The String Class

Members of the String Class

- ***Length*** – returns the length of the string
- Syntax:
  - **<variable>=string_variable.Length**
  - Returns an integer
- Example

```
Dim len As Integer
Dim s as String = "Hello"
len=s.Length
```

Members of the String Class

- ***Equals*** - determines if two strings are equal, **case-sensitive**
- Syntax:
  - **<variable>=String.Equals(str1, str2)**
  - Returns a boolean value

Members of the String Class

- SubString()-retrieves a substring from a string
- ToLower()-returns a copy of the string in lowercase
- ToUpper()-returns a copy of the string in uppercase
- Trim()-removes all the spaces or other specified character from the start and end of the string

ToUpper(),ToLower()

- Example

```
String s As String="Hello"
String up,low  As String
up=s.ToUpper()      'up="HELLO"
low=s.ToLower()    'low="hello"
```

Trim

- To remove any starting or trailing blank spaces from a string
- Example

```
String s As String="      He   llo      "
String res As String
res=s.Trim()     'res="He   llo"
```

SubString()

- Syntax
  - **subString(start):** returns a string from starting index to end of string (a string starting index is zero)
  - **subString(start,length):** same as above but returns a string of a specified length from the starting index

SubString()

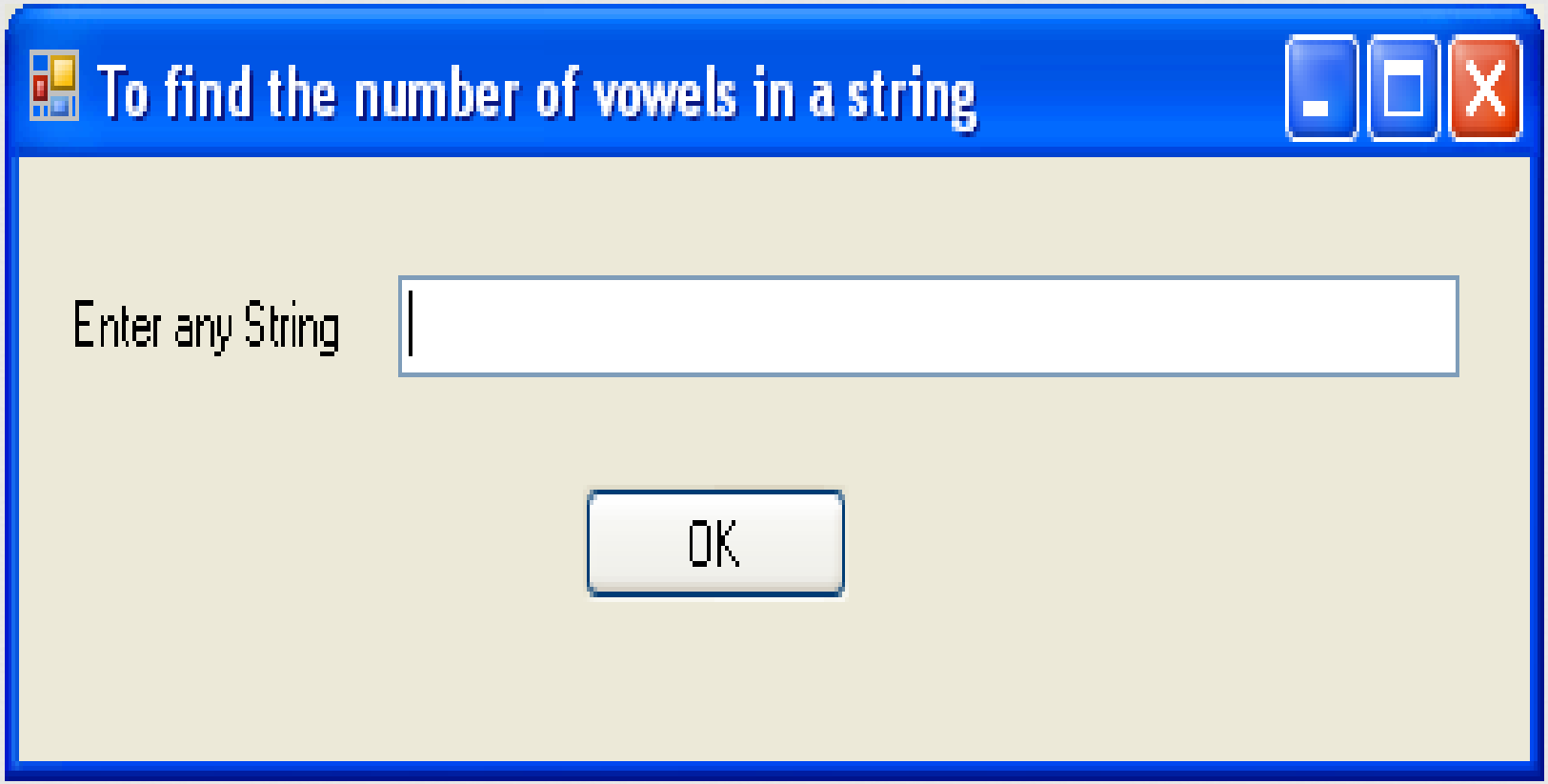
- Example

```
Dim s As String="Visual Basic Programming"
Dim s1, s2 As String
s1=s.subString(7,5)
s2=s.subString(7)
MsgBox("s1= " & s1 & " s2=" & s2)
```

