HEINSOHN
BUSINESS TECHNOLOGY

La librería de clases de .NET

Microsoft Partner
Gold Software Development
Gold Web Development
Gold Software Asset Management

HEINSOHN
BUSINESS TECHNOLOGY

Agenda

- Base Class Library
  - Introducción
  - Aplicaciones tipo Consola
  - Manejo de Strings
  - XML
  - Entrada y Salida de Archivos
  - Compresion
  - Colecciones de Objetos

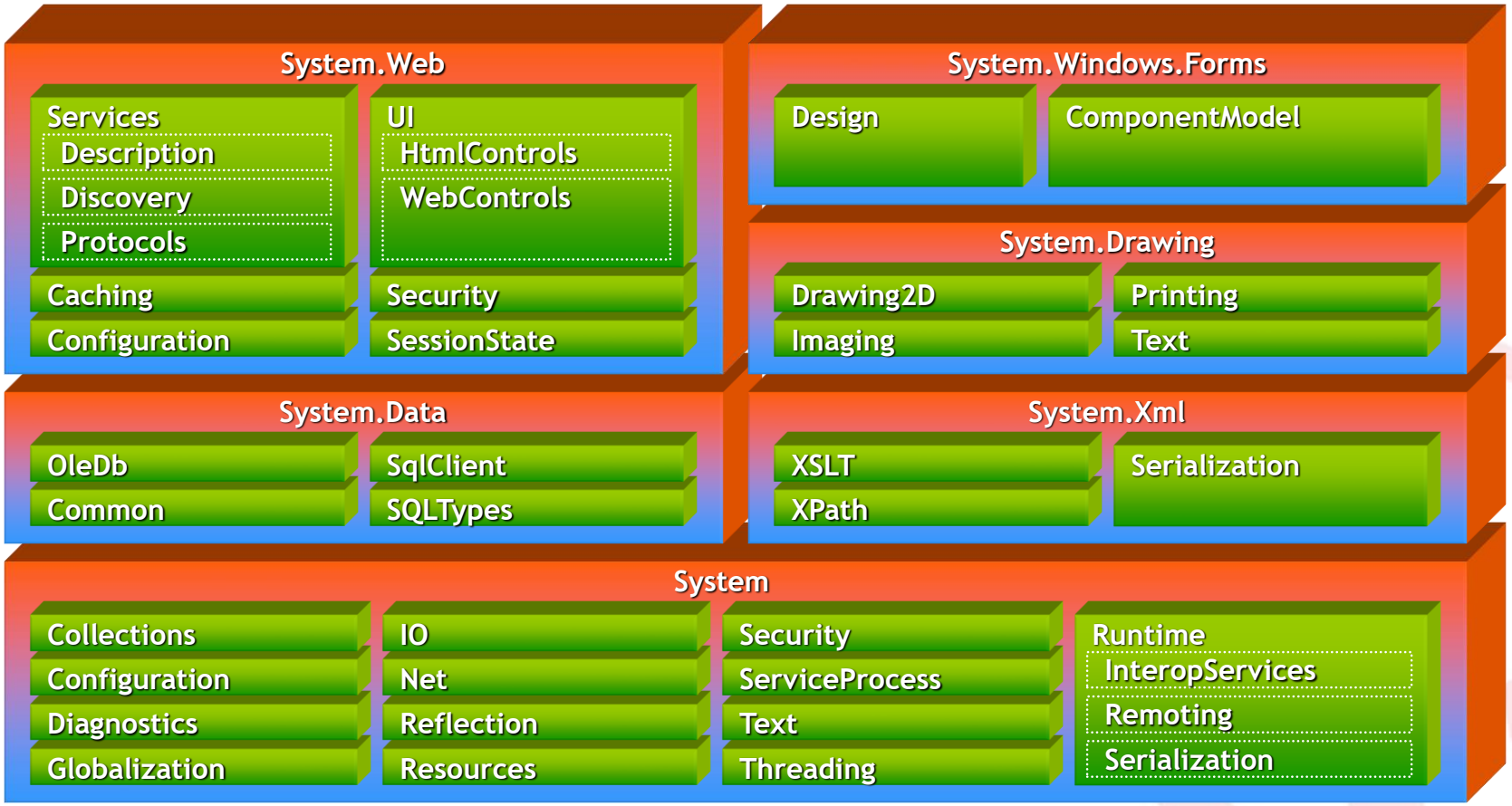
HEINSOHN
BUSINESS TECHNOLOGY

Introducción

- La BCL es un conjunto de tipos (clases, interfaces, etc.) básicos que vienen predefinidos en el .NET Framework
- Estos tipos están organizados en jerarquías lógicas de nombres, denominado NAMESPACE
- Estos tipos también son INDEPENDIENTES del lenguaje de desarrollo
- La BCL es extensible y totalmente orientada a objetos

