

Программирование на C++



| Минцифры
РОССИИ

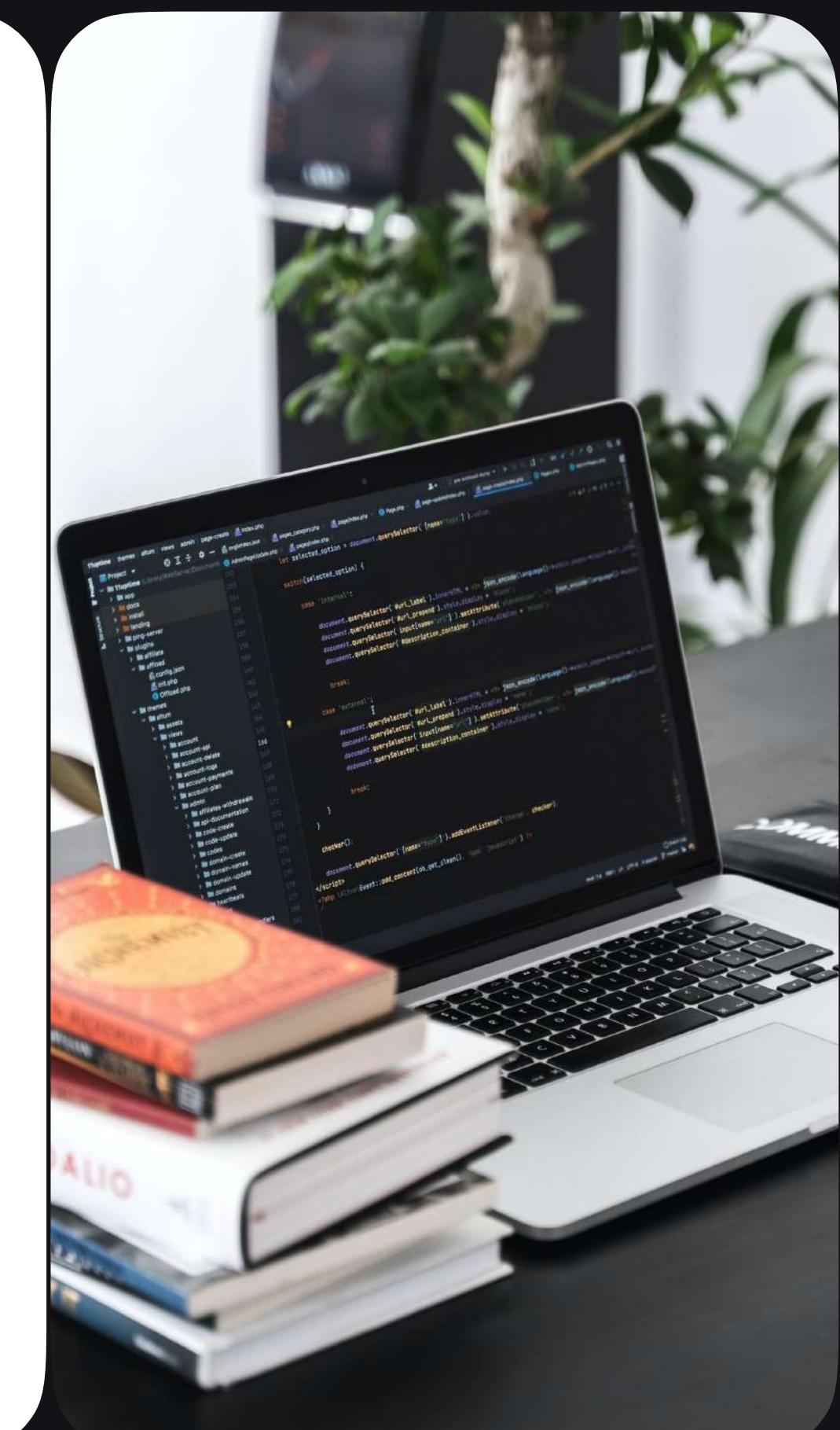
UCHI DOMA

20.35
УНИВЕРСИТЕТ

Урок 9 Модуль 4

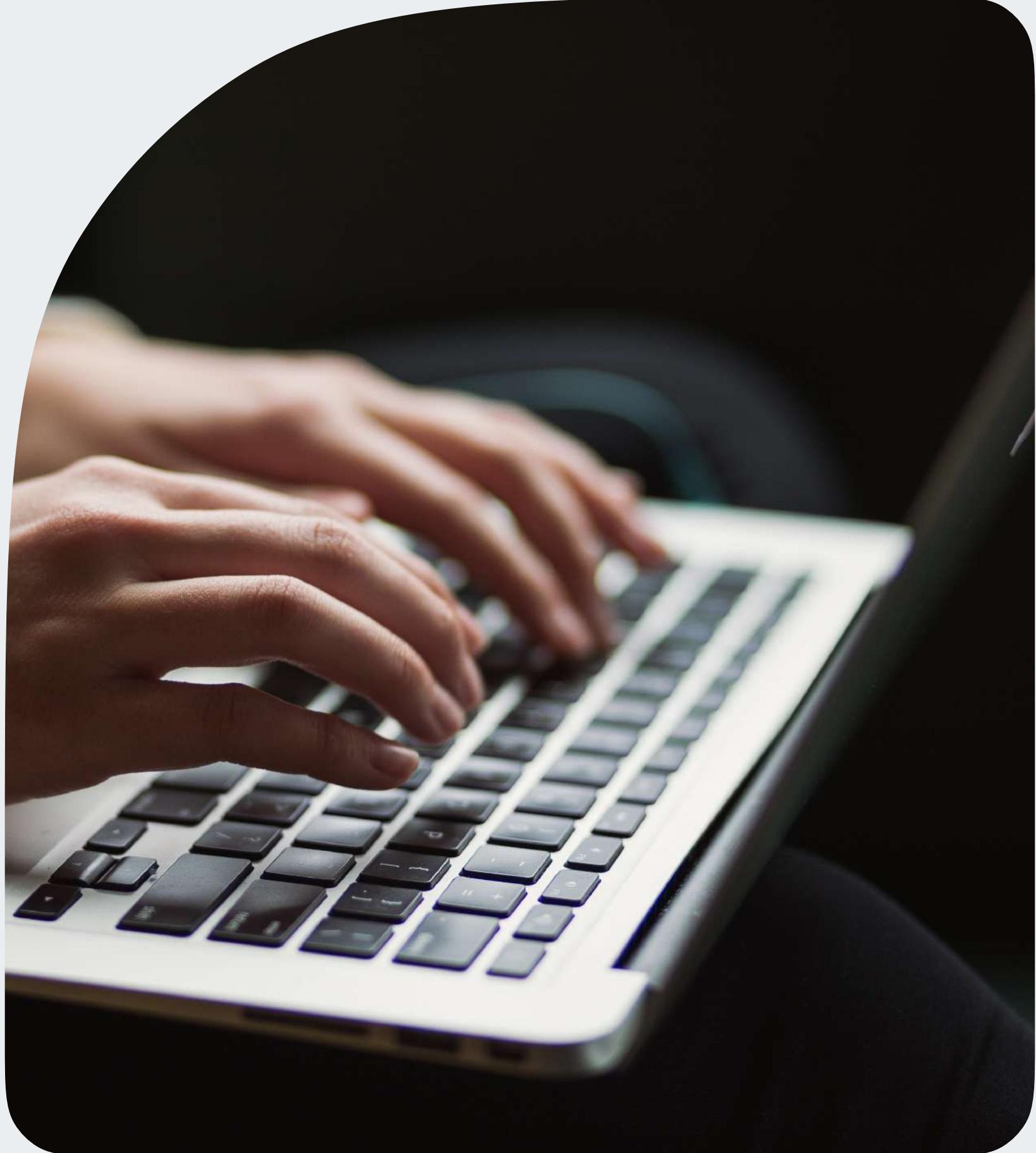
Создание визуальных компонентов (продолжение)

Полезные материалы



Цели урока

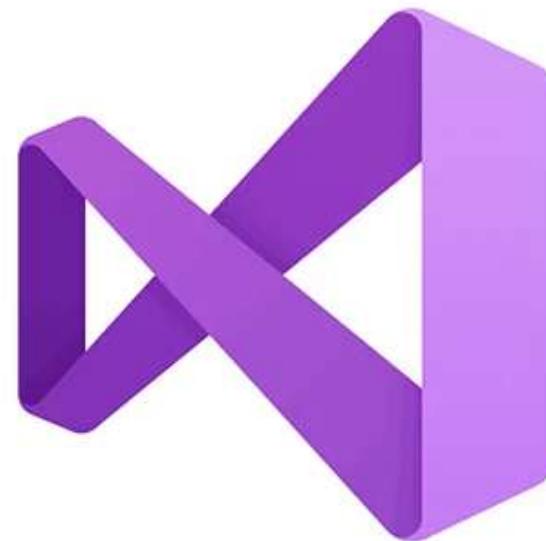
- ❖ Завершить разработку приложения «Калькулятор»
- ❖ Отработать на практике написание приложений с графическим интерфейсом на C++



