Chapter 4

VB .NET Programming with Supplied Classes

Objectives

In this chapter, you will:
• Use the namespaces and classes supplied with VB .NET
• Use the String class
• Create a String array
• Use the ArrayList class

Objectives

In this chapter, you will:
• Work with dates
• Format numeric output
• Use the MessageBox class
• Display a Form

Using the Namespaces and Classes Supplied with VB .NET

• VB .NET is part of Microsoft's Visual Studio .NET
• Visual Studio .NET is part of .NET technology
• Two important pieces of .NET framework:
  – Common Language Runtime (CLR)
  – .NET class library
• CLR supports common services used by Visual Studio .NET languages

Using the Namespaces and Classes Supplied with VB .NET

• .NET class library: predefined classes and their methods, organized into namespaces
• Namespace: group of related classes
• Keyword Imports: gives compiler access to classes contained in specific namespaces

Using the String Class

• String: collection of characters
• VB .NET stores string data in instances of String class
• String class is a member of System namespace
• Code to declare a string:

  Dim s1 As String = "Hello Again"
Using the String Class

- String values in VB .NET are immutable – they cannot be changed
  - Methods that appear to change a string value actually create and return a new String instance
- Index
  - Each character in a String instance has an index
  - Index values in VB .NET begin with zero

Using String Methods

- Copy method returns a second String that is a copy of the String sent as an argument

```
' create a copy of s1
Dim s2 As String = String.Copy(s1)
Console.WriteLine("s2, a copy of s1, contains ", 
  " & s2)
```

- Chars method returns the character at a specified index

```
Console.WriteLine("char at index 6 of s1 is ", 
  s1.Chars(6))
```

- There are two options to see whether two string values are equal
  - Equal sign (=)
    
    If s1 = s2 Then Console.WriteLine("s1 = s2")
  - Equals method
    
    Console.WriteLine("s1.Equals(s2) returns ", 
      " & s1.Equals(s2))
```

Using String Methods

- SubString method:
  - Extracts one or more characters from a String instance
  - Returns a new String instance containing extracted characters

```
Console.WriteLine("s1.Substring(0, 5) returns ", 
  
  s1.Substring(0, 5))
```
Using String Methods

- **Replace method** replaces one or more characters in a string with one or more other characters
  
  ```csharp
  Console.WriteLine("s1.Replace(Hello, Hi) returns " & _
  s1.Replace("Hello", "Hi"))
  Console.WriteLine("After Replace s1 contains " & s1)
  ```

Using String Methods

- **Insert method** inserts one or more characters into an existing string beginning at a specified index
  
  ```csharp
  Console.WriteLine("s1.Insert(6, There ) returns " &  _
  s1.Insert(6, "There ")
  ```

Using String Methods

- **StartsWith method**
  - Compares the string argument with beginning characters of the String instance
  - Returns True or False depending on whether there is a match
  
  ```csharp
  Console.WriteLine("s1.StartsWith(Hi) returns " & _
  s1.StartsWith("Hi"))
  ```

Using String Methods

- **EndsWith method** compares the ending characters with the argument
  
  ```csharp
  Console.WriteLine("s1.EndsWith(Again) returns " & _
  s1.EndsWith("Again"))
  ```

- **Use** **ToUpper method** to change the case of a string value to uppercase
- **Use** **ToLower method** to change the case of a string value to lowercase

Using String Methods

- **IndexOf method**
  - Searches a String instance for a specific value
  - Returns index of the beginning of the value
  - Returns -1 if no matching value was found
  
  ```csharp
  Console.WriteLine("s1.IndexOf(Again) returns " & _
  s1.IndexOf("Again"))
  ```

Using String Methods

- **Use** **ToString method** to convert numeric values to a string
  
  ```csharp
  Dim i As Integer = 5
  Console.WriteLine("value of Convert.ToString(i) is " & _
  Convert.ToString(i))
  ```

- **Use methods in Convert class** to convert String values containing numeric data to primitive data types
Creating a String Array

- Code to create a String array:
  `' declare a String array with 4 elements
  Dim stringArray(3) As String

- Elements of stringArray are reference variables of data type String

Creating a String Array

- Code to create four String instances and populate the array elements
  ```
  stringArray(0) = "Hello"
  stringArray(1) = "World"
  stringArray(2) = "Wide"
  stringArray(3) = "Web"
  ```

- A specific element can be accessed using the index of that element
- String methods can be invoked for String instances referenced by array elements

Creating a String Array

- Major limitation of arrays: fixed size
- ArrayList class
  - Member of System.Collections namespace
  - Creates dynamically resizable array – the number of elements can change during runtime

Using the ArrayList Class

- Code to create an instance of ArrayList class:
  ```
  ' create an ArrayList instance with 3 elements
  Dim anArrayList As ArrayList = New ArrayList(3)
  ```

- Code to create four instances of String class:
  ```
  ' create String instances
  Dim s1 As String = New String("Hello")
  Dim s2 As String = New String("World")
  Dim s3 As String = New String("Wide")
  Dim s4 As String = New String("Web")
  ```
Using the ArrayList Class

- Add method is used to populate the elements
- When a fourth element is added, the number of elements increases automatically

