

Шпаргалка по теме "Контейнеры и управление расположением компонентов в Windows Forms"

Контейнеры — это специальные элементы управления, которые группируют и организуют другие элементы интерфейса (кнопки, текстовые поля и т. д.).

Основные задачи контейнеров:

- ✓ Группировка элементов управления.
- ✓ Управление расположением элементов.
- ✓ Создание сложных и адаптивных интерфейсов.
- ✓ Упрощение управления элементами.

Основные контейнеры в Windows Forms

1. Panel – Простая панель

- Используется для группировки элементов без визуальных эффектов.
- Поддерживает AutoScroll для прокрутки.
- Может содержать другие контейнеры.

Основные свойства:

- BackColor — цвет фона.
- BorderStyle — стиль границы (None, FixedSingle, Fixed3D).
- AutoScroll — включение/выключение прокрутки.

◆ Пример:

```
Button btn = new Button();
btn.Text = "Кнопка";
btn.Padding = new Padding(10); // Внутренний отступ
btn.Margin = new Padding(5);   // Внешний отступ
this.Controls.Add(btn);
```

2. GroupBox – Группа с заголовком

- Логически объединяет связанные элементы управления.
- Визуально выделяет группу с заголовком.

Основные свойства:

- Text — заголовок группы.
- FlatStyle — стиль рамки (Standard, Flat, Popup, System).

◆ Пример:

```
// Создаем GroupBox
GroupBox groupBox = new GroupBox();
groupBox.Text = "Выберите цвет:"; // Устанавливаем заголовок
groupBox.Size = new Size(200, 100); // Устанавливаем размер

// Добавляем на форму
this.Controls.Add(groupBox);
```

3. TabControl – Вкладки

- Позволяет размещать элементы на отдельных вкладках.
- Удобен для создания настроек и многосекционных форм.

Основные свойства:

- TabPages — коллекция вкладок.
- SelectedIndex — индекс активной вкладки.

◆ Пример:

```
// Создаем TabControl
TabControl tabControl = new TabControl();
tabControl.Size = new Size(300, 200); // Устанавливаем размер

// Создаем вкладки
TabPage tabPage1 = new TabPage("Общие");
TabPage tabPage2 = new TabPage("Настройки");

// Добавляем вкладки в TabControl
tabControl.TabPages.Add(tabPage1);
tabControl.TabPages.Add(tabPage2);

// Добавляем TabControl на форму
this.Controls.Add(tabControl);
```

4. FlowLayoutPanel – Автоматическое расположение элементов.

- Размещает элементы в строку или столбец.
- Автоматически переносит элементы на новую строку при нехватке места.
- Полезен для создания адаптивных интерфейсов.

Основные свойства:

- FlowDirection — направление потока (LeftToRight, RightToLeft, TopDown, BottomUp).
- WrapContents — перенос элементов на новую строку/столбец.

◆ Пример:

```
// Создаем FlowLayoutPanel
FlowLayoutPanel flowPanel = new FlowLayoutPanel();
flowPanel.FlowDirection = FlowDirection.LeftToRight; // Элементы идут слева направо
flowPanel.WrapContents = true; // Включаем перенос на новую строку
flowPanel.Size = new Size(300, 100); // Устанавливаем размер

// Создаем кнопки
Button btn1 = new Button() { Text = "Кнопка 1" };
Button btn2 = new Button() { Text = "Кнопка 2" };

// Добавляем кнопки в FlowLayoutPanel
flowPanel.Controls.Add(btn1);
flowPanel.Controls.Add(btn2);

// Добавляем FlowLayoutPanel на форму
this.Controls.Add(flowPanel);
```

5. TableLayoutPanel – Табличное расположение

- Размещает элементы в таблице с заданным количеством строк и столбцов.
- Поддерживает растягивание элементов по ячейкам.

Основные свойства:

- RowCount — количество строк.
- ColumnCount — количество столбцов.
- RowStyles и ColumnStyles — параметры стилей строк и столбцов.

◆ Пример:

```
// Создаем TableLayoutPanel
TableLayoutPanel tablePanel = new TableLayoutPanel();
tablePanel.RowCount = 2; // Количество строк
tablePanel.ColumnCount = 2; // Количество столбцов
tablePanel.Size = new Size(200, 100); // Устанавливаем размер
tablePanel.Dock = DockStyle.Fill; // Заполняем доступное пространство

// Создаем кнопки
Button btn1 = new Button() { Text = "1" };
Button btn2 = new Button() { Text = "2" };
Button btn3 = new Button() { Text = "3" };
Button btn4 = new Button() { Text = "4" };

// Размещаем кнопки в ячейках таблицы
tablePanel.Controls.Add(btn1, 0, 0);
tablePanel.Controls.Add(btn2, 1, 0);
tablePanel.Controls.Add(btn3, 0, 1);
tablePanel.Controls.Add(btn4, 1, 1);

// Добавляем TableLayoutPanel на форму
this.Controls.Add(tablePanel);
```

6. SplitContainer – Разделение области

- Делит форму на две независимые части, между которыми можно изменять размеры.

Основные свойства:

- Orientation — ориентация разделения (Horizontal или Vertical).
- Panel1 и Panel2 — две разделённые области.

◆ Пример:

```
// Создаем SplitContainer
SplitContainer splitContainer = new SplitContainer();
splitContainer.Dock = DockStyle.Fill; // Растягиваем на всю форму
splitContainer.Orientation = Orientation.Vertical; // Вертикальное разделение

// Задаем цвет фона для наглядности
splitContainer.Panel1.BackColor = Color.LightBlue;
splitContainer.Panel2.BackColor = Color.LightGreen;

// Добавляем SplitContainer на форму
this.Controls.Add(splitContainer);
```

Свойства для управления расположением элементов

1. Свойство Dock

Описание: закрепляет элемент управления относительно краёв контейнера.

Значения:

- None — не закреплено (по умолчанию).
- Top — закрепление к верхнему краю.
- Bottom — закрепление к нижнему краю.
- Left — закрепление к левому краю.
- Right — закрепление к правому краю.
- Fill — заполнение всего доступного пространства.

Пример:

```
Button btn = new Button();
btn.Text = "Заполнить";
btn.Dock = DockStyle.Fill; // Растягиваем кнопку на всю доступную область
this.Controls.Add(btn);
```

2. Свойство Anchor

Описание: привязывает элемент управления к указанным краям контейнера.

Значения:

Можно привязать к любому из краёв: Top, Bottom, Left, Right.

Пример:

```
TextBox textBox = new TextBox();
textBox.Anchor = AnchorStyles.Top | AnchorStyles.Left | AnchorStyles.Right;
// Элемент изменяет ширину при изменении размеров окна
this.Controls.Add(textBox);
```

3. Свойство AutoSize

Описание: Автоматически изменяет размеры элемента управления в зависимости от его содержимого.

Пример:

```
Label label = new Label();
label.Text = "Длинный текст";
label.AutoSize = true; // Автоматически подстраивается под текст
this.Controls.Add(label);
```

4. Свойства Padding и Margin

Описание:

- Padding — отступы внутри элемента (между содержимым и границей).
- Margin — отступы вокруг элемента (между элементом и другими объектами).

Пример:

```
Button btn = new Button();
btn.Text = "Кнопка";
btn.Padding = new Padding(10); // Внутренний отступ
btn.Margin = new Padding(5); // Внешний отступ
this.Controls.Add(btn);
```