

# C# Essential

Статические и вложенные классы

# C# Essential

После урока обязательно



Повторите этот урок в видео формате на [ITVDN.com](http://itvdn.com)



Проверьте как Вы усвоили данный материал  
на [TestProvider.com](http://testprovider.com)

# Статические и вложенные классы

# Статические члены

## Статические переменные

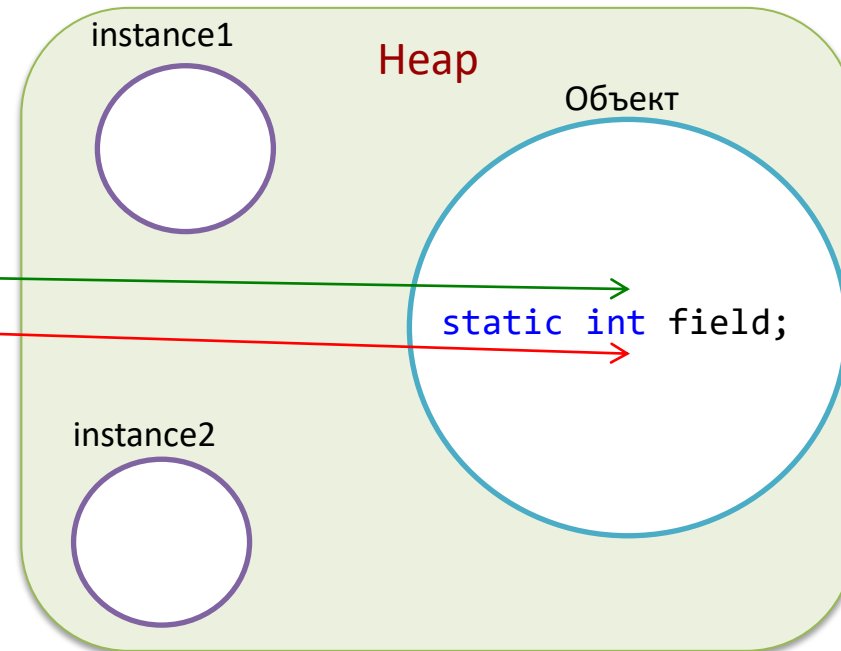
**Статическая переменная** – это общая переменная для всех экземпляров класса, которая хранится в объекте.

Объекты содержат в себе статические поля и методы.

```
static void Main()
{
    MyClass instance1 = new MyClass();
    MyClass instance2 = new MyClass();

    MyClass.field = 2;
    MyClass.field = 5;
}
```

```
class MyClass
{
    public static int field;
}
```



# Статические классы

## Константы

Константа не может быть объявлена как `static`, по сколько по своему поведению, уже является статической.

```
public const float e = 2.71828182845904523536f;
```

Поле `const` относится к типу, а не к экземплярам типа. Поэтому к полям `const` можно обращаться с использованием той же нотации **ИмяКласса.ИмяЧлена**, что и в используемой для статических полей.

# Статические методы и свойства

## Static method and field

**Статическими могут быть методы и свойства:**

```
static void Main()  
{  
    Console.WriteLine("static Method");  
}
```

```
public static int Property  
{  
    get { return field; }  
    set { field = value; }  
}
```



Статические методы не могут обращаться к нестатическим полям.  
Статические члены не могут быть виртуальными, переопределенными и абстрактными.

# Статические конструкторы

## Static Constructors

Классы и статические классы могут иметь статические конструкторы.

```
class MyClass
{
    public static int field;

    static MyClass()
    {
        field = 10;
    }
}
```



Статический конструктор всегда отработывает первым.

# Статические конструкторы

## Свойства статического конструктора

- Статический конструктор не имеет модификаторов доступа и не принимает параметров.
- Статический конструктор вызывается автоматически для инициализации класса перед созданием первого экземпляра или ссылкой на какие-либо статические члены.
- Статический конструктор нельзя вызывать напрямую.
- Пользователь не управляет тем, когда статический конструктор выполняется в программе.
- Типичным использованием статических конструкторов является случай, когда класс использует файл журнала и конструктор применяется для добавления записей в этот файл.