Практика 9

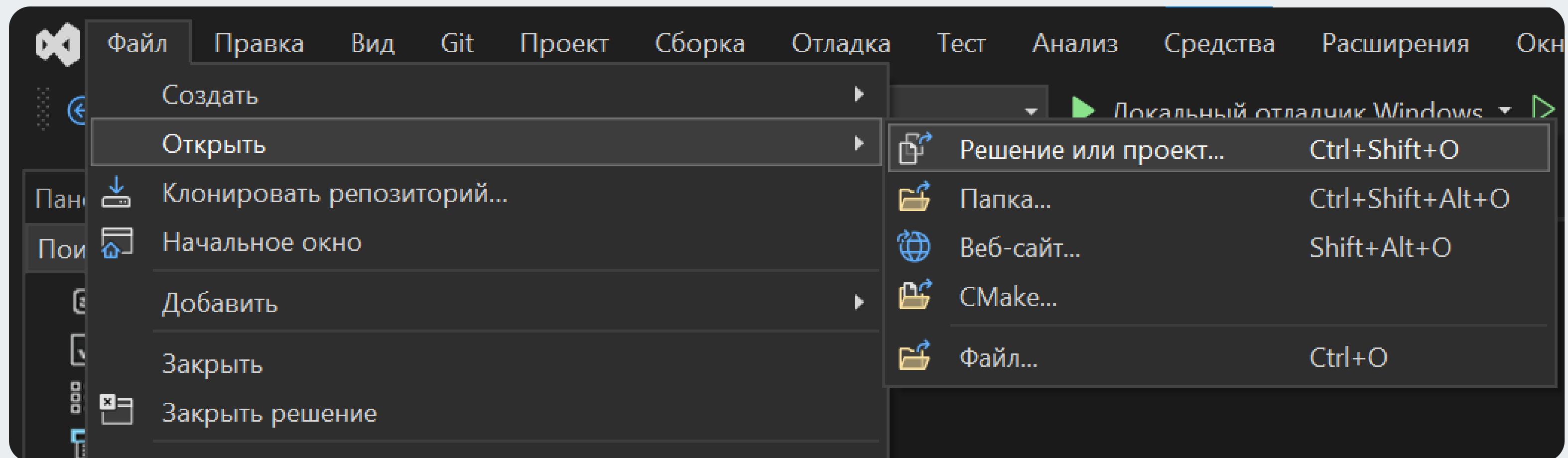
Будем работать в MS Visual Studio.

Запустите Visual Studio.

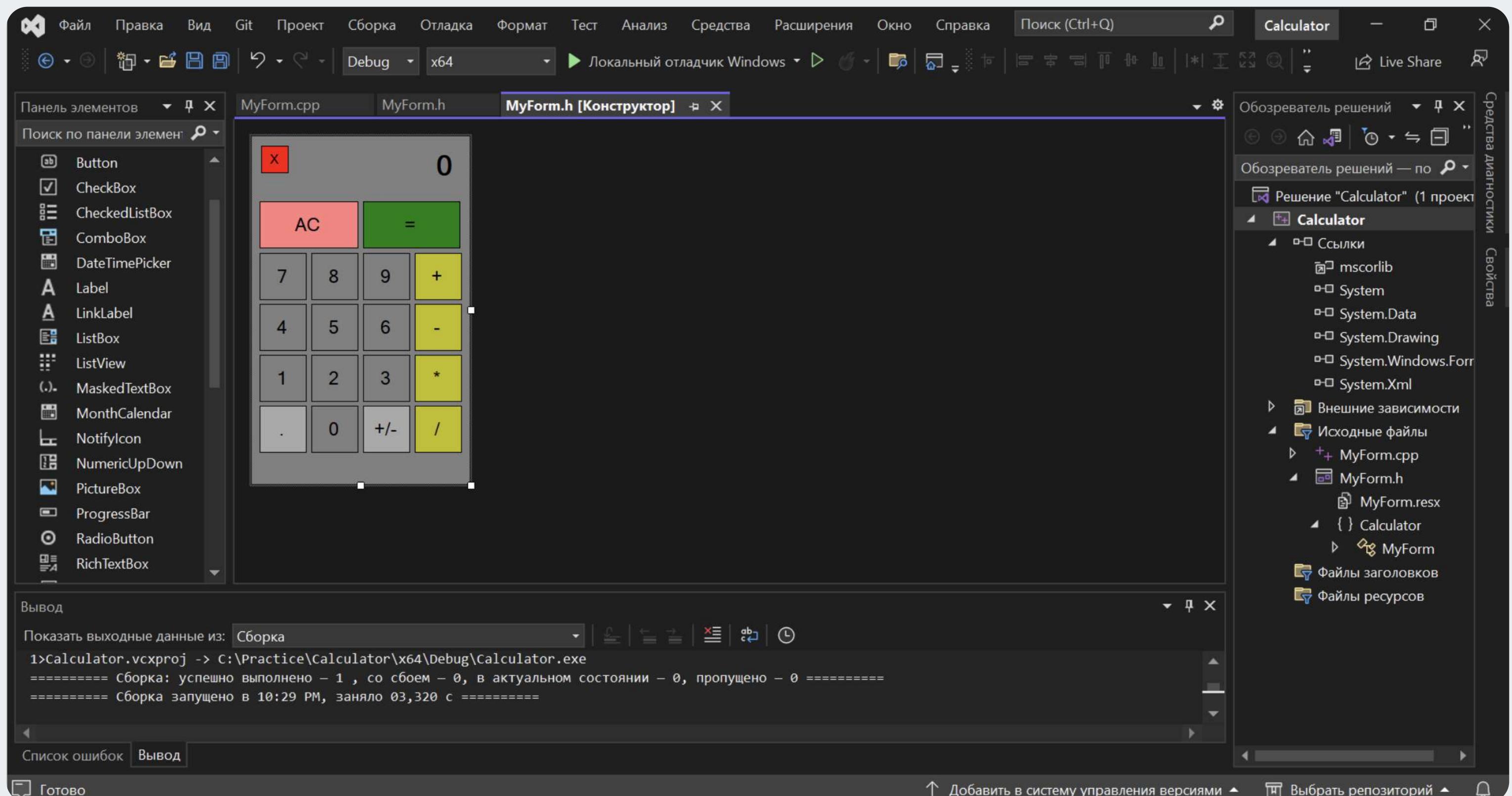


Visual
Studio

Откройте проект «Calculator»



