

Шпаргалка по теме: Введение в Windows Forms.

1. Структура проекта Windows Forms

Проект Windows Forms состоит из нескольких ключевых компонентов:

1. Файл Program.cs

Точка входа в приложение. Содержит метод Main, который запускает главную форму.

Пример:

```
static class Program
{
    [STAThread]
    static void Main()
    {
        Application.EnableVisualStyles();
        Application.SetCompatibleTextRenderingDefault(false);
        Application.Run(new MainForm()); // Запуск главной формы
    }
}
```

2. Файл формы (Form1.cs)

Содержит код формы (логику и обработчики событий).

1. Form1.Designer.cs — автоматически генерируемый файл, содержащий код для инициализации элементов управления.
2. Form1.resx — файл ресурсов формы (например, строки, изображения).

3. Файл App.config

Конфигурационный файл приложения. Может содержать настройки, строки подключения и т.д.

4. Папка Resources

Хранит ресурсы приложения (изображения, иконки, строки).

5. Папка Properties

Содержит файл AssemblyInfo.cs с метаданными сборки (версия, автор и т.д.).

2. Основные элементы управления Windows Forms

Элементы управления — это компоненты, которые используются для создания пользовательского интерфейса.

Элемент управления	Описание	Пример использования
Button	Кнопка для выполнения действий.	button1.Text = "Нажми меня"; button1.Click += Button1_Click;
Label	Текстовое поле для отображения информации.	label1.Text = "Привет, мир!";
TextBox	Поле для ввода текста.	textBox1.Text = "Введите текст";
RichTextBox	Расширенное текстовое поле с поддержкой форматирования.	richTextBox1.Text = "Привет, мир!"; richTextBox1.Font = new Font("Arial", 12);
MessageBox	Диалоговое окно для отображения сообщений.	MessageBox.Show("Сообщение", "Заголовок", MessageBoxButtons.OK, MessageBoxIcon.Info);

Пример использования элементов управления:

```
public partial class MainForm : Form
{
    public MainForm()
    {
        InitializeComponent();

        // Настройка элементов управления
        button1.Text = "Нажми меня";
        label1.Text = "Привет, мир!";
        textBox1.Text = "Введите текст";
        comboBox1.Items.AddRange(new string[] { "Вариант 1", "Вариант 2" });
    }

    private void button1_Click(object sender, EventArgs e)
    {
        label1.Text = textBox1.Text;
    }
}
```

5. Архитектура приложения Windows Forms

Архитектура приложения Windows Forms обычно строится на основе **модели событий** (event-driven). Основные компоненты:

1. Форма (Form)

Основное окно приложения. Содержит элементы управления и логику обработки событий.

2. Элементы управления (Controls)

Компоненты, которые взаимодействуют с пользователем (кнопки, текстовые поля и т.д.).

3. События (Events)

Действия, которые происходят в приложении (например, нажатие кнопки, изменение текста). Обрабатываются с помощью методов-обработчиков.

4. Модель данных (Data Model)

Классы, которые представляют данные приложения. Могут быть связаны с элементами управления через привязку данных (Data Binding).

6. Настройка свойств элементов управления

Каждый элемент управления в Windows Forms имеет набор свойств, которые можно настроить для изменения его внешнего вида и поведения. Свойства можно задавать как в дизайнере форм, так и программно в коде.

Основные свойства элементов управления:

Свойство	Описание	Пример
Text	Текст, отображаемый на элементе.	button1.Text = "Нажми меня";
Font	Шрифт текста.	label1.Font = new Font("Arial", 12);
ForeColor	Цвет текста.	label1.ForeColor = Color.Red;
BackColor	Цвет фона элемента.	textBox1.BackColor = Color.Yellow;

Size	Размер элемента (ширина и высота).	button1.Size = new Size(100, 50);
Location	Позиция элемента на форме (X, Y).	button1.Location = new Point(10, 10);
Enabled	Включен или отключен элемент.	button1.Enabled = false;

Пример настройки свойств программно:

```
public partial class MainForm : Form
{
    public MainForm()
    {
        InitializeComponent();

        // Настройка свойств элементов
        button1.Text = "Нажми меня";
        button1.Font = new Font("Arial", 12);
        button1.ForeColor = Color.White;
        button1.BackColor = Color.Blue;
        button1.Size = new Size(150, 50);
        button1.Location = new Point(50, 50);

        label1.Text = "Привет, мир!";
        label1.Font = new Font("Arial", 14, FontStyle.Bold);
        label1.ForeColor = Color.Green;

        textBox1.Multiline = true;
        textBox1.Size = new Size(200, 100);
        textBox1.BackColor = Color.LightYellow;
    }
}
```

7. Обработка событий

События — это действия, которые происходят в приложении (например, нажатие кнопки, изменение текста, перемещение мыши). Обработка событий позволяет реагировать на эти действия.

