



# Разработка ОКОННЫХ приложений с использованием WINDOWS FORMS

МОРИН ИЛЬЯ

# Содержание



**01**

## Что такое WF?

Общие сведения о технологии

**02**

## Понятие формы

Сведения о функционале форм

**03**

## Элементы управления

Функционал элементов управления

**04**

## События и Свойства

Настройка поведения элементов

**05**

## Visual Studio и WF

Функции упрощающие разработку

**06**

## Использование файлов

Воспроизведение мультимедиа

01

# ЧТО ТАКОЕ WINDOWS FORMS ?



„ О Windows, сын  
ошибок трудных!

*Автор Неизвестен*

01

# ЧТО ТАКОЕ WINDOWS FORMS ?



Windows Forms — это технология пользовательского интерфейса для .NET, представляющая собой набор управляемых библиотек, которые упрощают выполнение стандартных задач, таких как чтение из файловой системы и запись в нее.

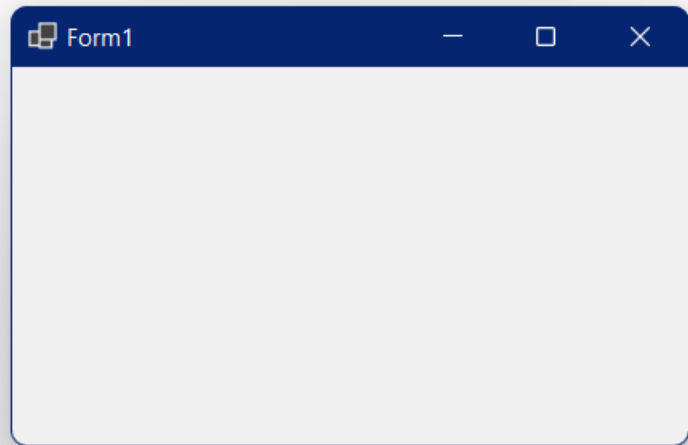
01

# ЧТО ТАКОЕ WINDOWS FORMS ?

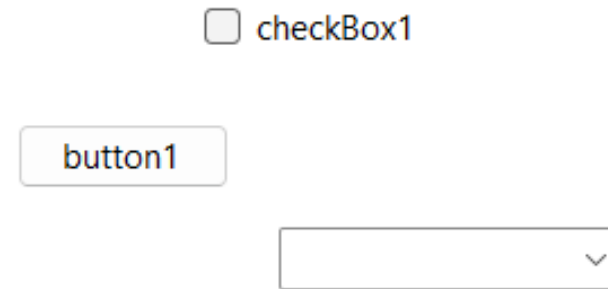


Приложение Windows Forms представляет собой событийно-ориентированное приложение, поддерживаемое Microsoft .NET Framework. Большая часть времени приложения тратится на ожидание от пользователя каких-либо действий, как, например, ввод текста в текстовое поле или клика мышкой по кнопке.

# Приложение WF включает в себя два основных элемента



**Форма**



**Элемент управления**

02

# ПОНЯТИЕ ФОРМЫ





# ПОНЯТИЕ ФОРМЫ



В Windows Forms форма — это визуальная поверхность, на которой выводится информация для пользователя. Обычно приложение Windows Forms строится путем добавления элементов управления в формы и создания кода для реагирования на действия пользователя, такие как щелчки мыши или нажатия клавиш.



# Каждая форма является экземпляром класса



```
namespace YourNamespace
{
    public partial class MainForm : Form // Все новые формы наследуются от Form
    {
        public MainForm() // Конструктор по умолчанию
        {
            InitializeComponent();
        }

        private void button_Click(object sender, EventArgs e) // Событие элемента управления
        {
        }
    }
}
```

# Одно приложение может иметь несколько форм



```
private void button_Click(object sender, EventArgs e)
{
    SecondForm secondForm = new SecondForm();
    if (true)
    {
        secondForm.Show(); // Этот вариант просто вызывает новую форму
        DialogResult result = secondForm.ShowDialog(); // Этот вариант вызовет форму,
                                                         // которая вернет результат работы
    }
}
```