Откройте проект «Calculator»



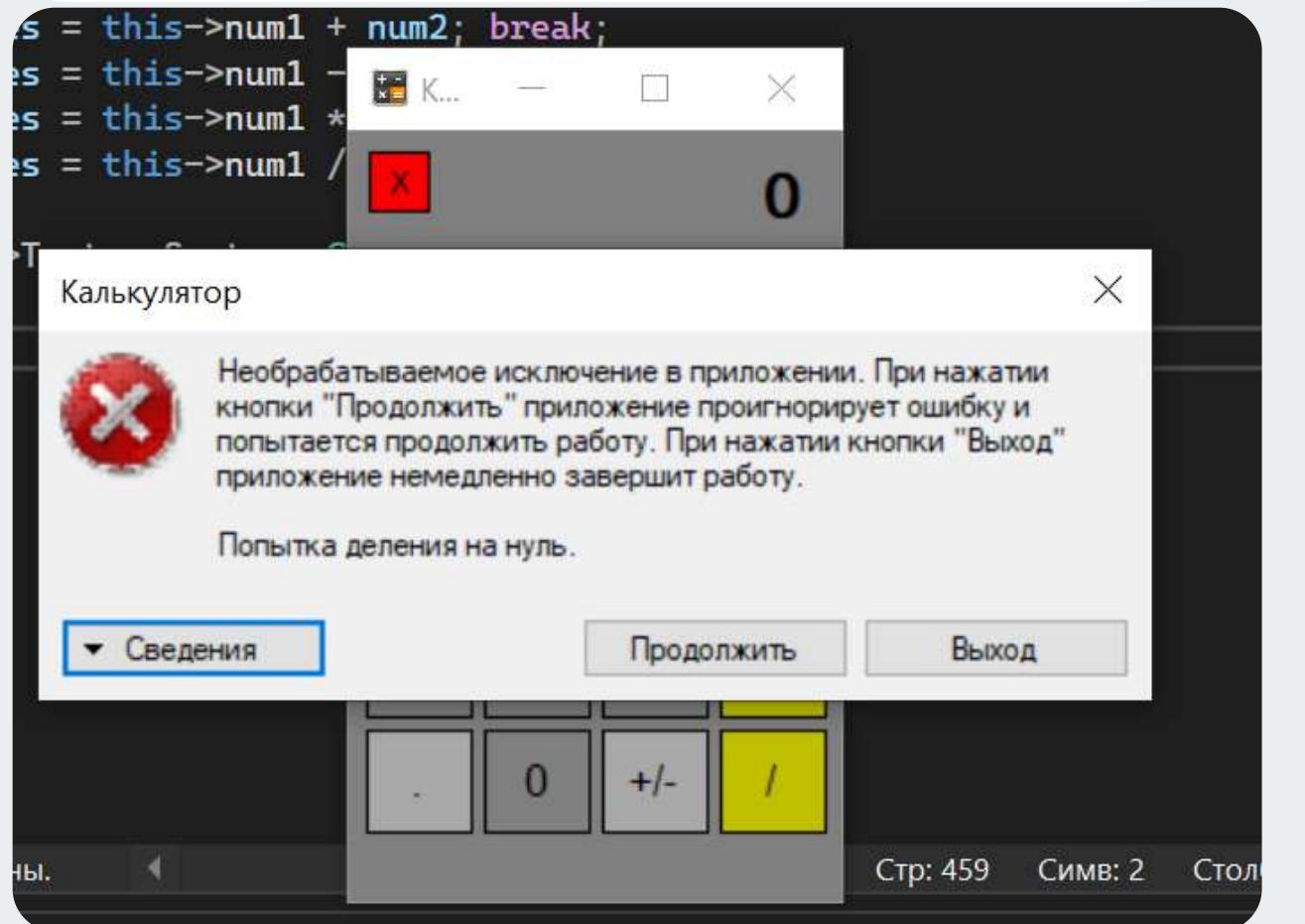
Деление на 0

Попробуйте разделить любое число на 0.

Что происходит?

