



C# Essential

Коллекции

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

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.

C# Essential

Q&A

Информационный видеосервис для разработчиков программного обеспечения

