

Коллекции

№ урока: 14 Курс: C# Essential

Средства обучения: Компьютер с установленной Visual Studio

Обзор, цель и назначение урока

Рассмотрение коллекций.
Рассмотрение оператора `yield`.

Изучив материал данного занятия, учащийся сможет:

- Понимать работу коллекций.
- Понимать работу оператора `yield`.

Содержание урока

1. Коллекции.
2. Интерфейсы `IEnumerable` и `IEnumerator`.
3. Работа цикла `foreach`.
4. Оператор `yield`.
5. Интерфейсы `IEnumerable<T>` и `IEnumerator<T>`.

Резюме

- Коллекция представляет собой набор объектов схожих типов, сгруппированных вместе.
- Емкость коллекции – это число элементов, которое она может содержать.
- Количество элементов коллекции – это число элементов, которое она реально содержит.
- Нижняя граница коллекции – это индекс ее первого элемента.
- Перечислитель – это объект, который выполняет итерацию в связанной с ним коллекции. Можно считать, что он является перемещаемым указателем на любой элемент коллекции. Перечислитель может быть связан только с одной коллекцией, но коллекция может иметь несколько перечислителей. Оператор `foreach` использует перечислитель и упрощает обращение с перечислителем.
- Оператор `foreach` предполагает, что все элементы коллекции имеют один и тот же тип.
- Все перечислители основаны на интерфейсе `IEnumerator` или на универсальном интерфейсе `IEnumerator<T>`, для чего им необходимо иметь перечисленные ниже члены:
 - Свойство `Current` указывает на текущий элемент коллекции.
 - Метод `MoveNext()` перемещает перечислитель к следующему элементу коллекции.
 - Метод `Reset()` перемещает перечислитель в начало коллекции. Свойство `Current` при этом указывает на положение перед первым элементом.
- Ключевое слово `yield` сообщает компилятору, что метод, в котором оно содержится, является блоком итератора. Для реализации поведения, определенного в блоке итератора, компилятор создает класс. В блоке итератора ключевое слово `yield` используется совместно с ключевым словом `return` для предоставления значения объекту перечислителя, например значения, возвращаемого в каждом цикле оператора `foreach`. Ключевое слово `yield` всегда используется вместе с ключевым словом `break` для обозначения конца итерации.
- Оператор `yield` не может использоваться в анонимных методах.

Закрепление материала

- Что такое коллекция?
- Перечислите члены интерфейса `IEnumerable`.

- Перечислите члены интерфейса `IEnumerator`.
- Что такое оператор `yield`?
- Перечислите члены интерфейса `IEnumerable<T>`.
- Перечислите члены интерфейса `IEnumerator<T>`.
- Опишите алгоритм работы цикла `foreach`.
- Можно ли использовать оператор `yield` в анонимных методах?

Дополнительное задание

Задание

Используя Visual Studio, создайте проект по шаблону Console Application.

Создайте метод, который в качестве аргумента принимает массив целых чисел и возвращает коллекцию всех четных чисел массива. Для формирования коллекции используйте оператор `yield`.

Самостоятельная деятельность учащегося

Задание 1

Выучите основные конструкции и понятия, рассмотренные на уроке.

Задание 2

Используя Visual Studio, создайте проект по шаблону Console Application.

Создайте коллекцию `MyList<T>`. Реализуйте в простейшем приближении возможность использования ее экземпляра аналогично экземпляру класса `List<T>`. Минимально требуемый интерфейс взаимодействия с экземпляром, должен включать метод добавления элемента, индексатор для получения значения элемента по указанному индексу и свойство только для чтения для получения общего количества элементов. Реализуйте возможность перебора элементов коллекции в цикле `foreach`.

Задание 3

Используя Visual Studio, создайте проект по шаблону Console Application.

Создайте коллекцию `MyDictionary<TKey,TValue>`. Реализуйте в простейшем приближении возможность использования ее экземпляра аналогично экземпляру класса `Dictionary<TKey,TValue>`. Минимально требуемый интерфейс взаимодействия с экземпляром, должен включать метод добавления элемента, индексатор для получения значения элемента по указанному индексу и свойство только для чтения для получения общего количества элементов. Реализуйте возможность перебора элементов коллекции в цикле `foreach`.

Задание 4

Используя Visual Studio, создайте проект по шаблону Console Application.

Создайте расширяющий метод:

```
public static T[] GetArray<T>(this IEnumerable<T> list){...}
```

Примените расширяющий метод к экземпляру типа `MyList<T>`, разработанному в домашнем задании 2 для данного урока. Выведите на экран значения элементов массива, который вернул расширяющий метод `GetArray()`.

Задание 5

Зайдите на сайт MSDN.

Используя поисковые механизмы MSDN, найдите самостоятельно описание темы по каждому примеру, который был рассмотрен на уроке, так, как это представлено ниже, в разделе «Рекомендуемые ресурсы», описания данного урока. Сохраните ссылки и дайте им короткое описание.

Рекомендуемые ресурсы

MSDN: Определение коллекций (Руководство по программированию на C#)

<https://msdn.microsoft.com/ru-ru/library/7y3x785f.aspx>

MSDN: Ключевое слово `yield` (Справочник по C#)

<http://msdn.microsoft.com/ru-ru/library/9k7k7cf0.aspx>

MSDN: Итераторы (Руководство по программированию на C#)

<http://msdn.microsoft.com/ru-ru/library/dscyy5s0.aspx>

MSDN: Управление коллекциями и их создание

<http://msdn.microsoft.com/ru-ru/library/14ek9axh.aspx>

MSDN: Перечисление коллекции

<http://msdn.microsoft.com/ru-ru/library/hwyysy67.aspx>