Деление на 0

Появляется сообщение:



Деление на 0

Дорабатаем функцию btn_result_click, добавим проверку num2 на 0.

```
private: System::Void btn_result_click(System::Object^ sender, System::EventArgs^ e)
{
    int num2= System::Convert::ToInt32(this->label1->Text);
    int res=0;
    switch (this->op)
    {
        case '+': res = this->num1 + num2; break;
        case '-': res = this->num1 - num2; break;
        case '*': res = this->num1 * num2; break;
        case '/':
            if (num2 == 0)
                MessageBox::Show("Деление на 0!", "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error);
            else
                res = this->num1 / num2;
            break;
    }
    this->label1->Text = System::Convert::ToString(res);
}
```

Деление на 0

Дорабатаем функцию `btn_result_click`, добавим проверку `num2` на 0.

```
MessageBox::Show("Деление на 0!", "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error);
```

Класс диалоговых
окон

Сообщение
в окне

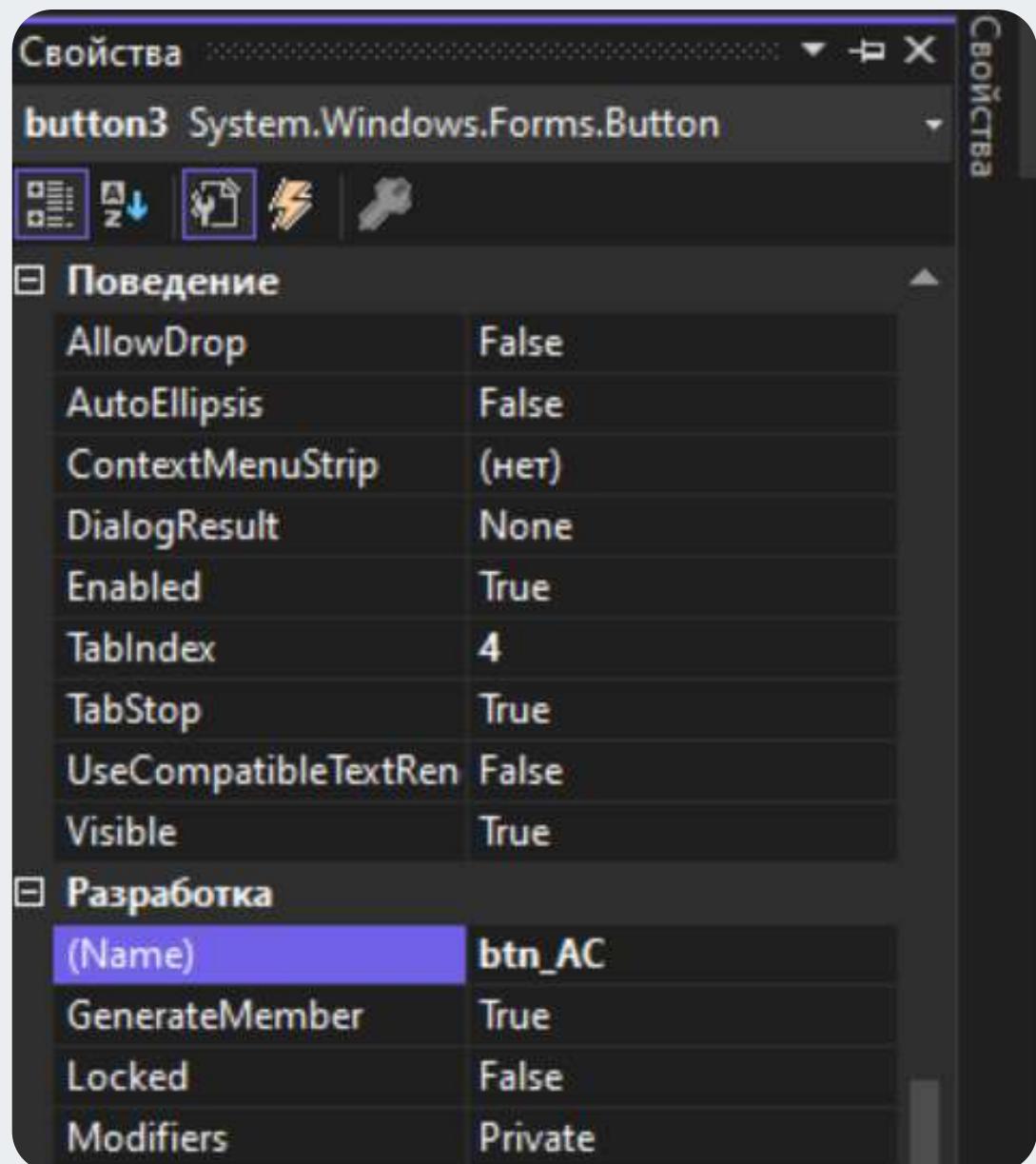
Заголовок
окна

Кнопки окна

Иконка окна



Реализуем функционал кнопки «AC»



Переименуем
кнопку в btn_AC



Реализуем функционал кнопки «AC»

Создайте функцию-обработчик события click
(кликните 2 раза по кнопке в конструкторе).

