C# LINQ

Abril 2017

Agenda

- **Introducción**
- LINQ to Objects
- LINQ to SQL
- LINQ to XML
- LINQ to DataSet

Genéricos

- Los tipos genéricos son una nueva característica en la versión 2.0 del lenguaje C# y CLR.

- Estos tipos agregan el concepto de parámetros de tipo a .NET Framework.

- Mediante la utilización de un parámetro de tipo genérico T, se puede escribir una clase única que otro código de cliente puede utilizar sin generar el costo o el riesgo de conversiones en tiempo de ejecución u operaciones de conversión boxing.

Genéricos

- Utilice los tipos genéricos para maximizar la reutilización, seguridad de tipos y rendimiento del código. El uso más común de genéricos es crear clases de colección.

- La biblioteca de clases de .NET Framework contiene varias nuevas clases de colección genéricas en el espacio de nombres System.Collections.Generic. Éstas se deberían utilizar siempre que sea posible en lugar de clases como ArrayList en el espacio de nombres System.Collections.

- Puede crear sus propias interfaces, clases, métodos, eventos y delegados genéricos. Las clases genéricas se pueden restringir para permitir el acceso a métodos en tipos de datos determinados.

- Se puede obtener información sobre los tipos utilizados en un tipo de datos genéricos en tiempo de ejecución y mediante reflexión.

http://msdn.microsoft.com/es-es/library/512aeb7t.aspx

Introducción

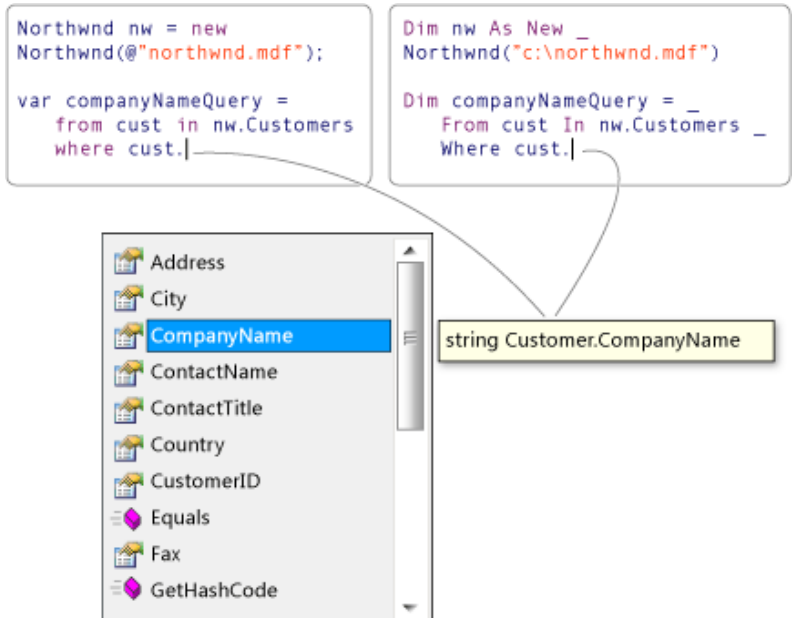
- Language-Integrated Query (LINQ) es una innovación introducida en Visual Studio 2008 y .NET Framework versión 3.5 que elimina la distancia que separa el mundo de los objetos y el mundo de los datos.

- Tradicionalmente, las consultas con datos se expresan como cadenas sencillas, sin comprobación de tipos en tiempo de compilación ni compatibilidad con IntelliSense. Además, es necesario aprender un lenguaje de consultas diferente para cada tipo de origen de datos:

  - Bases de datos SQL
  - Documentos XML
  - Servicios Web, etc

Introducción

- LINQ convierte una consulta en una construcción de lenguaje de primera clase en C# y Visual Basic.
- Consulta LINQ parcialmente completada en una base de datos SQL Server en C#:



IntelliSense

- **Microsoft IntelliSense** es la aplicación de autocompletar utilizado en el IDE Microsoft Visual Studio. Además de completar el símbolo de los nombres que el programador está escribiendo, IntelliSense sirve como documentación de los nombres de variables, funciones y métodos de utilización de metadatos basados en la reflexión.

LINQ

- Las consultas Linq pueden realizarse con bases de datos SQL Server, documentos XML, conjuntos de datos ADO.NET y cualquier colección de objetos que admita IEnumerable o la interfaz genérica IEnumerable<T>.

