Анонимные и динамические типы. LINQ.

№ урока: 17 **Курс:** C# Essential

Средства обучения: Компьютер с установленной Visual Studio

Обзор, цель и назначение урока

Рассмотрение анонимных типов.
Рассмотрение LINQ.
Рассмотрение динамических типов.

Изучив материал данного занятия, учащийся сможет:

- Понимать и использовать анонимные типы.
- Понимать механизмы работы LINQ.
- Понимать и использовать динамические типы.

Содержание урока

1. Анонимные типы.
2. LINQ.
3. Динамические типы.

Резюме

- Анонимные типы предлагают удобный способ инкапсуляции набора свойств в один объект без необходимости предварительного явного определения типа.
- Имя типа создается компилятором и недоступно на уровне исходного кода.
- Анонимные типы являются ссылочными типами, которые происходят непосредственно от класса `object`. Компилятор присваивает им имена, несмотря на то что эти имена недоступны для приложения.
- В анонимных типах можно использовать анонимные типы
- Анонимные типы обычно используются в предложении select выражения запроса для возврата подмножества свойств из каждого объекта в исходной последовательности
- **LINQ** (Language-Integrated Query) представляет собой набор функций, расширяющих мощные возможности запроса в синтаксисе языка C#. LINQ представляет стандартные, легко изучаемые шаблоны для создания запросов и обновления данных; технология может быть расширена для поддержки потенциально любого типа хранилища данных. Visual Studio включает сборки поставщиков LINQ, позволяющие использовать LINQ с коллекциями платформы .NET Framework, базами данных SQL Server, наборами данных ADO.NET и XML-документами
- LINQ (Language-Integrated Query) является революционной инновацией в .NET Framework версии 3.5, которая является мостом между миром объектов и миром данных
- Все операции запроса LINQ состоят из трех различных действий.
  1. Получение источника данных.
  2. Создание запроса.
  3. Выполнение запроса
- Выражение запроса должно начинаться с **from**. Кроме того, выражение запроса может включать вложенные запросы, начинающиеся с from. Ключевое слово from определяет следующее:
  - Источник данных, применительно к которому запрос или вложенный запрос будет выполняться.
  - Локальную переменную диапазона, представляющую каждый элемент исходной последовательности

- **where** – используется в выражении запроса для указания элементов, возвращаемых из источника данных, в выражении запроса. Оно применяет логическое условие (*предикат*) к каждому исходному элементу (со ссылкой переменной диапазона) и возвращает элементы, для которых заданное условие является истинным. В одном выражении запроса может присутствовать несколько where, а в одном where – несколько частей выражения предиката
- В выражении запроса **select** задает тип значений, получаемых при выполнении запроса. Результат основывается на анализе всех предыдущих запросов и на любых выражениях внутри предложения select. Выражение запроса должно завершаться select или group
- **group** – возвращает последовательность объектов `IGrouping`<TKey, TElement>, содержащих ноль или более элементов, соответствующих значению ключа группы.
- Чтобы выполнить дополнительные запросы для каждой из групп, можно указать временный идентификатор, воспользовавшись для этого контекстным ключевым словом into. При использовании ключевого слова into необходимо продолжить запрос и завершить его инструкцией select или group.
- В выражении запроса **orderby** осуществляет сортировку возвращенной последовательности по возрастанию или по убыванию. Для выполнения одной или нескольких операций последующей сортировки можно указать несколько ключей. Сортировка выполняется функцией сравнения по умолчанию для данного типа элементов. По умолчанию используется порядок сортировки по возрастанию. Можно также указать пользовательскую функцию по умолчанию
- **join** – используется для связывания элементов из различных последовательностей источников, которые не имеют прямых связей в объектной модели. Единственное требование – элементы в каждом источнике должны совместно использовать некоторое значение, которое можно сравнить на предмет равенства
- join в качестве ввода принимает две последовательности источников. Элементы в каждой последовательности должны быть свойством или содержать свойство, которое можно сравнить с соответствующим свойством в другой последовательности. join сравнивает указанные ключи на предмет равенства при помощи специального ключевого слова equals.
- **let** – представляет новый локальный идентификатор, на который можно ссылаться в остальной части запроса. Его можно представить, как локальную переменную видимую только внутри выражения запроса
- В C# 4.0 появился новый тип – `dynamic`. Тип является статическим типом, но объект типа `dynamic` обходит проверку статического типа. В большинстве случаев он функционирует, как тип `object`. Во время компиляции предполагается, что элементы с типом `dynamic` поддерживают любые операции. Поэтому разработчику не нужно следить за тем, откуда объект получает свое значение.
- В отличие от ключевого слова `var`, объект, объявленный как `dynamic`, может менять тип во время выполнения. При использовании ключевого слова `var` определение типа объекта откладывается. Но как только он определен компилятором, изменять его уже нельзя. Что касается объекта `dynamic`, то можно не просто изменить его тип, но делать это многократно. Это отличается от приведения объекта от одного типа к другому. При приведении объекта создается новый объект с другим, но совместимым типом.

Закрепление материала

- Что такое анонимные типы?
- Что такое LINQ?
- Что такое выражение запроса?
- Что такое переменная запроса?
- Что такое переменная диапазона?
- Что такое динамический тип?
- Что делает оператор `dynamic`?

Дополнительное задание

Задание
Используя Visual Studio, создайте проект по шаблону Console Application.
Создайте класс `Calculator`, методы которого принимают аргументы и возвращают значения типа `dynamic`.

Самостоятельная деятельность учащегося

Задание 1
Выучите основные конструкции и понятия, рассмотренные на уроке.

Задание 2
Используя Visual Studio, создайте проект по шаблону Console Application
Представьте, что вы пишите приложение для Автостанции и вам необходимо создать простую коллекцию автомобилей со следующими данными: Марка автомобиля, модель, год выпуска, цвет. А также вторую коллекцию с моделью автомобиля, именем покупателя и его номером телефона. Используя простейший LINQ запрос, выведите на экран информацию о покупателе одного из автомобилей и полную характеристику приобретенной ними модели.

Задание 3
Используя Visual Studio, создайте проект по шаблону Console Application
Используя динамические и анонимные типы данных, создайте Англо-Русский словарь на 10 слов и выведите на экран его содержание.

Задание 4
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: Анонимные типы (Руководство по программированию на C#)
http://msdn.microsoft.com/ru-ru/library/bb397696(v=VS.100).aspx

MSDN: LINQ
http://msdn.microsoft.com/ru-ru/library/bb397926(VS.90).aspx

MSDN: dynamic (Справочник по C#)
http://msdn.microsoft.com/ru-ru/library/dd264741.aspx

MSDN: Дино Эспозито. Применение ключевого слова `dynamic` в C# 4.0
http://msdn.microsoft.com/ru-ru/magazine/ee336309.aspx