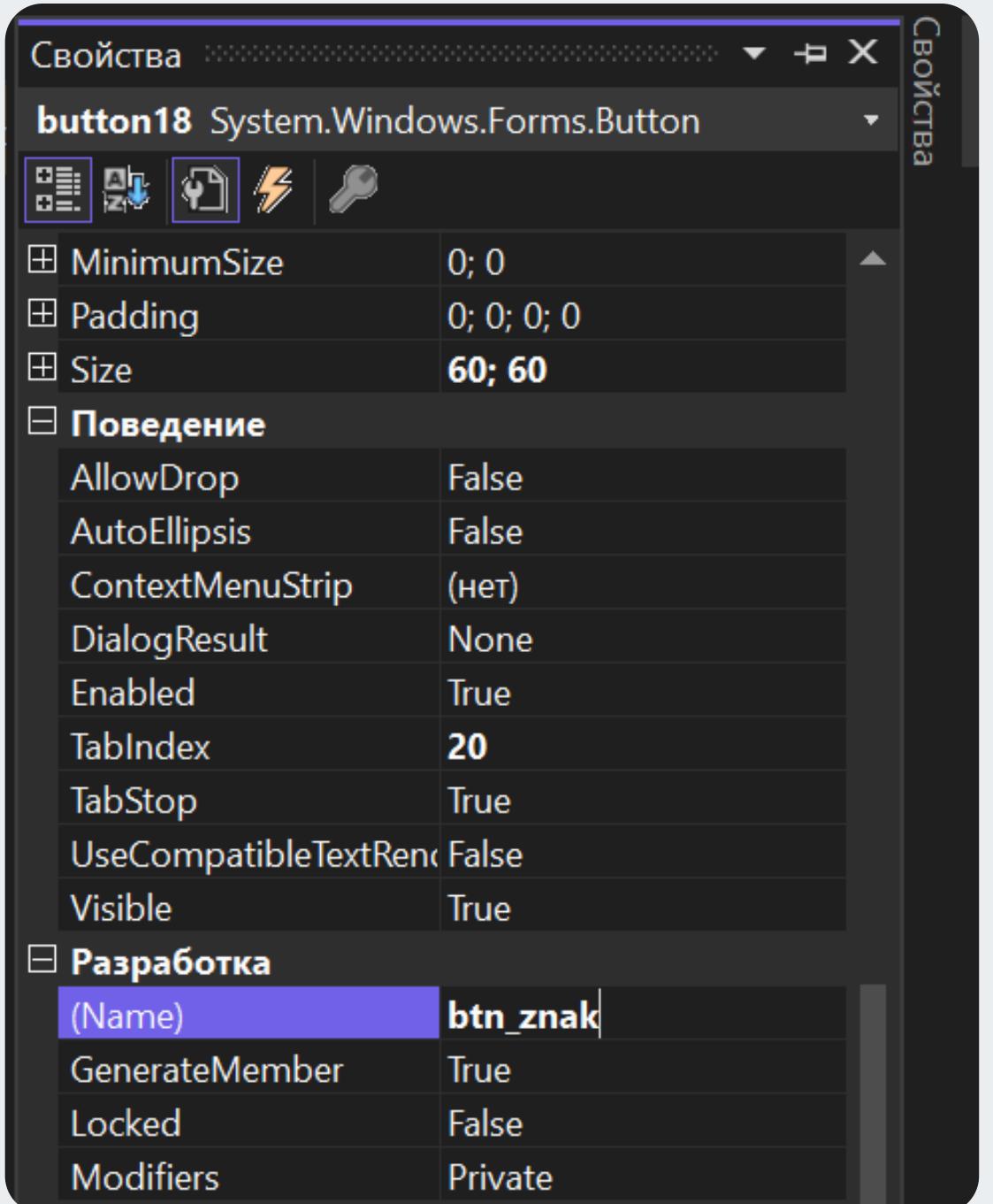
```
private: System::Void btn_AC_Click(System::Object^ sender, System::EventArgs^ e) {  
}  
};
```

Реализуем функционал кнопки «AC»

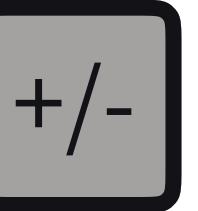
Будем выводить в label1 – «0», и присваивать переменной op=‘ ‘.

```
private: System::Void btn_AC_Click(System::Object^ sender, System::EventArgs^ e)
{
    this->label1->Text = "0";
    this->op = ' ';
}
```

Реализуем функционал кнопки «+/-»



Переименуем кнопку
в btn_znak



Реализуем функционал кнопки «+/-»

Создайте функцию-обработчик события click
(кликните 2 раза по кнопке в конструкторе).

