

Универсальные шаблоны

№ урока: 10 Курс: C# Essential

Средства обучения: Компьютер с установленной Visual Studio

Обзор, цель и назначение урока

Рассмотрение универсальных типов.

Изучив материал данного занятия, учащийся сможет:

- Понимать работу параметризованных типов.
- Использовать **Nullable** типы.

Содержание урока

1. Обобщения.
2. Ковариантность и контравариантность обобщений.
3. **Nullable** типы.
4. Операции поглощения.

Резюме

- Обобщение – элемент кода, способный адаптироваться для выполнения общих (сходных) действий над различными типами данных.
- Универсальные шаблоны были добавлены в язык C# версии 2.0 и среду CLR. Эта возможность CTS (Common Type System – общая система типов), названа обобщениями (generics).
- Обобщения обеспечивают большую производительность, так как не происходит операции "упаковки-распаковки" (boxing-unboxing).
- Обобщения обеспечивают безопасность типов, так как могут содержать только типы, которые Вы задаете при объявлении.
- Обобщения позволяют создавать открытые (open-ended) типы, которые преобразуются в закрытые во время выполнения.
- Идентификатор <T> – это указатель места заполнения, вместо которого подставляется любой тип.
- Перегрузки обобщенных типов различаются количеством параметров типа, а не их именами. Правильная перегрузка: `MyClass<T>{ }`, `MyClass<T,R>{ }`
- Пример открытого типа: `MyClass<T>`
- Пример закрытого типа: `MyClass<int>`
- Каждый закрытый тип получает свою собственную копию набора статических полей.
- Общие сведения об универсальных шаблонах:
 - 1) Используйте универсальные типы для достижения максимального уровня повторного использования кода, безопасности типа и производительности.
 - 2) Наиболее частым случаем использования универсальных шаблонов является создание классов коллекций.
 - 3) Можно создавать собственные универсальные интерфейсы, классы, методы, события и делегаты.
 - 4) Доступ универсальных классов к методам можно ограничить определенными типами данных
- Ковариантность обобщений – upcast параметров типов.
- Контравариантность обобщений – downcast параметров типов.
- Ковариантность и контравариантность обобщений в языке C# ограничена делегатами и интерфейсами.

- Ключевое слово `default` применяется для установки значения по умолчанию для параметра типа.
- Тип `Nullable<T>` представляет типы значений с пустыми (нулевыми) значениями.

Закрепление материала

- Что такое обобщение?
- Что такое закрытый тип?
- Что такое открытый тип?
- Объясните понятия ковариантности и контравариантности обобщений.
- Какие преимущества использования обобщений?
- Что такое `Nullable` тип?
- Что такое операция поглощения?
- Какие вы знаете типы ограничений для обобщений?

Дополнительное задание

Задание

Используя Visual Studio, создайте проект по шаблону Console Application.

Создайте класс `MyClass<T>`, содержащий статический фабричный метод – `T FactoryMethod()`, который будет порождать экземпляры типа, указанного в качестве параметра типа (указателя места заполнения типом – `T`).

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 `MyList<T>`. Реализуйте в простейшем приближении возможность использования его 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.

Задание 3

Используя Visual Studio, создайте проект по шаблону Console Application.

Создайте класс `MyDictionary<TKey, TValue>`. Реализуйте в простейшем приближении возможность использования его экземпляра аналогично экземпляру класса `Dictionary` (Урок 6 пример 5). 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 расширяющий метод: `public static T[] GetArray<T>(this MyList<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: 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: Ключевое слово `default` (Справочник по C#)
[http://msdn.microsoft.com/ru-ru/library/xwth0h0d\(v=VS.90\).aspx](http://msdn.microsoft.com/ru-ru/library/xwth0h0d(v=VS.90).aspx)

MSDN: Универсальные методы (Руководство по программированию на C#)
[http://msdn.microsoft.com/ru-ru/library/twcad0zb\(v=VS.90\).aspx](http://msdn.microsoft.com/ru-ru/library/twcad0zb(v=VS.90).aspx)