

Шпаргалка по теме: C#. Windows Forms. Основные понятия ООП.

1. Класс

Класс — это шаблон для создания объектов. Он определяет структуру и поведение объектов.

Пример:

```
public class Car
{
    // Поля
    private string brand;
    private int speed;

    // Свойства
    public string Brand
    {
        get { return brand; }
        set { brand = value; }
    }

    public int Speed
    {
        get { return speed; }
        set { speed = value; }
    }

    // Метод
    public void Accelerate(int increment)
    {
        speed += increment;
    }
}
```

2. Объект

Объект — это экземпляр класса. Он создается с помощью ключевого слова **new**.

Пример:

```
Car myCar = new Car();
myCar.Brand = "Toyota";
myCar.Speed = 60;
myCar.Accelerate(20); // Увеличиваем скорость на 20
```

3. Поля и свойства

- **Поля** — переменные, объявленные внутри класса. Они хранят данные объекта.
- **Свойства** — предоставляют контролируемый доступ к полям. Они могут иметь логику в **get** и **set**.

Пример:

```
public class Person
{
    // Поле
    private string name;

    // Свойство
    public string Name
    {
        get { return name; }
        set
        {
            if (!string.IsNullOrEmpty(value))
                name = value;
        }
    }
}
```

4. Модификаторы доступа

Модификаторы доступа определяют видимость членов класса:

- **public** — доступен из любого места.
- **private** — доступен только внутри класса.
- **protected** — доступен внутри класса и в производных классах.
- **internal** — доступен в пределах одной сборки.
- **static** — принадлежит классу, а не объекту. Статические члены общие для всех экземпляров класса.

Пример:

```
public class Animal
{
    private string sound; // Доступен только внутри класса
    public string Name { get; set; } // Доступен везде

    protected void MakeSound()
    {
        Console.WriteLine(sound);
    }
}
```

```
public class Animal
{
    private string sound; // Доступен только внутри класса
    public string Name { get; set; } // Доступен везде

    protected void MakeSound()
    {
        Console.WriteLine(sound);
    }

    // Статическое поле
    public static int AnimalCount = 0;

    // Статический метод
    public static void DisplayCount()
    {
        Console.WriteLine($"Total animals: {AnimalCount}");
    }
}
```

5. Методы

Методы — это функции, объявленные внутри класса. Они определяют поведение объекта.

Пример:

```
public class Calculator
{
    // Метод сложения
    public int Add(int a, int b)
    {
        return a + b;
    }

    // Метод вычитания
    public int Subtract(int a, int b)
    {
        return a - b;
    }
}
```

Использование:

```
Calculator calc = new Calculator();
int result = calc.Add(5, 3); // Результат: 8
```

6. Конструкторы

Конструктор — это специальный метод, который вызывается при создании объекта. Он используется для инициализации полей.

Пример:

```
public class Book
{
    public string Title { get; set; }
    public string Author { get; set; }

    // Конструктор по умолчанию
    public Book()
    {
        Title = "Unknown";
        Author = "Unknown";
    }

    // Параметризованный конструктор
    public Book(string title, string author)
    {
        Title = title;
        Author = author;
    }
}
```

Использование:

```
Book book1 = new Book(); // Используется конструктор по умолчанию
Book book2 = new Book("1984", "George Orwell"); // Используется параметризованный конструктор
```

Пример использования в Windows Forms

Класс для работы с формой

```
public partial class MainForm : Form
{
    // Поле
    private int counter = 0;

    // Конструктор
    public MainForm()
    {
        InitializeComponent();
    }

    // Метод для обработки нажатия кнопки
    private void button1_Click(object sender, EventArgs e)
    {
        counter++;
        label1.Text = $"Количество нажатий: {counter}";
    }
}
```

Как это работает:

- Создается форма **MainForm**.
- При нажатии на кнопку **button1** увеличивается значение поля **counter**.
- Текст метки **label1** обновляется, отображая количество нажатий.

Итоговая таблица

| Понятие | Описание | Пример |
|---------------------|---|---|
| Класс | Шаблон для создания объектов. | <code>public class Car { ... }</code> |
| Объект | Экземпляр класса. | <code>Car myCar = new Car();</code> |
| Поля | Переменные, хранящие данные объекта. | <code>private string brand;</code> |
| Свойства | Контролируемый доступ к полям. | <code>public string Brand { get; set; }</code> |
| Модификаторы | Управляют видимостью членов класса. | <code>public, private, protected, internal</code> |
| Методы | Функции, определяющие поведение объекта. | <code>public void Accelerate(int increment) { ... }</code> |
| Конструкторы | Специальные методы для инициализации объекта. | <code>public Book(string title, string author) { ... }</code> |