

# Интегрированные среды разработки приложений (IDE)

# Интегрированная среда разработки (ИСР)

ИСР (Integrated development environment) – среда, в которой есть все необходимое для проектирования, запуска и тестирования программного продукта.

Составные части ИСР:

1. Текстовый редактор.
2. Транслятор.
3. Средства автоматической сборки проекта.
4. Инструментальные средства проектирования графического интерфейса пользователя.
5. Отладчик.

# Ссылки для скачивания

- Qt Creator

<https://www.qt.io/download-open-source>




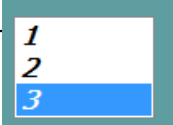
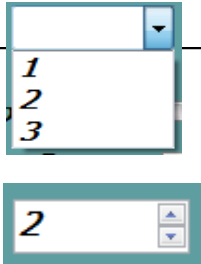

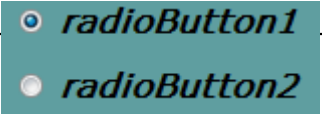
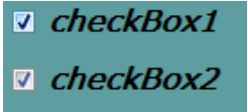
- Visual Studio

<https://visualstudio.microsoft.com/ru/free-developer-offers/>

- Wx Dev C++ (мало документации, давно не выходили новые версии)

<http://wxdsgn.sourceforge.net/?q=node/4>

# Визуальные компоненты

<i>Компонент</i>	<i>QT Creator</i>	<i>Visual Studio</i>	<i>Внешний вид</i>
Текстовая метка	Label	Label	
Окно ввода текста (одна строка)	LineEdit	TextBox	
Окно ввода текста (несколько строк)	TextEdit	TextBox (Multiline = true)	
Список для выбора одного из элементов	ListWidget	ListBox	
Выпадающий список для выбора одного из элементов	ComboBox	ComboBox	
Окно для ввода (изменения) целого числа	SpinBox	NumericUpDown	
Кнопка	PushButton	Button	
Кнопка выбора одного из вариантов	RadioButtton	RadioButtton	

# СВОЙСТВА ВИЗУАЛЬНЫХ КОМПОНЕНТОВ

- Text для однострочных компонентов
- Text для многострочных компонентов
- `setStyleSheet (color, background-color) / BackColor, ForeColor`
- Enabled
- Font
- `setGeometry ( x, y, width, height) / Height, Width, Left, Top`
- Items
- Visible
- `isChecked / Checked`
- Value

# Изменение свойств компонента

- ввод текста или выбор элемента в инспекторе объектов
- кликом правой кнопки мыши на компоненте и изменение через специальное окно
- изменение кодом программы

# События (слоты)

- Click
- DoubleClick
- Change
- Activated

# Создание слота / события

## В QT:

- для выделенного на форме компонента в его контекстном меню (клик правой кнопкой мыши) выбрать пункт Перейти к слоту, затем выбрать нужный слот,

или

- объявить и реализовать слот в программном коде самостоятельно.

## В VS:

- для выделенного на форме компонента в инспекторе объектов перейти на вкладку События (Events), найти нужное событие, сделать напротив него в пустой строке двойной клик мыши,

или

- объявить и реализовать событие в программном коде самостоятельно.



# Text для однострочных компонентов

Получить (изменить) содержимое текстовой метки

Применяется для Label, LineEdit/TextBox, PushButton/Button, RadioButtton, CheckBox

- QT

```
QString str = ui->label->text(); //получить текст метки  
ui->label->setText("Hello, world!"); //изменить текст метки  
QString str = ui->lineEdit->text(); //получить текст окна  
ui->lineEdit->setText("Hello, world!"); //изменить  
ui->pushButton->setText("Выполнить");
```

- VS

```
System::String ^str = label1->Text; //получить текст метки  
label1->Text = str + "Hello, world!"; //изменить текст
```

# Text для многострочных компонентов

Получить (изменить) текст многострочного компонента

Применяется для textEdit/TextBox

- Qt

```
ui->textEdit->setText("abc");
```

```
ui->textEdit->append("de");
```

- VS

```
textBox3->Multiline = true;
```

```
textBox3->Text = textBox3->Text + textBox1->Text +  
Environment.NewLine;
```

# setStyleSheet (color, background-color) / ForeColor, BackColor

Изменить цвет компонента (фона / содержимого)

Применяется для всех компонентов

- Qt

```
ui->centralWidget->setStyleSheet("background-color: green");
```

```
ui->label->setStyleSheet("color: red;");
```

```
ui->textEdit->setStyleSheet("color: blue; background-color: yellow;");
```

- VS

```
label1->ForeColor = Color::Red;
```

```
label1->BackColor = Color::Green;
```