Основные события элементов управления:

Событие	Описание	Пример
Click	Происходит при нажатии на элемент.	button1.Click += Button1_Click;
TextChanged	Происходит при изменении текста в TextBox.	textBox1.TextChanged += TextBox1_TextChanged;
CheckedChanged	Происходит при изменении состояния CheckBox или RadioButton.	checkBox1.CheckedChanged += CheckBox1_CheckedChanged;
SelectedIndexChanged	Происходит при изменении выбранного элемента в ComboBox или ListBox.	comboBox1.SelectedIndexChanged += ComboBox1_SelectedIndexChanged;
MouseEnter	Происходит при наведении курсора на элемент.	button1.MouseEnter += Button1_MouseEnter;
MouseLeave	Происходит при выходе курсора за пределы элемента.	button1.MouseLeave += Button1_MouseLeave;

KeyPress	Происходит при нажатии клавиши, когда элемент в фокусе.	textBox1.KeyPress += TextBox1_KeyPress;
FormClosing	Происходит при закрытии формы.	this.FormClosing += MainForm_FormClosing;

Пример обработки событий:

```

public partial class MainForm : Form
{
    public MainForm()
    {
        InitializeComponent();

        // Подписка на события
        button1.Click += Button1_Click;
        textBox1.TextChanged += TextBox1_TextChanged;
        checkBox1.CheckedChanged += CheckBox1_CheckedChanged;
        comboBox1.SelectedIndexChanged += ComboBox1_SelectedIndexChanged;
        button1.MouseEnter += Button1_MouseEnter;
        button1.MouseLeave += Button1_MouseLeave;
        textBox1.KeyPress += TextBox1_KeyPress;
        this.FormClosing += MainForm_FormClosing;
    }

    // Обработчик нажатия кнопки
    private void Button1_Click(object sender, EventArgs e)
    {
        MessageBox.Show("Кнопка нажата!");
    }

    // Обработчик изменения текста
    private void TextBox1_TextChanged(object sender, EventArgs e)
    {
        label1.Text = textBox1.Text;
    }

    // Обработчик изменения состояния CheckBox
    private void CheckBox1_CheckedChanged(object sender, EventArgs e)
    {
        if (checkBox1.Checked)
            MessageBox.Show("CheckBox отмечен");
        else
            MessageBox.Show("CheckBox снят");
    }

    // Обработчик изменения выбранного элемента в ComboBox
    private void ComboBox1_SelectedIndexChanged(object sender, EventArgs e)
    {
        MessageBox.Show($"Выбран элемент: {comboBox1.SelectedItem}");
    }
}

```

```

// Обработчик наведения курсора на кнопку
private void Button1_MouseEnter(object sender, EventArgs e)
{
    button1.BackColor = Color.Red;
}

// Обработчик выхода курсора за пределы кнопки
private void Button1_MouseLeave(object sender, EventArgs e)
{
    button1.BackColor = Color.Blue;
}

// Обработчик нажатия клавиши в TextBox
private void TextBox1_KeyPress(object sender, KeyPressEventArgs e)
{
    if (e.KeyChar == (char)Keys.Enter)
    {
        MessageBox.Show("Нажата клавиша Enter");
    }
}

// Обработчик закрытия формы
private void MainForm_FormClosing(object sender, FormClosingEventArgs e)
{
    DialogResult result = MessageBox.Show("Вы уверены, что хотите закрыть форму?", "Подтверждение", MessageBoxButtons.YesNo);
    if (result == DialogResult.No)
    {
        e.Cancel = true; // Отмена закрытия формы
    }
}

```

Итоговая таблица

Компонент	Описание	Пример
Свойства элементов	Настройка внешнего вида и поведения элементов.	button1.Text = "Нажми меня";
События элементов	Действия, которые обрабатываются в приложении.	button1.Click += Button1_Click;
Обработчики событий	Методы, которые выполняются при возникновении события.	private void Button1_Click(object sender, EventArgs e) { ... }