03

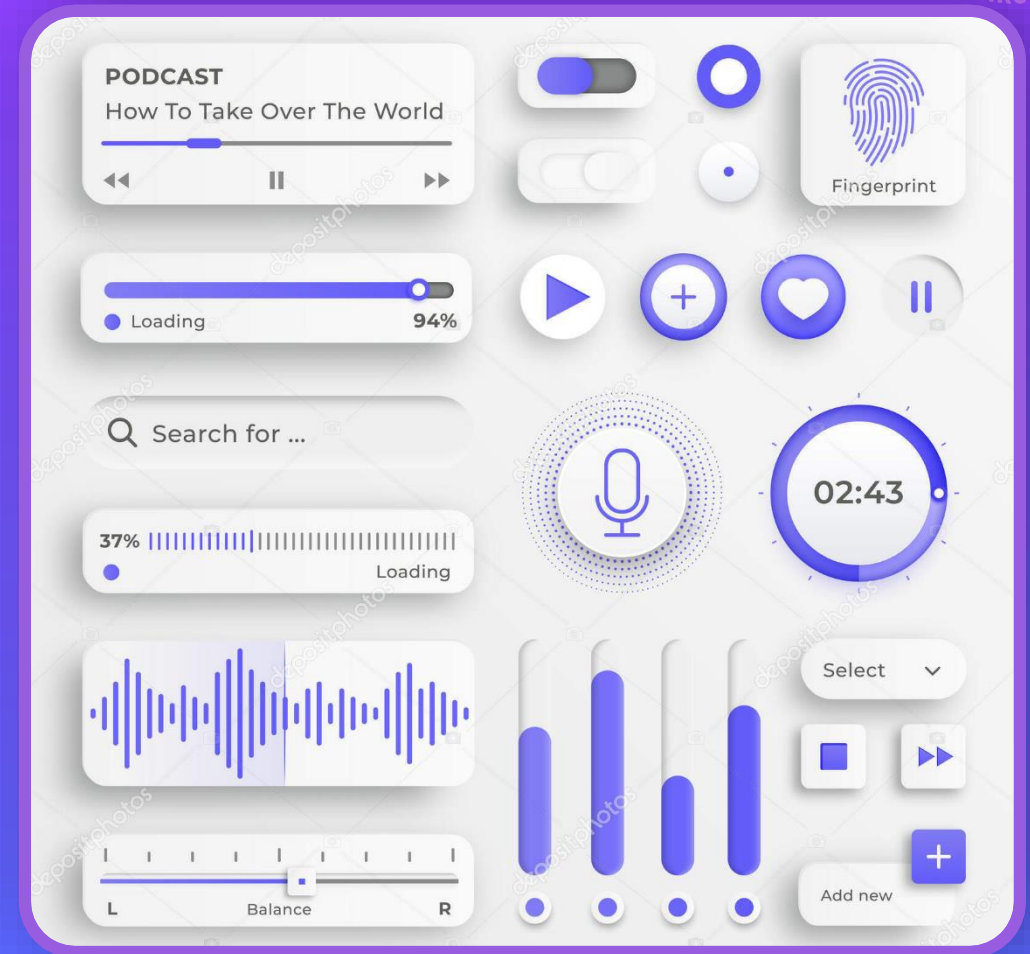
# ЭЛЕМЕНТЫ УПРАВЛЕНИЯ



# ЭЛЕМЕНТЫ УПРАВЛЕНИЯ



Элемент управления — это отдельный элемент пользовательского интерфейса, предназначенный для отображения или ввода данных.



# ЭЛЕМЕНТЫ УПРАВЛЕНИЯ



Элементы управления могут отображать текстовые поля, кнопки, раскрывающиеся списки, переключатели и даже веб-страницы. Если предусмотренные элементы управления не подходят для ваших целей, в Windows Forms можно создавать собственные пользовательские элементы управления с помощью класса Control.

