



# C# Essential

Коллекции



# C# Essential

Автор курса



Александр Шевчук  
MCT



MCID: 9230440

# C# Essential

После урока обязательно



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

Доступ можно получить через руководство вашего учебного центра



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

# C# Essential

Тема

Коллекции

# Коллекция

## Collection

**Коллекция** – это класс, предназначенный для группировки связанных объектов, управления ими и обработки их в циклах.



Коллекции являются важным инструментом программиста, но решение о их применении не всегда оказывается очевидным.

# Применение коллекций

## Use of Collections

**Коллекции стоит применять, если:**

- Отдельные элементы используются для одинаковых целей и одинаково важны.
- На момент компиляции число элементов не известно или не зафиксировано.
- Необходима поддержка операции перебора всех элементов.
- Необходима поддержка упорядочивания элементов.
- Необходимо использовать элементы из библиотеки, от которой потребитель ожидает наличия типа коллекции.

# Интерфейс

## IEnumerable

Методы интерфейса **IEnumerable**:

**IEnumerator** GetEnumerator() – возвращает перечислитель, который можно использовать для навигации по коллекции.

# Интерфейс

## IEnumerator

Свойства интерфейса **IEnumerator**:

`object` Current { `get`; } – возвращает текущий элемент коллекции.

Методы интерфейса **IEnumerator**:

`bool` MoveNext() – перемещает перечислитель на следующий элемент коллекции.

`void` Reset() – возвращает перечислитель на начало коллекции.



# Ключевое слово

## yield

- Блок, в котором содержится ключевое слово `yield`, расценивается компилятором, как блок итератора.
- Ключевое слово `return` используется для предоставления значения объекту перечислителя.
- Ключевое слово `break` используется для обозначения конца итерации.

```
public static IEnumerable Power()  
{  
    yield return "Hello world!";  
}
```

```
public static IEnumerable Power()  
{  
    yield break;  
}
```

# Циклическая конструкция

## foreach

Циклическая конструкция **foreach** позволяет выполнять навигацию по коллекции, используя реализации интерфейсов **IEnumerable** и **IEnumerator**.

```
foreach (var element in myCollection)
{
}
```



**var** — локальная переменная с неявным типом имеет строгую типизацию, как если бы тип был задан явно, только тип определяет компилятор.

# Интерфейс

## IEnumerable<T>

**IEnumerable<T>** – унаследован от **IEnumerable**

Методы интерфейса **IEnumerable<T>**:

**IEnumerator<T>** GetEnumerator() – возвращает обобщенный перечислитель, который можно использовать для навигации по коллекции.

# Смотрите наши уроки в видео формате

ITVDN.com



Посмотрите этот урок в видео формате на образовательном портале [ITVDN.com](http://itvdn.com) для закрепления пройденного материала.

Все курсы записаны сертифицированными тренерами, которые работают в учебном центре CyberBionic Systematics



# Проверка знаний

TestProvider.com



TestProvider – это online сервис проверки знаний по информационным технологиям. С его помощью Вы можете оценить Ваш уровень и выявить слабые места. Он будет полезен как в процессе изучения технологии, так и общей оценки знаний IT специалиста.

После каждого урока проходите тестирование для проверки знаний на [TestProvider.com](http://TestProvider.com)

Успешное прохождение финального тестирования позволит Вам получить соответствующий Сертификат.



# C# Essential

Q&A

# Информационный видеосервис для разработчиков программного обеспечения

