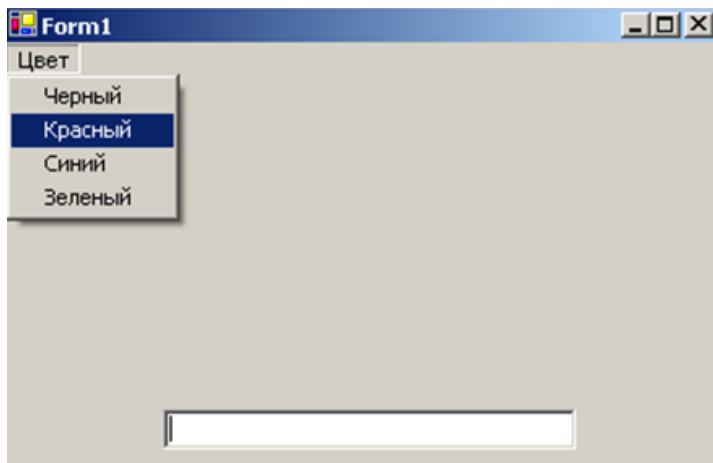


## Лабораторная работа. Работа с элементом MenuStrip (меню формы)

**Задание 1.** Разработайте приложение, которое изменяет задний фон текстового поля, реализовав эту функцию с помощью меню формы.

**Инструментарий:**

Объект	Свойство name
Форма	
Меню формы (MenuStrip)	<i>MenuStrip1</i>
Текстовое поле (textbox)	<i>txt</i>



**Выполнение:**

1. Для создания меню на панели инструментов выберите элемент управления *MenuStrip* и поместите его на форму, в области *(Вводить здесь)* введите меню верхнего уровня с текстом *Цвет*.

2. Переместитесь на нижнюю область и введите текст *Черный*. Заполните элемент *MenuStrip* следующим образом:

**Цвет**

Черный

Красный

Синий

Зеленый

3. Запрограммируйте событие *Click* для каждого пункта; например, для элемента *Черный* необходимо написать следующий код (дважды щелкнув на пункте, чтобы открыть код):

```
private: System::Void черныйToolStripMenuItem_Click(System::Object^ sender, System::EventArgs^ e) {  
    this->txt->BackColor=System::Drawing::Color::Black;  
}
```

4. Запрограммируйте аналогично остальные пункты *Цвет*.

5. Добавьте пункт в меню верхнего уровня: *Цвет* фона и подпункты: Белый, Синий, Красный, изменяющий цвет фона формы.

6. Запрограммируйте подпункты этого пункта, используя код:

*this->BackColor=Color::Red;*

7. Добавьте пункты в меню верхнего уровня: *Разработчик*, *Выход* и напишите код для их обработки:

а) для меню *Разработчик*: использовать окно сообщений

*MessageBox::Show("Разработчик – ФИО (написать свое), "Разработчик",  
MessageBoxButtons::OK, MessageBoxIcon::Information);*

б) для обработки пункта *Выход*: *this->Close();*

**Дополнительное задание:**

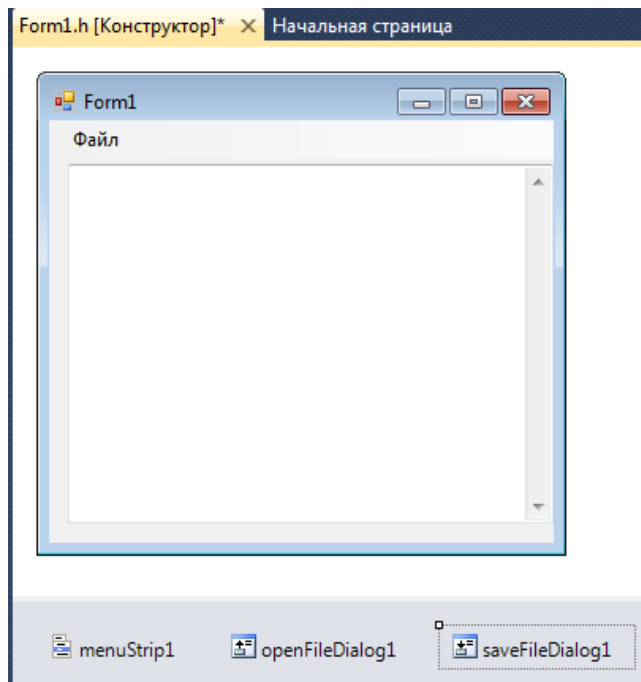
1. При выборе какого-либо пункта после выполнения действий сделайте его недоступным для повторного выбора (свойство *Имя\_элемента->Enabled=false*).