# Код пользовательского элемента управления



```
public class AwesomeButton : Control // Элемент управления должен наследоваться от Control
{
    private Color borderColor; // Поля для хранения данных свойств

    [Category("Доп. настройки")] // Для свойств элемента управления
    [Description("Цвет, используемый для обводки")] // предусмотрены разные Атрибуты
    public Color BorderColor {}

    public AwesomeButton() {} // Конструктор по умолчанию элемента управления

    protected override void OnTextChanged(EventArgs e) // Событие изменения текста
    {
        base.OnTextChanged(e);
        Invalidate(); // Функция отвечает за обновление содержимого элемента
    }

    protected override void OnPaint(PaintEventArgs e) { } // Событие отрисовки - внешний вид элемента задается здесь
}
```

04

# СОБЫТИЯ И СВОЙСТВА





# СОБЫТИЯ И СВОЙСТВА



Каждый элемент управления WF имеет список событий, срабатывающих на определенные действия пользователя (нажатие клавишей мыши, перетаскивание и т.д.). Подписавшись на это событие можно написать функцию с определенной логикой, которая будет выполняться при выбранном действии на форме



```
this.button.Click += new System.EventHandler(this.button_Click); // Подписка на функцию (В Visual Studio делается через интерфейс конструктора)  
private void button_Click(object sender, EventArgs e) // Функция, подписанная на событие  
    {  
        SecondForm secondForm = new SecondForm();  
        secondForm.Show(); // Этот вариант просто вызывает новую форму  
        DialogResult result = secondForm.ShowDialog(); // Этот вариант вызовет форму,  
                                                // которая вернет результат работы  
    }
```

# СОБЫТИЯ И СВОЙСТВА



Свойства являются способом редактирование элементов управления и форм. Свойства можно менять в коде программы, а также с помощью конструктора Visual Studio

```
private void InitializeComponent() // Функция инициализации, используемая в конструкторе формы
{
    this.button = new System.Windows.Forms.Button(); // Создание элемента кнопки

    this.button.Location = new System.Drawing.Point(196, 147); // Здесь и далее задаются свойства для кнопки
    this.button.Name = "button";
    this.button.Size = new System.Drawing.Size(94, 29);
    this.button.TabIndex = 0;
    this.button.Text = "button1";
    this.button.UseVisualStyleBackColor = true;
    this.button.Click += new System.EventHandler(this.button_Click); // Подписка на событие нажатия
}
```