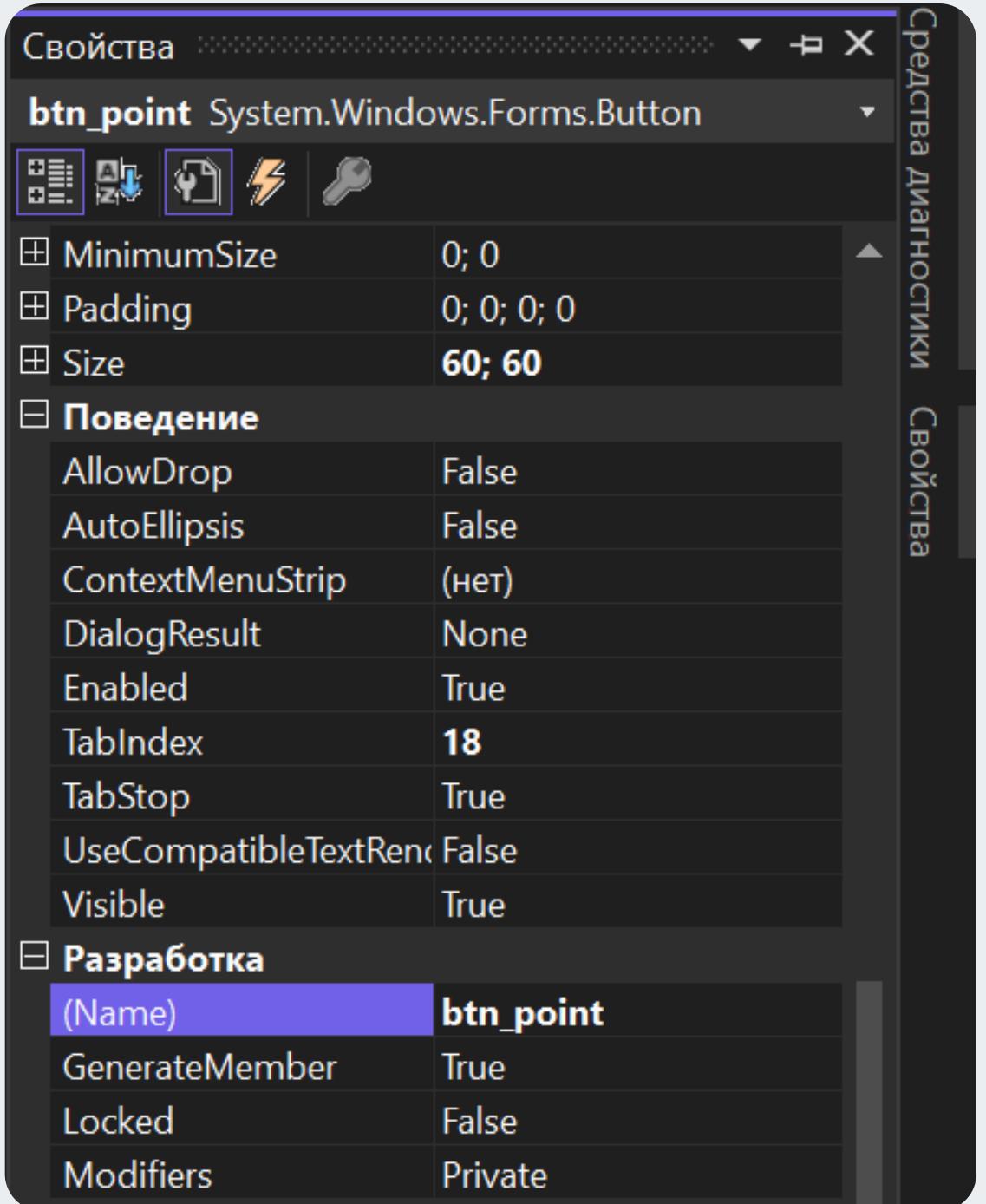
```
private: System::Void btn_znak_Click(System::Object^ sender, System::EventArgs^ e) {  
}
```

Реализуем функционал кнопки «+/-»

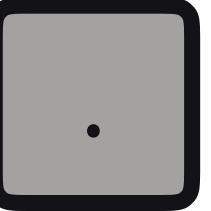
При нажатии кнопки будем домножать введенное число на -1.

```
private: System::Void btn_znak_Click(System::Object^ sender, System::EventArgs^ e)
{
    this->label1->Text = System::Convert::ToString(System::Convert::ToInt32(this->label1->Text) * -1);
}
```

Реализуем функционал кнопки «.»



Переименуем кнопку
в `btn_point`



Реализуем функционал кнопки «.»

При нажатии кнопки будем добавлять запятую к label1.