- También se ha previsto la compatibilidad de LINQ con ADO.NET Entity Framework, y otros fabricantes se encuentran escribiendo proveedores LINQ para muchos servicios Web y otras implementaciones de bases de datos.

LINQ

- Se puede utilizar consultas LINQ en proyectos nuevos o junto a consultas que no son LINQ en proyectos existentes. El requisito es que el proyecto esté orientado a .NET Framework 3.5 o posterior.

LINQ

- Todas las operaciones de consulta LINQ se componen de tres acciones distintas:
  - Obtención del origen de datos.
  - Creación de la consulta.
  - Ejecución de la consulta.

LINQ

```
class IntroToLINQ
{
   static void Main()
   {
      // Tres partes de una consulta LINQ
      //  1. Data source.
      int[] numbers = new int[7] { 0, 1, 2, 3, 4, 5, 6 };

      // 2. Creación de consulta.
      // numQuery is an IEnumerable<int>
      var numQuery =
         from num in numbers
         where (num % 2) == 0
         select num;

      // 3. Ejecución de consulta.
      foreach (int num in numQuery)
      {
         Console.Write("{0,1} ", num);
      }
   }
}
```

Agenda

- Introducción
- **LINQ to Objects**
- LINQ to SQL
- LINQ to XML
- LINQ to DataSet

LINQ to Objects

- El término "LINQ to Objects" hace referencia al uso directo de consultas LINQ con cualquier colección IEnumerable o IEnumerable<T> sin utilizar ninguna API o proveedor LINQ intermedio, como LINQ to SQL o LINQ to XML.

LINQ to Objects - Ventajas

- Además, las consultas LINQ ofrecen tres ventajas principales respecto a los bucles foreach tradicionales:
  - Son más concisas y legibles, sobre todo al filtrar varias condiciones.
  - Proporcionan funcionalidad eficaz de filtrado, ordenación y agrupación con código de aplicación mínimo.
  - Se pueden trasladar a otros orígenes de datos con pocas o ningunas modificaciones.

LINQ to Objects - Ventajas

- Cuanto más compleja sea la operación que se desee realizar con los datos, mayor será el número de ventajas al utilizar LINQ en lugar de las técnicas de iteración tradicionales.

Agenda

- Introducción
- LINQ to Objects
- **LINQ to SQL**
- LINQ to XML
- LINQ to DataSet

LINQ to SQL

- En LINQ to SQL, el modelo de datos de una base de datos relacional se asigna a un modelo de objetos expresado en el lenguaje de programación del programador.

- Cuando la aplicación se ejecuta, LINQ to SQL convierte a SQL las consultas integradas en el lenguaje en el modelo de objetos y las envía a la base de datos para su ejecución.

- Cuando la base de datos devuelve los resultados, LINQ to SQL los vuelve a convertir en objetos con los que pueda trabajar en su propio lenguaje de programación.

Agenda

- Introducción
- LINQ to Objects
- LINQ to SQL
- **LINQ to XML**
- LINQ to DataSet

LINQ to XML

- LINQ to XML proporciona una interfaz de programación XML en memoria que aprovecha las características de .NET Language-Integrated Query (LINQ) Framework. LINQ to XML utiliza las características más recientes del lenguaje .NET Framework y es comparable a una actualizada y rediseñada interfaz de programación XML para el Modelo de objetos de documento (DOM).

- La familia de tecnologías de LINQ proporciona un completo entorno de creación de consultas para objetos (LINQ to Objects), bases de datos relacionales (LINQ to SQL) y XML (LINQ to XML).

Agenda

- Introducción
- LINQ to Objects
- LINQ to SQL
- LINQ to XML
- **LINQ to DataSet**

LINQ to DataSet

- LINQ to DataSet facilita y acelera las consultas en datos almacenados en caché en un objeto DataSet.

- LINQ to DataSet simplifica la consulta permitiendo a los desarrolladores escribir consultas a partir del lenguaje de programación mismo, en lugar de utilizar un lenguaje de consulta diferente.

- Esto resulta especialmente útil para desarrolladores de Visual Studio, que ahora pueden aprovechar la comprobación de sintaxis en tiempo de compilación, los tipos estáticos y la compatibilidad con IntelliSense que proporciona Visual Studio en las consultas.