

Универсальные шаблоны. Ограничения

№ урока: 11 Курс: C# Essential

Средства обучения: Компьютер с установленной Visual Studio

Обзор, цель и назначение урока

Рассмотрение универсальных типов.

Изучив материал данного занятия, учащийся сможет:

- Использовать ограничения параметризованных типов.
- Использовать параметризованные коллекции ([List<T>](#), [Dictionary<TKey,TValue>](#)).

Содержание урока

1. Ограничения обобщений.
2. Преимущества использования обобщенных коллекций.

Резюме

- При определении универсального типа можно ограничить виды типов, которые могут использоваться клиентским кодом в качестве аргументов типа при инициализации соответствующего класса. При попытке клиентского кода создать экземпляр класса с помощью типа, который не допускается ограничением, в результате возникает ошибка компиляции. Это называется ограничениями.
- Ограничения определяются с помощью контекстно-зависимого ключевого слова [where](#).
- Ограничение [new](#) указывает, что аргумент любого типа в объявлении общего класса должен иметь открытый конструктор без параметров. Использовать ограничение [new](#) можно только в том случае, если тип не является абстрактным.
- [where T : struct](#) – аргумент типа должен быть структурного типа, кроме [Nullable](#).
- [where T : class](#) – аргумент типа должен иметь ссылочный тип; это также распространяется на тип любого класса, интерфейса, делегата или массива.
- [where T : <base class name>](#) – аргумент типа должен являться или быть производным от указанного базового класса.
- [where T : U](#) – аргумент типа, поставляемый для T, должен являться или быть производным от аргумента, поставляемого для U. Это называется неприкрытым ограничением типа.
- Коллекция [ArrayList](#) – коллекция с динамическим увеличением размера до нужного значения.
- Емкость коллекции [ArrayList](#) – это количество элементов, которое может содержать [ArrayList](#). При добавлении элементов в коллекцию [ArrayList](#) ее емкость автоматически увеличивается нужным образом за счет перераспределения внутреннего массива.
- Если размер коллекции можно оценить, целесообразно указать для нее начальную емкость, чтобы избавиться от необходимости выполнять операции изменения размера при добавлении элементов в коллекцию [ArrayList](#).
- [List<T>](#) – класс представляет строго типизированный список объектов, доступных по индексу. Поддерживает методы для поиска по списку, выполнения сортировки и других операций со списками.
- [Dictionary<TKey, TValue>](#) – класс представляет коллекцию ключей и значений.

Закрепление материала

- Что такое обобщение?

- Что такое ограничение обобщений?
- Какие виды ограничений обобщений вы знаете?
- Объясните понятия ковариантности и контравариантности обобщений.
- Какие преимущества использования обобщений?
- Какие обобщенные коллекции вы знаете?

Дополнительное задание

Задание

Используя Visual Studio, создайте проект по шаблону Console Application.

В коллекцию `ArrayList`, через вызов метода `Add` добавьте элементы структурного и ссылочного типа, переберите данную коллекцию с помощью, цикла `for`. С какой проблемой вы столкнулись?

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 класс `CarCollection<T>`. Реализуйте в простейшем приближении возможность использования его экземпляра для создания парка машин. Минимально требуемый интерфейс взаимодействия с экземпляром, должен включать метод добавления машин с названием машины и года ее выпуска, индексатор для получения значения элемента по указанному индексу и свойство только для чтения для получения общего количества элементов.

Создайте метод удаления всех машин автопарка.

Задание 3

Используя Visual Studio, создайте проект по шаблону Console Application.

Создайте класс `Dictionary<TKey,TValue>`. Реализуйте в простейшем приближении возможность использования его экземпляра аналогично экземпляру класса `Dictionary` из пространства имен `System.Collections.Generic`. Минимально требуемый интерфейс взаимодействия с экземпляром, должен включать метод добавления пар элементов, индексатор для получения значения элемента по указанному индексу и свойство только для чтения для получения общего количества пар элементов.

Задание 4

Используя Visual Studio, создайте проект по шаблону Console Application.

Создайте класс `ArrayList`. Реализуйте в простейшем приближении возможность использования его экземпляра аналогично экземпляру класса `ArrayList` из пространства имен `System.Collections`.

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: Универсальные шаблоны (Руководство по программированию на C#)

[http://msdn.microsoft.com/ru-ru/library/512aeb7t\(v=VS.90\).aspx](http://msdn.microsoft.com/ru-ru/library/512aeb7t(v=VS.90).aspx)

MSDN: Ограничения параметров типа (Руководство по программированию на C#)

[http://msdn.microsoft.com/ru-ru/library/d5x73970\(v=vs.90\).aspx](http://msdn.microsoft.com/ru-ru/library/d5x73970(v=vs.90).aspx)

MSDN: Ограничение `new` (Справочник по C#)

[http://msdn.microsoft.com/ru-ru/library/sd2w2ew5\(v=vs.90\).aspx](http://msdn.microsoft.com/ru-ru/library/sd2w2ew5(v=vs.90).aspx)