# Методы расширения

## Extension Methods

Методы расширения позволяют "**добавлять**" методы в существующие типы без создания нового производного типа, перекомпиляции или иного изменения исходного типа.

Расширяющие методы могут быть только статическими и создаваться только в статических классах.

```
static class MyClass
{
    public static void Method(this string value)
    {
        Console.WriteLine(value);
    }
}
```

```
static void Main()
{
    string text = "Тестовая строка";

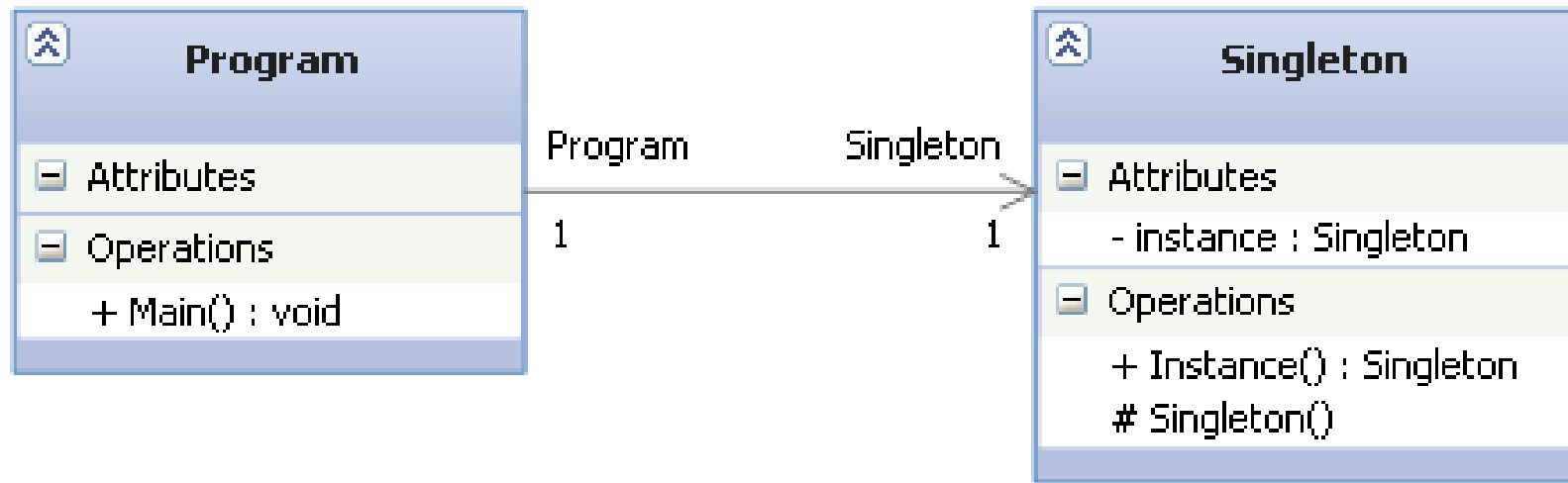
    MyClass.Method(text);
    text.Method();
}
```



Аргумент расширения всегда должен быть только один и стоять первым в списке аргументов.

# Шаблон проектирования

## Паттерн Singleton



Паттерн Singleton гарантирует, что у класса есть только один экземпляр, и предоставляет к нему глобальную точку доступа.

# Вложенные классы

## Nested Classes

Тип, определенный внутри класса или структуры, называется вложенным типом.

Экземпляр внутреннего класса не может существовать без привязки к включающему его классу верхнего уровня.

```
class Container
{
    class Nested
    {
        int field;
    }
}
```

Статические классы могут в себе содержать нестатические **Nested** классы.

# C# Essential

Q&A

# Информационный видеосервис для разработчиков программного обеспечения