HEINSOHN
BUSINESS TECHNOLOGY

El namespace raíz es SYSTEM



Base Class Library

HEINSOHN
BUSINESS TECHNOLOGY

System.Console

- Funcionalidades:
  - Get y set del tamaño de la consola.
  - Cambiar los colores y el título de la consola
  - Capturar la presión de teclas
  - Imprimir en pantalla.

HEINSOHN
BUSINESS TECHNOLOGY

System.String

- String es una colección secuencial de objetos System.Char
- Métodos:
  - Subtring
    - Recupera una subcadena de la instancia. La subcadena comienza en una posición de carácter especificada
  - Split
    - Identifica las subcadenas de la instancia que están delimitadas por uno o varios caracteres especificados en una matriz, y las coloca después en una matriz de elementos String.
  - Concat
    - Concatena una o más instancias de String o las representaciones de tipo String de los valores de una o más instancias de Object.

System.String

- Insert
  - Inserta una instancia especificada de String en una posición de índice especificada de la instancia.
- Replace
  - Reemplaza todas las apariciones de un carácter Unicode o un objeto String en la instancia por otro carácter Unicode u otro objeto String.
- Trim
  - Quita todas las apariciones de un conjunto de caracteres especificados desde el principio y el final de la instancia.
- ToLower
  - Devuelve una copia de String en minúsculas.
- ToUpper
  - Devuelve una copia de String en mayúsculas.

System.DateTime

- Representa un instante de tiempo
- Métodos más comunes
  - ToString
    - Convierte el valor de esta instancia en la representación de cadena equivalente
  - AddDays (..AddHours, AddYear, AddMinutes, etc)
    - Agrega el número de días especificado al valor de esta instancia

System.DateTime

- Propiedades más comunes
  - Date
    - Obtiene el componente correspondiente a la fecha de esta instancia a las 0hs
  - Day
    - Obtiene el día del mes
  - Now
    - Obtiene un DateTime que constituye la fecha y hora locales actuales de este equipo

HEINSOHN
BUSINESS TECHNOLOGY

Laboratorio

- Consola, Strings & DateTimes

Qué es exactamente XML?

- eXtensible Markup Language
- Estándar de la industria (W3C)
- Independientes de la aplicación
- XML define sólo la estructura de los datos, no la forma de presentación

- HTML - Lenguaje de Presentación

```
<span style="font-family:Arial">Date: 05/04/2005</span>
<span style="font-family:Verdana">Subject: .NET</span>
<span style="font-family:Arial">Speaker: Bill Gates</span>
```

- XML - Lenguaje Descriptivo

```
<Event>
  <Date>05/04/2005</Date>
  <Subject>XML</Subject>
  <Speaker>Bill Gates</Speaker>
</Event>
```

TXT

```
15334, 29.00, "Steve B.", "XML", MS Press …
```

Se puden enviar datos de una aplicación a otra (incluso en plataformas distintas), pero el layout de los mismos tiene que ser conocido por ambas aplicaciones.

XML

```
<Book>
    <id> 15534 </id>
    <Price> 29.00 </Price>
    <author> Steve B. </author>
    <title> XML </title>
    <editor> MS Press </editor>
</Book>
```

La representación de los datos (metadata) está en el mismo lugar que los datos, por lo cual estos pueden ser consumidos por cualquier aplicación en cualquier plataforma.