Output

```
Basic
Basic  Programming
```

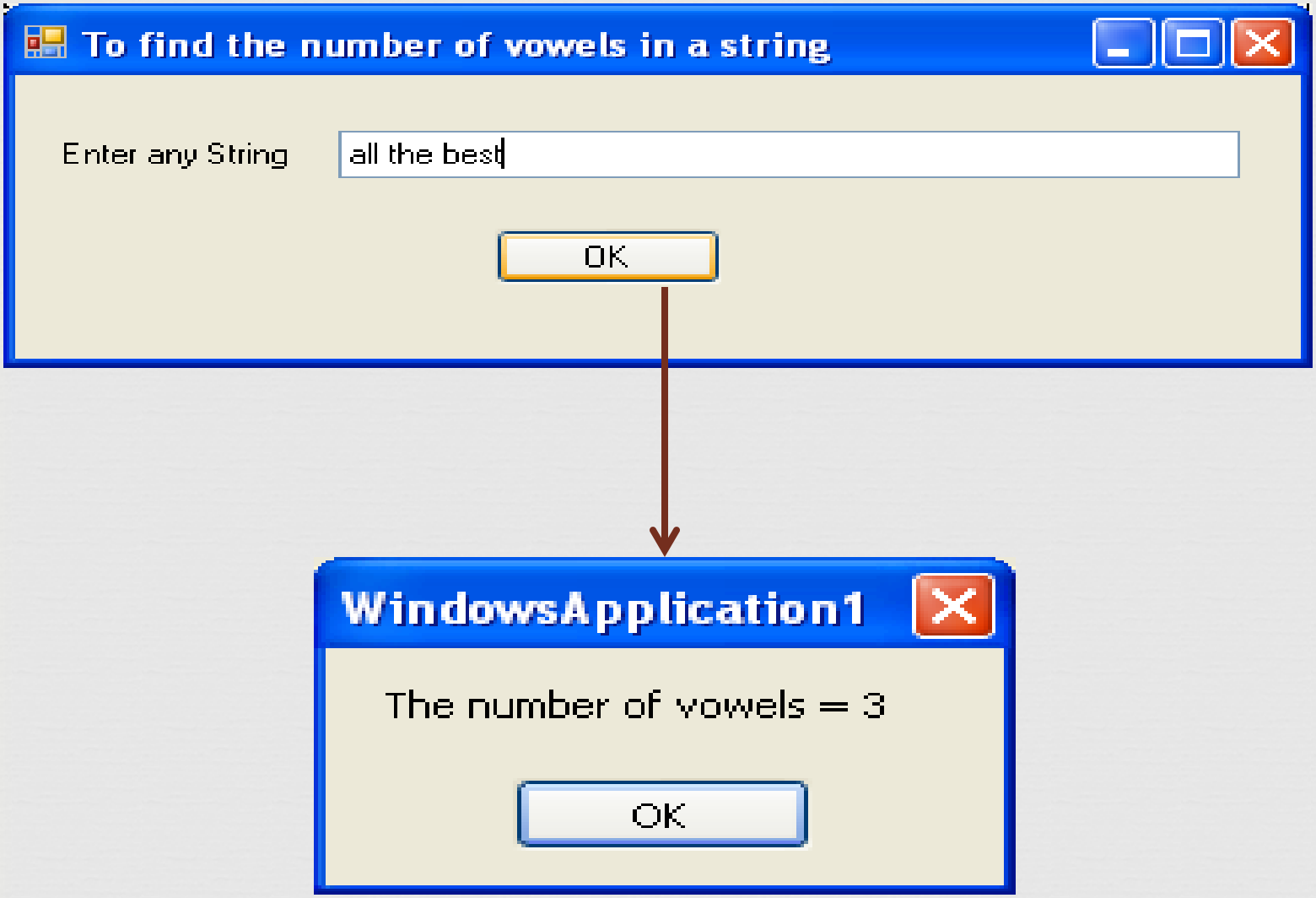
Program to find vowel in a String



```
Public Class Form2

    Private Sub btnOk_Click(ByVal sender As System.Object,
            ByVal e As System.EventArgs) Handles btnOk.Click
        Dim s As String
        Dim count As Integer
        s = Trim(txtString.Text)
        For i = 0 To s.Length - 1
            Select Case s(i)
                Case "a", "A", "e", "E", "i", "I", "o", "O", "u", "U"
                    count += 1
            End Select
        Next
        MsgBox("The number of vowels = " & count)
    End Sub

End Class
```