# Enabled

Сделать компонент доступным (недоступным) для событий

Применяется для всех компонентов

- Qt

```
ui->lineEdit->setEnabled(true); //можно вводить текст
```

```
ui->pushButton->setEnabled(false); //нельзя нажать
```

- VS

```
button1->Enabled = false;
```

# Font

Настроить параметры шрифта в компоненте

Применяется для всех компонентов

- Qt

```
ui->lineEdit->setFont(QFont("Arial", 18, 75, true));
```

QFont:

1-й параметр - Название, 2-й параметр - размер, 3-й параметр – вес (75 – жирный), 4-й параметр – включить курсив (true – включить)

- VS

```
Drawing::Font^ f = gcnew Drawing::Font("Arial", 18,  
FontStyle::Bold | FontStyle::Italic);
```

```
button1->Font = f;
```

# setGeometry ( x, y, width, height) / Height, Width, Left, Top

Настроить положение и размер компонента

Применяется для всех компонентов

- Qt

```
ui->lineEdit->setGeometry(ui->lineEdit->x(),  
                           ui->lineEdit->y(),  
                           ui->lineEdit->width() + 10,  
                           ui->lineEdit->height() + 5);
```

- VS

```
textBox1->Width = 100;  
textBox1->Height = 20;  
textBox1->Top = 0;  
textBox1->Left = 10;
```

# Items

Добавить элементы в список компонента

Применяется для ListWidget/ListBox, ComboBox

- Qt

```
ui->comboBox->addItem( "%" );
```

- VS

```
listBox1->Items->Add( "abc" );
```

```
listBox1->Items->Insert(1, "%" );
```

```
listBox1->Items->Clear();
```

# Visible

Управлять видимостью компонента

Применяется для всех компонентов

- Qt

```
ui->label->setVisible(true); //сделать видимым
```

```
ui->pushButton->setVisible(false);
```

- VS

```
textBox3->Visible = true;
```

```
comboBox1->Visible = !comboBox1->Visible;
```



# isChecked / Checked

Проверить выбор элементов сгруппированных кнопок

Применяется для RadioButton, CheckBox

- Qt

```
if(ui->radioButton->isChecked())  
    ui->label->setText("isChecked");
```

- VS

```
if(radioButton1->Checked)  
    comboBox1->Visible = !comboBox1->Visible;
```

# Value

Получить (изменить) численное значение компонента с кнопками UpDown

Применяется для SpinBox/NumericUpDown

- Qt

```
int x = ui->spinBox->value();
```

```
ui->spinBox->setValue(x+1);
```

- VS

```
int x = Convert::ToInt32(numericUpDown1->Value);
```

```
x += 3;
```

```
numericUpDown1->Value = x;
```

# Создание слота DoubleClick в Qt

- В заголовочном файле ( .h), в разделе private slots класса MainWindow объявить слот:

```
void mouseDoubleClickEvent(QMouseEvent *event);
```

- В файле с кодом программы ( .cpp) описать реализацию слота:

```
void MainWindow::mouseDoubleClickEvent(QMouseEvent  
*event){
```

```
//тут описаны необходимые действия
```

```
}
```

где MainWindow – имя класса программы

## Другие события

- `MouseMove`;
- `MouseDown`;
- `MouseUp`.

# Всплывающее сообщение

- Qt

```
#include <QMessageBox>
```

```
QMessageBox msg;
```

```
msg.setText("Hello, world!");
```

```
msg.setWindowTitle("My first program");
```

```
msg.setStandardButtons(QMessageBox::Ok | QMessageBox::Cancel);
```

```
msg.setIcon(QMessageBox::Warning);
```

```
msg.exec();
```

- VS

```
MessageBox::Show("Hello, world!", "My first program",  
MessageBoxButtons::OKCancel, MessageBoxIcon::Exclamation);
```

# Функции преобразования текста в численное значение и обратно

## Qt Creator:

- `текст.toInt()` – преобразование из строки (`QString`) в целое
- `текст.toDouble()` - преобразование из строки (`QString`) в вещественное
- `QString::number(значение)` - преобразование из числа в строку (`QString`)