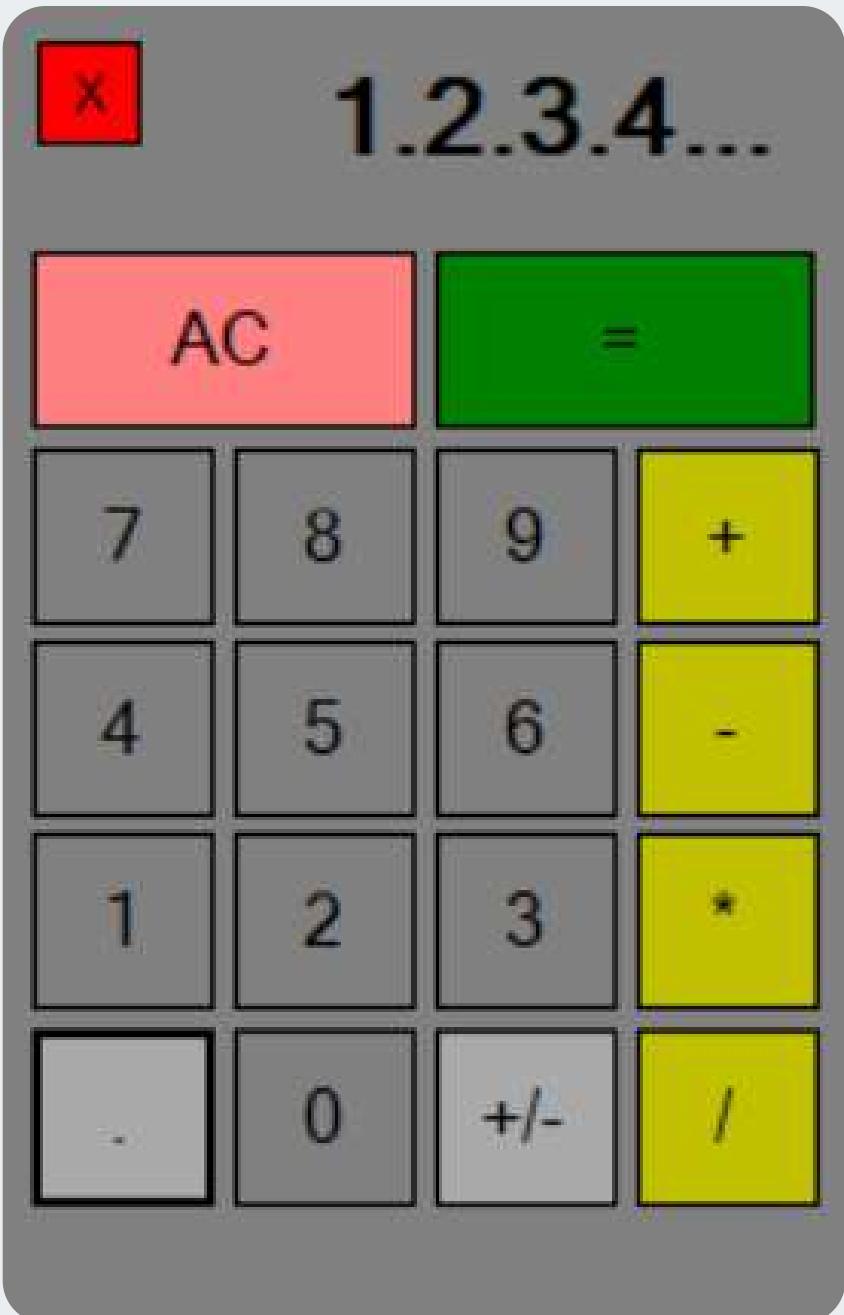
```
private: System::Void btn_point_Click(System::Object^ sender, System::EventArgs^ e)
{
    this->label1->Text = this->label1->Text + ",";
}
```

Реализуем функционал кнопки «..»

Запустите приложение и проверьте работу кнопки.

Корректно работает?

Реализуем функционал кнопки «.»



Работает некорректно.

Запятая добавляется неограниченное количество раз.

Реализуем функционал кнопки «.»

Будем проверять, есть ли уже запятая в числе.



Используем метод **Text->Contains(s)**. Данный метод возвращает True, если аргумент s найдет в строке Text, и возвращает значение False, если не найден.

Будем определять, есть ли в label1 уже запятая:

```
this->label1->Text->Contains(",")
```

Нас интересует случай, когда запятой нет, то есть когда будет возвращено значение False, поэтому в условии запишем:

```
!this->label1->Text->Contains(",")
```

Реализуем функционал кнопки «.»

Будем проверять, есть ли уже запятая в числе.

```
private: System::Void btn_point_Click(System::Object^ sender, System::EventArgs^ e)
{
    if (!this->label1->Text->Contains(","))
        this->label1->Text = this->label1->Text + ",";
}
```

Реализуем функционал кнопки «.»

Проверьте функционал кнопки «.»

Корректно работает?

Реализуем функционал кнопки «.»



Работает не корректно, появляется сообщение об ошибке.

Теперь у нас число с запятой, а мы конвертируем его в Int32 (целое число).

Реализуем функционал кнопки «.»

Изменим тип переменной num1, num2 и res на double.

```
private: int num1;
```



```
private: double num1;
```

```
int num2= System::Convert::.ToDouble(this->label1->Text);  
int res=0;
```



```
double num2= System::Convert::.ToDouble(this->label1->Text);  
double res=0;
```

Реализуем функционал кнопки «.»

В коде конвертация в int32 (System::Convert::ToInt32)
заменяем на конвертацию в Double
(System::Convert::ToDouble).

Реализуем функционал кнопки «.»

Теперь арифметические операции с действительными числами работают корректно.