## Задание 2. Создание простого текстового редактора

Создадим приложение, позволяющее открывать текстовый файл и редактировать его, вписывать текст в окне редактора и сохранять его как новый текстовый файл в нужную папку.

Будем использовать следующие элементы: **MenuStrip**, **textBox**, **openFileDialog**, **saveFileDialog**.

1) Разместите все эти элементы на форме. Для элемента **textBox** задайте свойства *Multiline – Yes, ScrollBars – Vertical*. Создайте пункт верхнего уровня меню **Файл** с подпунктами: *Открыть, Сохранить как..., Выход*. Добавьте на форму элементы: *openFileDialog, saveFileDialog*. После добавления они отобразятся внизу конструктора формы.



Программные коды:

### Программный код для загрузки формы (двойной щелчок на форме)

```
private: System::Void Form1_Load(System::Object^ sender, System::EventArgs^ e) {
    this->Text = "Текстовый редактор";
    openFileDialog1->FileName = "D:\\Text2.txt";
    openFileDialog1->Filter = "Текстовые файлы (*.txt)|*.txt|All files (*.*)|*.*";
    saveFileDialog1->Filter = "Текстовые файлы (*.txt)|*.txt|All files (*.*)|*.*";
}
```

### Программный код для пункта **Открыть**.

```
private: System::Void открытьToolStripMenuItem_Click(System::Object^ sender,
System::EventArgs^ e) {
    openFileDialog1->ShowDialog();
    if (openFileDialog1->FileName == nullptr) return;
    try
    {
        auto MyReader = gcnew IO::StreamReader(openFileDialog1->FileName,
        System::Text::Encoding::GetEncoding(1251));
        textBox1->Text= MyReader->ReadToEnd();
        MyReader->Close();
    }
```

```

}
catch (IO::FileNotFoundException^ Ситуация)
{
    MessageBox::Show(Ситуация->Message + "\nФайл не найден", "Ошибка",
    MessageBoxButtons::OK, MessageBoxIcon::Exclamation);
}
catch (Exception^ Ситуация)
{
    MessageBox::Show(Ситуация->Message, "Ошибка", MessageBoxButtons::OK,
    MessageBoxIcon::Exclamation);
}
}

```

### **Программный код для пункта Сохранить как.**

```

private: System::Void сохранитьКакToolStripMenuItem_Click(System::Object^ sender,
System::EventArgs^ e) {
    saveFileDialog1->FileName = openFileDialog1->FileName;
    if (saveFileDialog1->ShowDialog() == System::Windows::Forms::DialogResult::OK) Save();
}
void Save()
{
    try
    {
        // Создание экземпляра StreamWriter для записи в файл:
        auto MyWriter = gcnew IO::StreamWriter(saveFileDialog1->FileName, false,
        System::Text::Encoding::GetEncoding(1251));
        MyWriter->Write(textBox1->Text);
        MyWriter->Close(); textBox1->Modified = false;
    }
    catch (Exception^ Ситуация)
    {
        MessageBox::Show(Ситуация->Message, "Ошибка", MessageBoxButtons::OK,
        MessageBoxIcon::Exclamation);
    }
}

```

### **Программный код для пункта Выход.**

```

private: System::Void выходToolStripMenuItem_Click(System::Object^ sender,
System::EventArgs^ e) {
    this->Close();
}

```

**Дополнительное задание.** Добавьте в меню верхнего уровня пункт **Цвет текста** с подпунктами: Синий, Красный, Зеленый, Черный, позволяющий менять цвет текста в окне редактора, пункт **Стиль текста** с подпунктами: Курсив, Жирный, Обычный, позволяющий менять начертание текста в окне редактора.

*P.S. Установка соответствующих свойств описана в лабораторной работе WF\_2*