```csharp
anArrayList.Add(s1)
anArrayList.Add(s2)
anArrayList.Add(s3)
anArrayList.Add(s4)
```

Working with Dates

- System namespace contains DateTime class and TimeSpan class
  - Instance of DateTime class contains a date value
  - Instance of TimeSpan class contains the difference between two dates

```csharp
DateTime dt = new DateTime(2023, 4, 15);
TimeSpan ts = dt.Subtract(new DateTime(2022, 4, 15));
```

Working with Dates

- DateTime properties
  - Today property gets system date and returns a DateTime instance
  - Now property captures the current time
- DateTime methods to perform arithmetic on a date value
  - AddMonths adds a value to the month
  - AddDays adds a value to the day
  - AddYears adds a value to the year

```csharp
DateTime now = DateTime.Now;
now.AddDays(7);
```

Formatting Numerical Output

- Formatting means inserting commas, decimal places, dollar signs, percent symbols, parentheses, hyphens, etc.
- Each primitive data type is represented by a structure, which has methods
- When a primitive variable is declared, certain methods associated with that variable can be invoked
  - For example: ToString method
Formatting Numerical Output

- ToString method is used to format numeric data
- Format mask
  - A series of characters that describes the format to be used
  - Passed to ToString method to carry out formatting
- For example:

  ```csharp
  s = i.ToString("$#,##0.00")
  Console.WriteLine("Integer 1234 with $#,##0.00 format is " & s)
  ```

Using the MessageBox Class

- A message box can be used to display a message and to get a response
- MessageBox class
  - Member of System.Windows.Forms namespace
  - Has a single method named Show
- Show method
  - Creates an instance of MessageBox class
  - Makes a message box appear

Using the MessageBox Class

- Show method can receive up to four arguments
  - First argument – Message to be displayed
  - Second argument – Caption to be displayed
  - Third argument – Specifies the buttons to be displayed
  - Fourth argument – Determines the default button

```csharp
Figure 4-14 MessageBox with a message
```

```csharp
Figure 4-15 MessageBox with a message and caption
```

```csharp
Figure 4-17 MessageBox with Yes, No, and Cancel buttons
```
Using the MessageBox Class

• Show method can receive up to four arguments
  – Fourth argument – Specifies the type of icon displayed

Using the MessageBox Class

• Return value obtained from Show method
  – Indicates which button was clicked by user
  – Is of data type DialogResult
  – Can be compared to specific values using an If statement

Displaying a Form

• Form
  – A class in System.Windows.Forms namespace
• GuiModule.vb module contains instructions to
  – Instantiate GuiForm.vb Form
  – Make it visible and invisible
• GuiForm.vb
  – A subclass of Form
  – Inherits Form attributes and methods

Summary

• A DateTime instance contains a date value
• A TimeSpan instance contains the difference between two dates
• ToString method is used to format numeric data
• A format mask is a series of characters that describes the format to be used
• A message box is used to display a message and, optionally, get a response
• Form is a class in System.Windows.Forms namespace

• A VB .NET namespace is a library of related classes
• VB .NET string data is contained in an instance of String class
• String class contains useful methods to manipulate string data
• String values are immutable
• ArrayList class can be used to create an array that is dynamically resizeable