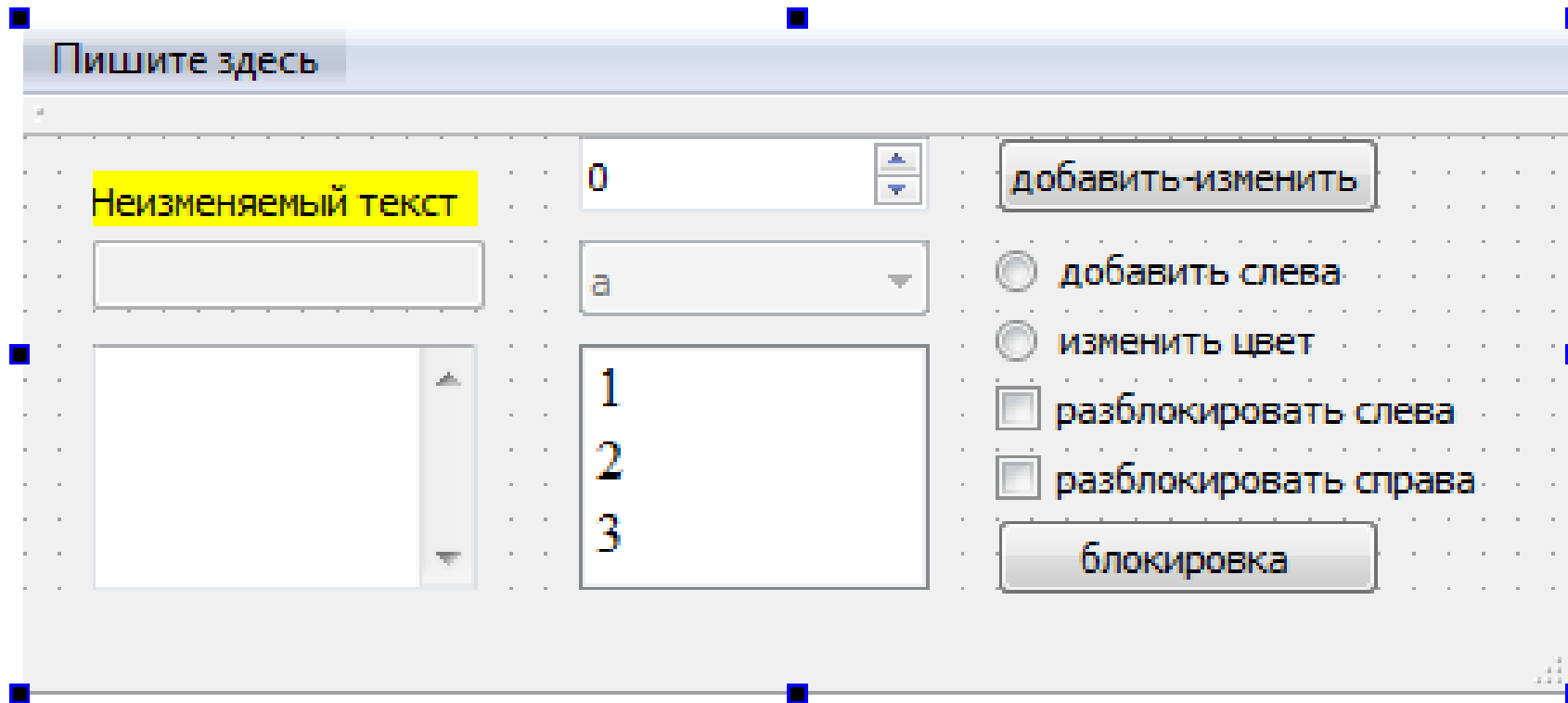
## Visual Studio:

- `Conver::ToInt32(текст)` - преобразование из строки (`System::String`) в целое
- `Conver::ToDouble(текст)` - преобразование из строки (`System::String`) в вещественное
- `Conver::ToString(значение)` - преобразование из числа в строку (`System::String`)

# Средства отладки программы

1. Окно сообщений об ошибках
2. Окно Watch
3. Точки прерывания
4. Проход по шагам:
  - Step Over
  - Trace Into
  - Trace to Next Source Line
  - Run to Cursor
  - Run Until Return

# Проект формы. Qt Creator





# Исходный код. Qt Creator

```
void MainWindow::on_pushButton_clicked(){
    if(ui->checkBox->checkState())
        ui->lineEdit->setEnabled(!ui->lineEdit->isEnabled());
    if(ui->checkBox_2->checkState())
        ui->comboBox->setEnabled(!ui->comboBox->isEnabled());
}
void MainWindow::on_pushButton_2_clicked(){
    if(ui->radioButton->isChecked())
        ui->textEdit->append(ui->lineEdit->text());
    if(ui->radioButton_2->isChecked())
        ui->lineEdit->setStyleSheet("background-color: green");
}
void MainWindow::on_comboBox_currentTextChanged(const QString &arg1){
    ui->listWidget->insertItem(1, arg1);
}
void MainWindow::on_listWidget_itemDoubleClicked(QListWidgetItem *item){
    ui->spinBox->setValue(ui->listWidget->row(item) + 1);
    //2-nd case    //ui->spinBox->setValue(ui->listWidget->currentRow() + 1);
}
void MainWindow::on_spinBox_valueChanged(int arg1){
    if(ui->spinBox->value()>3) ui->listWidget->hide();
    else ui->listWidget->show();
}
```

# Проект формы. Visual Studio

The screenshot shows a Windows Forms application window titled "MyForm". The window contains a form with several controls and a settings panel on the right.

