

Пользовательские коллекции

№ урока: 1 Курс: C# Professional

Средства обучения: Компьютер с установленной Visual Studio

Обзор, цель и назначение урока

На этом уроке рассматривается работа с пользовательскими коллекциями в Microsoft .NET Framework. Учащимся демонстрируются различные примеры создания пользовательских коллекций, разъясняется работа ключевых интерфейсов (`IEnumerable` и `IEnumerator`, `IEnumerable<T>` и `IEnumerator<T>`, `ICollection` и `ICollection<T>`, `IList`), отдельное внимание уделяется рассмотрению принципов работы оператора `yield` и цикла `foreach`. Кроме того, слушатели познакомятся с концепцией шаблона проектирования `Iterator`.

Изучив материал данного занятия, учащийся сможет:

- Понимать работу коллекций.
- Понимать работу оператора `yield` и цикла `foreach`.
- Эффективно использовать шаблон проектирования `Iterator`.
- Создавать собственные коллекции, наиболее соответствующие поставленным задачам, организовывать эффективную навигацию по их содержимому.

Содержание урока

1. Коллекции.
2. Интерфейсы `IEnumerable`, `IEnumerator`, `IEnumerable<T>`, `IEnumerator<T>`.
3. Работа цикла `foreach`.
4. Оператор `yield`.
5. Интерфейсы `ICollection` и `ICollection<T>`.
6. Создание простой коллекции с помощью реализации интерфейса `IList`.
7. Рассмотрение коллекции созданной с использованием паттерна `Iterator`.

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 коллекцией.
- Оператор `foreach` предполагает, что все элементы коллекции имеют один и тот же тип.
- Все перечислители основаны на интерфейсах `IEnumerator` и `IEnumerable` или на их обобщенных аналогах `IEnumerator<T>` и `IEnumerable<T>`.
- 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.

- Интерфейс **ICollection** определяет размер, перечислители и методы синхронизации для всех нестандартных коллекций. Интерфейс **ICollection** — это базовый интерфейс для классов в пространстве имен `System.Collections`, а интерфейс **ICollection<T>** является базовым интерфейсом для классов в пространстве имен `System.Collections.Generic`.
- Интерфейс **ICollection** расширяет интерфейсы **IEnumerable**, **IDictionary** и **IList** и является более специализированным интерфейсом.
- Реализация **IList** является коллекцией и доступ к ее элементам может быть осуществлен с помощью индекса, например, как в классе `ArrayList`.
- **Iterator** – паттерн поведения объектов, предоставляющий последовательный доступ ко всем элементам составного объекта, не раскрывая его внутреннего представления. Данный паттерн допустимо применять для доступа к содержимому агрегированных объектов без раскрытия их внутреннего представления, поддержки нескольких видов активных обходов одного и того же агрегированного объекта, или для предоставления единообразного интерфейса с целью обхода различных агрегированных структур (то есть, для поддержки полиморфной итерации).

Закрепление материала

- Что такое коллекция?
- Перечислите члены интерфейсов **IEnumerable**, **IEnumerator**, **IEnumerable<T>**, **IEnumerator<T>**.
- Что такое оператор `yield`?
- Опишите алгоритм работы цикла `foreach`.
- Можно ли использовать оператор `yield` в анонимных методах?
- В каких случаях следует применять паттерн `Iterator` и какое его основное предназначение?

Дополнительное задание

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

Самостоятельная деятельность учащегося

Задание 1

Выучите основные конструкции и понятия, рассмотренные на уроке.

Задание 2

Создайте коллекцию, в которой бы хранились наименования 12 месяцев, порядковый номер и количество дней в соответствующем месяце. Реализуйте возможность выбора месяцев, как по порядковому номеру, так и количеству дней в месяце, при этом результатом может быть не только один месяц.

Задание 3

Создайте коллекцию, которая бы по своей структуре напоминала «родовое дерево» (имя человека, год рождения), причем в нее можно добавлять/удалять нового родственника, есть возможность увидеть всех наследников выбранного человека, отобразить родственников по году рождения.

Задание 4

Создайте коллекцию, в которую можно записывать два значения одного слова, по типу русско-англо-украинский словарь. И в ней можно для украинского слова найти либо только русское значение, либо только английское и вывести его на экран.

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#)
<http://msdn.microsoft.com/ru-ru/library/akyhke97.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>

MSDN: Пространство имен System.Collections
<http://msdn.microsoft.com/ru-ru/library/k166wx47.aspx>