



Объектно-ориентированное программирование

Лабораторная работа №2

Элемент формы «dataGridView»

- Для того чтобы запретить добавление пользователем строк в dataGridView нужно использовать свойство AllowUserToAddRows, которому задать значение false.

Действия с матрицами

Введите количество строк первой матрицы

Введите количество столбцов первой, строк второй матрицы

Введите количество столбцов второй матрицы

Задать матрицы

	j = 1	j = 2	j = 3
▶	0	0	0
	0	0	0

	j = 1	j = 2
▶	0	0
	0	0

Умножить

Умножение матриц. Каркас

Действия с матрицами

Введите количество строк первой матрицы

Введите количество столбцов первой, строк второй матрицы

Введите количество столбцов второй матрицы

Задать матрицы

Умножить

Дружелюбный интерфейс

Действия с матрицами

Введите количество строк первой матрицы

Введите количество столбцов первой, строк второй матрицы

Введите количество столбцов второй матрицы

Свойства

dataGridView1 System.Windows.Forms.DataGridView

ImeMode	NoControl
MultiSelect	True
ReadOnly	False
RowHeadersWidthSizeMode	EnableResizing
SelectionMode	RowHeaderSelect
StandardTab	False
TabIndex	7
TabStop	True
VirtualMode	False
Visible	True
Проектирование	True
(Name)	False
GenerateMember	True
Locked	False
Modifiers	Private

Листинг «Задать матрицы»

```
// Нажатие на кнопку "Задать" размер матрицы  
int i, j, n, m, k;
```

```
//Проверка наличия ввода размеров  
if ((textBox1->Text!="")&(textBox2->Text!="")&(textBox3->Text!=""))  
{  
    n = System::Convert::ToInt32(textBox1->Text);  
    m = System::Convert::ToInt32(textBox2->Text);  
    k = System::Convert::ToInt32(textBox3->Text);  
}
```

```
//Задаём количество столбцов  
dataGridView1->ColumnCount = m;
```

```
//Называем столбцы первой матрицы  
for (j=0; j<m; j++)  
{  
    dataGridView1->Columns[j]->Name = L"j = "+  
        System::Convert::ToString(j+1);  
    dataGridView1->Columns[j]->Width = 60;  
}
```

```
//Задаём количество строк  
dataGridView1->RowCount = n;  
//Обнуляем значения  
for (i=0; i<n; i++)  
    for (j=0; j<m; j++)  
        dataGridView1->Rows[i]->Cells[j]->Value = L"0";
```

```
//Задаём количество столбцов второй матрицы  
dataGridView2->ColumnCount = k;
```

```
//Называем столбцы первой матрицы  
for (j=0; j<k; j++)  
{  
    dataGridView2->Columns[j]->Name = L"j = "+  
        System::Convert::ToString(j+1);  
    dataGridView2->Columns[j]->Width = 60;  
}
```

```
//Задаём количество строк  
dataGridView2->RowCount = m;  
//Обнуляем значения  
for (i=0; i<m; i++)  
    for (j=0; j<k; j++)  
        dataGridView2->Rows[i]->Cells[j]->Value = L"0";
```

```
dataGridView1->Visible = true;  
dataGridView2->Visible = true;  
button2->Visible = true;
```

Вспомогательные вопросы

- Как добраться до числовых значений матрицы?

```
for (i=0;i<n;i++)
{
    for (j=0;j<m;j++)
    {
        A[i][j] = System::Convert::ToDouble(dataGridView1->Rows[i]->Cells[i]->Value);
    }
}
```

- Как внести данные в ячейку?

```
for (i=0;i<n;i++)
{
    for (j=0;j<k;j++)
    {
        dataGridView3->Rows[i]->Cells[i]->Value=System::Convert::ToString(C[i][j]);
    }
}
```

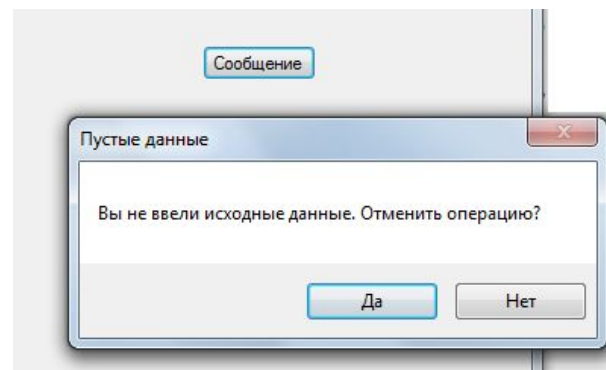
```
textBox3->Text = String::Format("{0:F4}", c);|
```

Обработка исключений

```
try {  
    if (textBox1->Text=="")  
    {  
        throw "E";  
    }  
}  
catch (const char * str) {  
    textBox1->Text = "не введен текст";  
}
```


Вывод всплывающих сообщений. Варианты задания

```
if ( textBox1->Text->Length == 0 )  
{  
    String^ message = "Вы не ввели исходные данные. Отменить операцию?";  
    String^ caption = "Пустые данные";  
    MessageBoxButtons buttons = MessageBoxButtons::YesNo;  
    System::Windows::Forms::DialogResult result;  
  
    result = MessageBox::Show( this, message, caption, buttons );  
    if ( result == System::Windows::Forms::DialogResult::Yes )  
    {  
        this->Close();  
    }  
}
```



```
MessageBoxButtons buttons = MessageBoxButtons::|
```

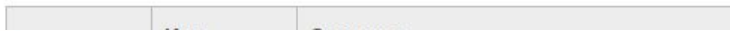
- AbortRetryIgnore
- OK
- OKCancel
- RetryCancel
- value_
- YesNo
- YesNoCancel

Универсальные справки по элементам форм и их свойствам

- Кнопка -

[https://msdn.microsoft.com/ru-ru/library/system.windows.forms.button\(v=vs.110\).aspx](https://msdn.microsoft.com/ru-ru/library/system.windows.forms.button(v=vs.110).aspx)

Свойства



Примеры

В следующем примере кода создается Button, задает его DialogResult Свойства DialogResult:OK, и добавляет его в Form.

```
C# C++ VB
private:
void InitializeMyButton()
{
    // Create and initialize a Button.
    Button^ button1 = gcnew Button;

    // Set the button to return a value of OK when clicked.
    button1->DialogResult = ::DialogResult::OK;

    // Add the button to the form.
    Controls->Add( button1 );
}
```

Задание на лабораторную работу №2

- Программа «Умножение матриц»
- Добавить кнопки «Поиск максимального элемента матрицы», «Поиск минимальных значений в строках»
- Форма контроля: отчет
 - Скриншоты программы
 - Проверка примеров в MathCad
 - Листинг программы