**Form Controls:**

- A label "Неизменяемый текст" (Immutable text) is highlighted in yellow.
- A text box containing the value "0".
- A dropdown menu showing the value "a".
- A list box containing the values "1", "2", and "3".
- A text box containing the value "1".

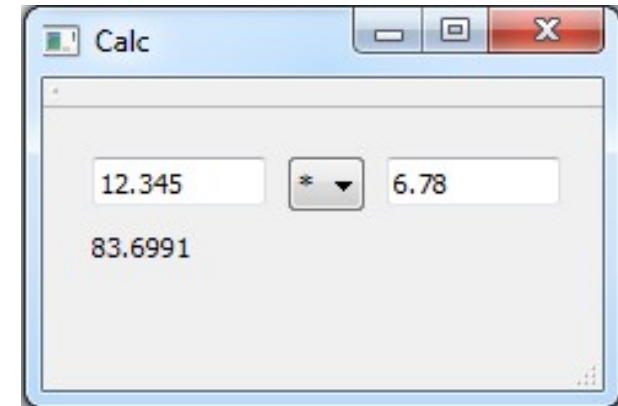
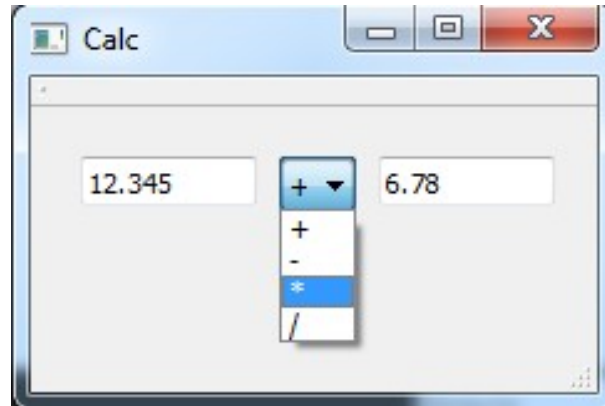
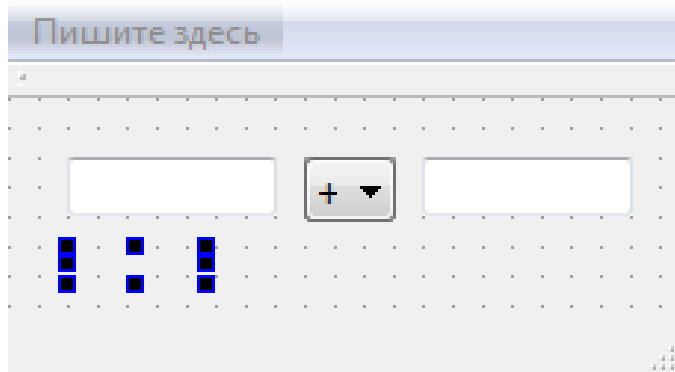
**Settings Panel (Right):**

- A button labeled "добавить-изменить" (Add-Change).
- Two radio buttons: "добавить слева" (Add left) and "изменить цвет" (Change color).
- Two checkboxes: "разблокировать слева" (Unlock left) and "разблокировать справа" (Unlock right).
- A button labeled "блокировка" (Locking).