System.XML

- Espacio de nombre System.XML
  - XMLReader
    - XMLTextReader
    - Name
    - NodeType
    - Value
  - XMLWriter
    - XMLTextWriter
    - WriteElementString
    - WriteElementString
    - WriteStartElement
    - WriteEndElement
    - WriteStartAttribute
    - WriteEndAttribute

HEINSOHN
BUSINESS TECHNOLOGY

Laboratorio

- Crear un documento XML
- Utilizar las clases de System.XML

Entrada/salida de archivos

- Espacio de nombre System.IO
- Directory y DirectoryInfo
- Path
- File y FileInfo
- Lectura y escritura de archivos
- E/S de XML

HEINSOHN
BUSINESS TECHNOLOGY

Colecciones de Objetos

- Colecciones:
  - Objetos que, internamente, gestionan arrays, pero que estan preparados para gestionarlos de forma opimizada
- Espacio de Nombres de System.Collection
  - ArrayList: Array cuyo número de elementos puede modificarse dinámicamente
  - HashTable: El acceso a los valores del array se realiza a traves de una clave asociada a cada elemento
  - SortedList: Variación de un HashTable en la que los elementos se ordenan por la clave según van siendo agregados, funcionamiento similar al HashTable

System.Collection.ArrayList

ArrayList – Ejemplo (C#)

```
using System;
using System.Collections;
public class SamplesArrayList
{
    public static void Main()
    {
        // Creates and initializes a new ArrayList.
        ArrayList myAL = new ArrayList();
        myAL.Add("Hello");
        myAL.Add("World");
        myAL.Add("!");
        // Displays the properties and values of the ArrayList.
        Console.WriteLine("myAL");
        Console.WriteLine("\tCount:    {0}", myAL.Count);
        Console.WriteLine("\tCapacity: {0}", myAL.Capacity);
        Console.Write("\tValues:");
        PrintValues(myAL);
    }
    public static void PrintValues(IEnumerable myList)
    {
        System.Collections.IEnumerator myEnumerator = myList.GetEnumerator();
        while (myEnumerator.MoveNext())
            Console.Write("\t{0}", myEnumerator.Current);
        Console.WriteLine();
    }
}
```

System.Collection.HashTable

HashTable – Ejemplo (C#)

```
using System;
using System.Collections;
public class SamplesHashtable
{
    public static void Main()
    {
        // Creates and initializes a new Hashtable.
        Hashtable myHT = new Hashtable();
        myHT.Add("First", "Hello");
        myHT.Add("Second", "World");
        myHT.Add("Third", "!");

        // Displays the properties and values of the Hashtable.
        Console.WriteLine("myHT");
        Console.WriteLine("  Count:    {0}", myHT.Count);
        Console.WriteLine("  Keys and Values:");
        PrintKeysAndValues(myHT);
    }
    public static void PrintKeysAndValues(Hashtable myList)
    {
        IDictionaryEnumerator myEnumerator = myList.GetEnumerator();
        Console.WriteLine("\t-KEY-\t-VALUE-");
        while (myEnumerator.MoveNext())
            Console.WriteLine("\t{0}:\t{1}", myEnumerator.Key, myEnumerator.Value);
        Console.WriteLine();
    }
}
```

System.Collection.SortedList

SortedList – Ejemplo (C#)

```
using System;
using System.Collections;
public class SamplesSortedList
{
    public static void Main()
    {
        // Creates and initializes a new SortedList.
        SortedList mySL = new SortedList();
        mySL.Add("First", "Hello");
        mySL.Add("Second", "World");
        mySL.Add("Third", "!");
        // Displays the properties and values of the SortedList.
        Console.WriteLine("mySL");
        Console.WriteLine("  Count:    {0}", mySL.Count);
        Console.WriteLine("  Capacity: {0}", mySL.Capacity);
        Console.WriteLine("  Keys and Values:");
        PrintKeysAndValues(mySL);
    }
    public static void PrintKeysAndValues(SortedList myList)
    {
        Console.WriteLine("\t-KEY-\t-VALUE-");
        for (int i = 0; i < myList.Count; i++)
        {
            Console.WriteLine("\t{0}:\t{1}", myList.GetKey(i), myList.GetByIndex(i));
        }
        Console.WriteLine();
    }
}
```

HEINSOHN
BUSINESS TECHNOLOGY

Laboratorio

- Utilizar las clases de System.Collection

HEINSOHN
BUSINESS TECHNOLOGY

Resumen

- Hasta aquí hemos visto algunas de las capacidades que nos brinda la Librería de Clases del .Net Framework.
- Vimos como utilizar una aplicación de consola, utilizar algunos de los métodos y propiedades de la clase String y DateTime
- Comprendimos básicamente la estructura básica de un documento XML y cómo aprovecharlo utilizando las herramientas de System.XML
- Aprendimos a leer y escribir un archivo
- Por último aprendimos a utilizar algunas clases del namespace System.Collection

HEINSOHN
BUSINESS TECHNOLOGY