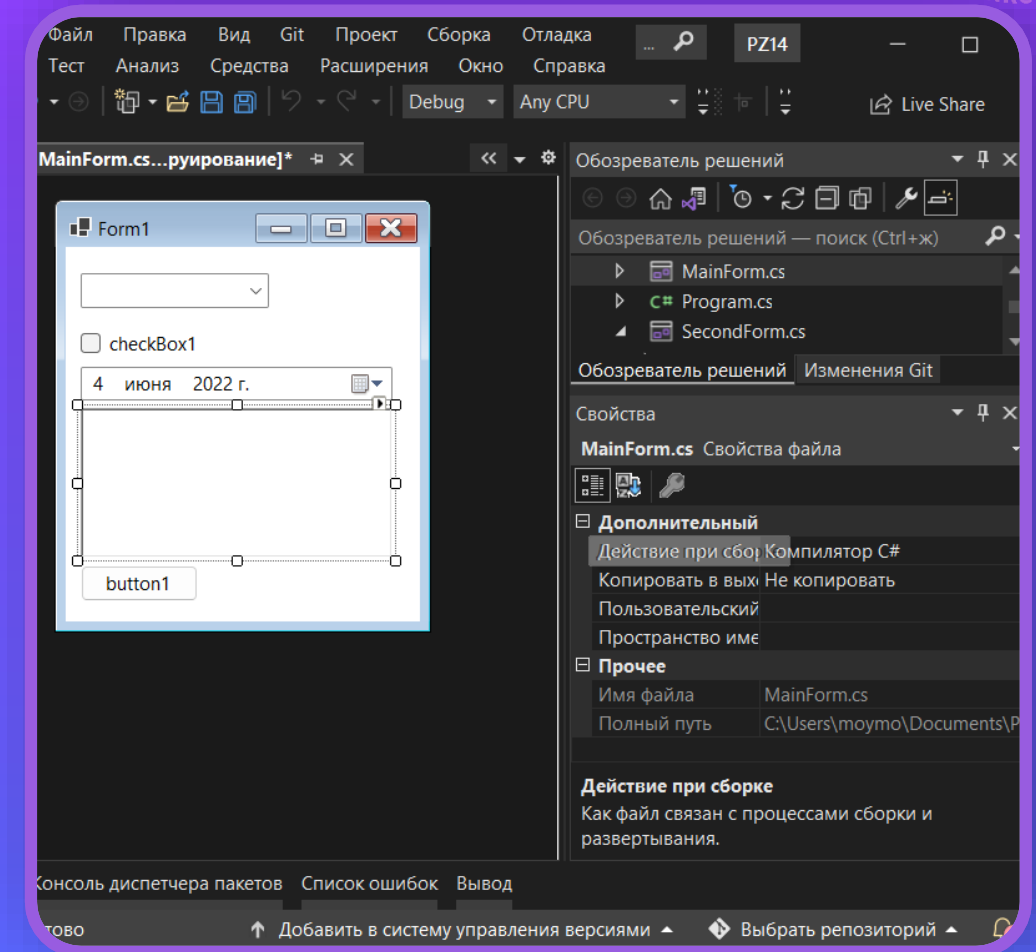
05

# VISUAL STUDIO & WINDOWS FORMS

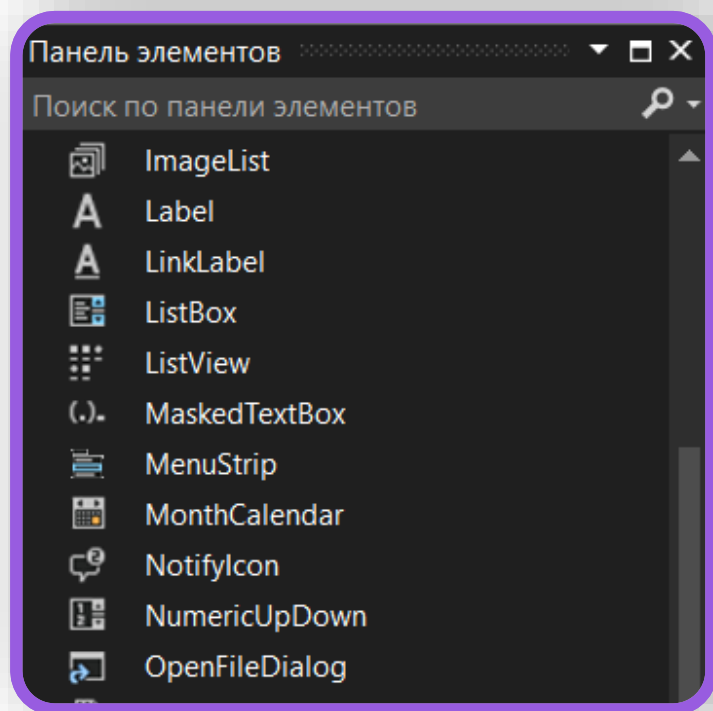


# VS & WF

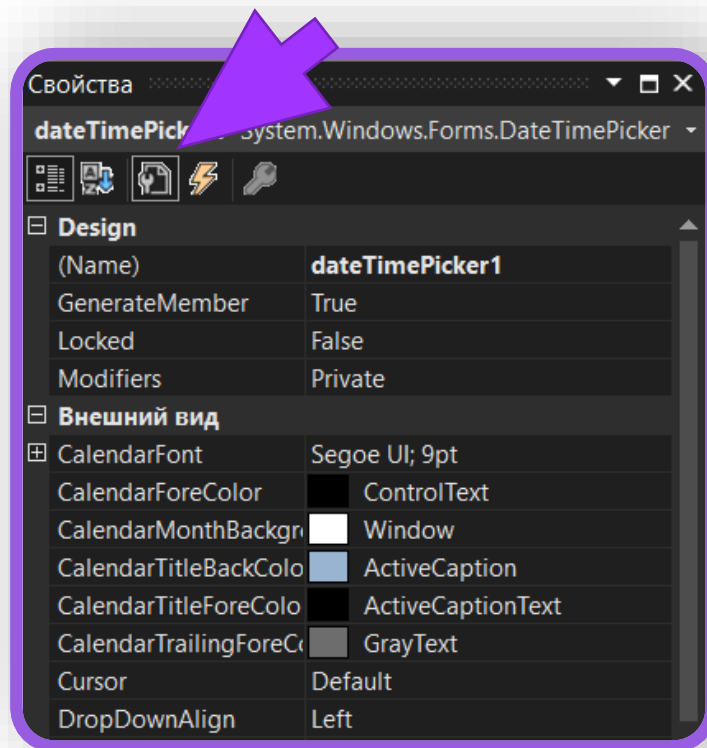
Visual Studio позволяет взаимодействовать с формой в интерактивном режиме т.е. расставлять элементы и менять свойства с помощью интерфейса редактора



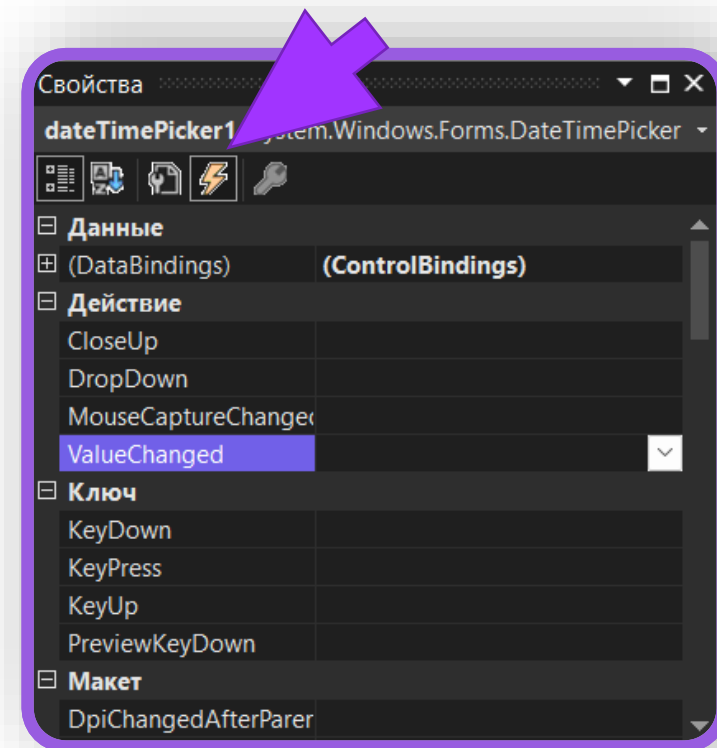
# Основные панели для работы с WF



Панель элементов  
Находится по умолчанию  
слева в главном окне



Окно свойств с режимом  
отображением свойств  
элемента управления



Окно свойств с режимом  
отображением событий  
элемента управления

05

# ИСПОЛЬЗОВАНИЕ ФАЙЛОВ



# ИСПОЛЬЗОВАНИЕ ФАЙЛОВ



Кроме элементов управления форма также может воспроизводить файлы изображений, звуков и прочего

Мы рассмотрим работу с изображением и звуками.

# ИСПОЛЬЗОВАНИЕ ФАЙЛОВ



```
private void ImageExampleForm_Paint(object sender, PaintEventArgs e)
{
    Image newImage = Image.FromFile("SampImag.jpg");

    Point ulCorner = new Point(100, 100); // Здесь задается координата верхнего левого угла изображения

    e.Graphics.DrawImage(newImage, ulCorner);
}
```

```
private void playSimpleSound( )
{
    SoundPlayer simpleSound = new SoundPlayer(@"c:\Windows\Media\chimes.wav");
    simpleSound.Play();
}
```





СПАСИБО ЗА  
ВНИМАНИЕ



Работу выполнил  
студент группы 940

Илья Морин