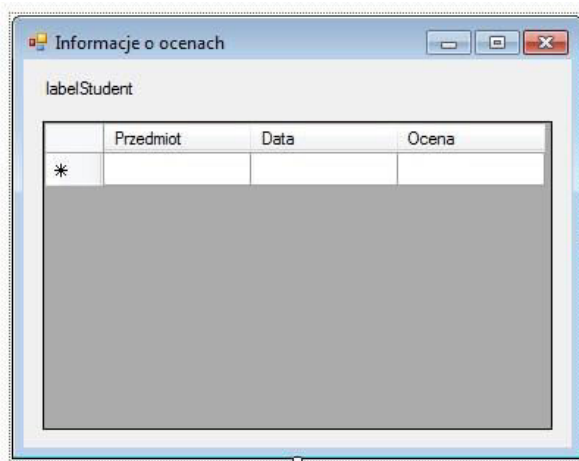
```
dataGridViewStudenci.SelectedRows[0];
```

Do wartości kolumny oznaczającej numer indeksu można odwołać się za pomocą konstrukcji:

```
dataGridViewStudenci.SelectedRows[0].Cells["nrindeksu"].Value;
```

gdzie „nrindeksu” to nazwa danej kolumny.

- Proszę o obsługę zdarzenia „Click” przycisku „Dodaj” na formie „DodajOcene”. Zdarzenie to powinno nie tylko dodawać ocenę danemu studentowi, ale także zwiększać liczbę ocen danego studenta w kontrolce „DataGridView”.
- Proszę o dodanie do projektu formy „WyświetlOceny” i dostosowanie jej według rysunku:



- Następnie proszę o obsługę wyświetlania ocen zaznaczonego studenta. Kontrolka typu „Label” o nazwie „labelStudent” powinna wyświetlać imię, nazwisko, nr indeksu i kierunek studiów studenta.