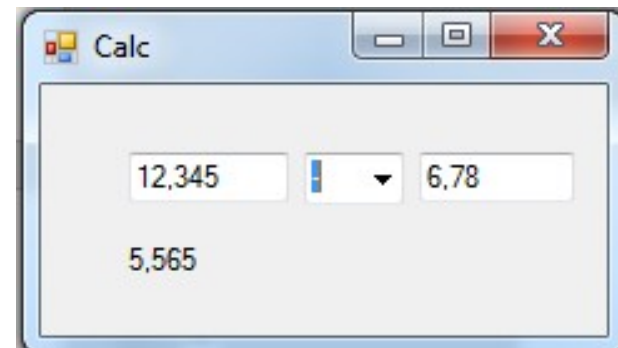
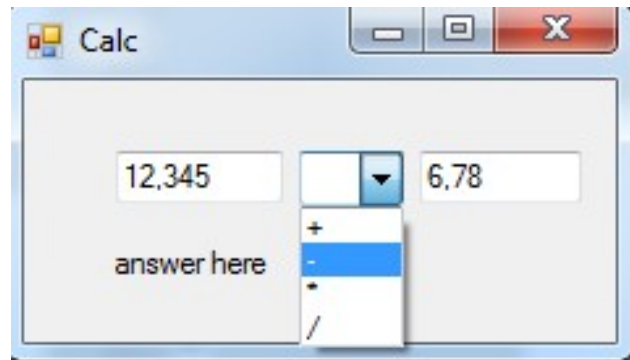
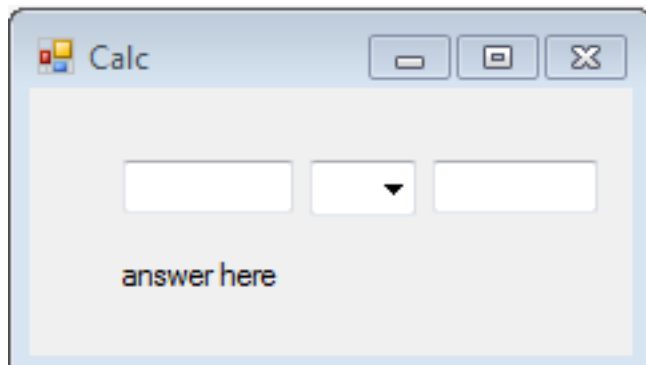
# Исходный код. Visual Studio

```
private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e) {
    if (checkBox1->Checked == true)
        textBox1->Enabled = ! textBox1->Enabled;
    if (checkBox2->Checked == true)
        this->comboBox1->Enabled = !this->comboBox1->Enabled;
}
private: System::Void comboBox1_SelectedIndexChanged(System::Object^ sender,
System::EventArgs^ e) {
    listBox1->Items->Insert(1, comboBox1->Text);
}
private: System::Void button2_Click(...) {
    if (radioButton1->Checked)
        textBox2->Text += Environment::NewLine + textBox1->Text;
    if(radioButton2->Checked)
        textBox1->BackColor=System::Drawing::Color::Green;
}
private: System::Void listBox1_MouseDoubleClick(...) {
    numericUpDown1->Value = listBox1->SelectedIndex + 1;
}
private: System::Void numericUpDown1_ValueChanged(...) {
    if (numericUpDown1->Value>3) listBox1->Visible = false;
    else listBox1->Visible = true;
}
```

# Приложение "Калькулятор"



QT  
точка



VS  
запятая

\* в настройках среды можно менять символ разделителя дробной и целой части, при это в разных средах он может иметь разные значения по умолчанию

# "Калькулятор" в Qt

```
void MainWindow::on_comboBox_activated(const QString &arg1)
{
    double a, b, c;
    a = ui->lineEdit->text().toDouble();
    b = ui->lineEdit_2->text().toDouble();
    QChar qSign = arg1[0];
    char sign = qSign.toLatin1();
    switch(sign){
        case '/': c = a / b; break;
        case '-': c = a - b; break;
        case '*': c = a * b; break;
        default : c = a + b; break;
    }
    ui->label->setText(QString::number(c));
}
```

# "Калькулятор" в VS

```
private: System::Void comboBox1_SelectedIndexChanged(System::Object^
sender, System::EventArgs^ e) {
    char sign = comboBox1->Text[0];
    double a, b, c;
    a = Convert::ToDouble(textBox1->Text);
    b = Convert::ToDouble(textBox2->Text);
    switch (sign) {
        case '+' : c = a + b; break;
        case '-' : c = a - b; break;
        case '*' : c = a * b; break;
        case '/' : c = a / b; break;
    }
    label3->Text = Convert::ToString(c);
}
```

# Добавление формы в Visual C++ (если не установлен шаблон Windows Forms для C++)

- создать пустой проект Visual C++ CLR
- добавить к проекту новый элемент UI -> Форма Windows Forms (Visual C++)
- отредактировать MyForm.cpp
- в свойствах проекта Компановщик->Система->Подсистема выбрать: SUBSYSTEM:WINDOWS
- в свойствах проекта Компановщик->Дополнительно->Точка входа записать: Main
- перед первой компиляцией конструктор формы закрыть и сохранить проект

# Содержимое MyForm.cpp

```
using namespace System;
using namespace System::Windows::Forms;
[STAThreadAttribute]
void Main(array<String^>^ args) {
    Application::EnableVisualStyles();
    Application::SetCompatibleTextRenderingDefault(false);
    Project::MyForm MyForm;
    Application::Run(%MyForm);
}
```

\*Project заменить на имя своего проекта, Form – на имя файла формы