To find the number of vowels in a string
Enter any String
all the best
OK
WindowsApplication1
The number of vowels = 3
OK

Chars

- The Chars property returns a type Char, the single character at a specified position in the string.

- Syntax is: **Chars(position)**

- Example:

```
Dim s As String="abcde"
Dim ch As Char
ch=s.Chars(1) 'c here has the value of "b"
```

PadLeft(), PadRight()

- These methods are used for appending a character to the left or right side of a string
- We need to specify length and returns a String datatype.
  - PadLeft(*len*)
  - PadRight(*len*)

  Adding a blank space

  - PadLeft(*len, char*)
  - PadRight(*len, char*)

Example

```
Dim s As String="abcde"
Dim c, d, e, f As String
Dim len as Integer
len=s.Length()
c=s.PadLeft(5+len) 'c= "     abcde"
d=s.PadRight(5+len) 'c= "abcde     "
e=s.PadLeft((5+len),"*") 'c= "*****abcde"
f=s.PadRight((5+len),"*") 'c= "abcde*****"
```

Remove

- This method deletes a specified number of characters from a specified position in a string and returns resultant string
- Syntax: **Remove(start, count)**

```
Dim s As String="abcde"
Dim c As String
c=s.Remove(1,2) 'c= "ade"
```

Replace

- This method replaces a string pattern with another string wherever it finds this pattern in the string and returns resultant string

- Syntax: **Replace(old_string, new_string)**

```
Dim s As String="abcdeabc"
Dim c As String
c=s.Replace("abc","la") 'c= "ladela"
```

Compare()

- A String function to compare one string with another
- It returns an integer value
- It is case-sensitive
- **Syntax:** <variable>=String.Compare(arg1, arg2)
  - variable is an integer
  - arg1 and arg2 are String datatype
- Results:
  - -1 if arg1 is less than arg2
  - 0 if arg1 is equal to arg2
  - 1 if arg1 is greater than arg2

Compare: Example

```
Dim s1, s2 As String
Dim result As Integer
s1 = "hello"
s2 = "Hello"
result = String.Compare(s1, s2)
Console.WriteLine(result)
```

Output: -1

EndsWith() & StartsWith()

- These functions determine if a particular string ends or starts with a specified text.
- Returns a Boolean datatype
- It's case-sensitive
- **Syntax:**

  <variable> = arg.EndsWith("string_match")

  <variable> = arg.StartsWith("string_match")

  - Here, <variable> is of Boolean Datatype

Example

```
Dim s As String = "Hello"
Dim res As Boolean
res = s.StartsWith("hel")
Console.WriteLine(res)      ‘False
res = s.EndsWith("lo")
Console.WriteLine(res)      ‘True
res = s.StartsWith("Hel")
Console.WriteLine(res)      ‘True
```

Equals()

- This function determines if two strings are equal
- It is case-sensitive
- Returns a Boolean value as result
- **Syntax: <variable>=String.Equals(arg1,arg2)**
  - Here, variable is a Boolean
  - arg1 & arg2 are of String datatype

Example

```
Dim s1 As String = "computer"
Dim s2 As String = "Computer"
Dim res As Boolean
res = String.Equals(s1, s2)
Console.WriteLine(res)
```

Output: False

IndexOf(), LastIndexOf()

- These functions returns the position, which is an integer, at which a specified text is found within a string.
- The IndexOf() starts from the beginning of the string and proceeds forward
- The LastIndexOf() starts from the end of the string and works backwards

IndexOf() & LastIndexOf

- Syntax:
  - <variable>=str.IndexOf(arg)
  - <variable>=str.LastIndexOf(arg)

  - <variable>=str.IndexOf(arg, start)
  - <variable>=str.LastIndexOf(arg, start)

  - <variable>=str.IndexOf(arg, start, length)
  - <variable>=str.LastIndexOf(arg,start, length)

Explanation of syntax

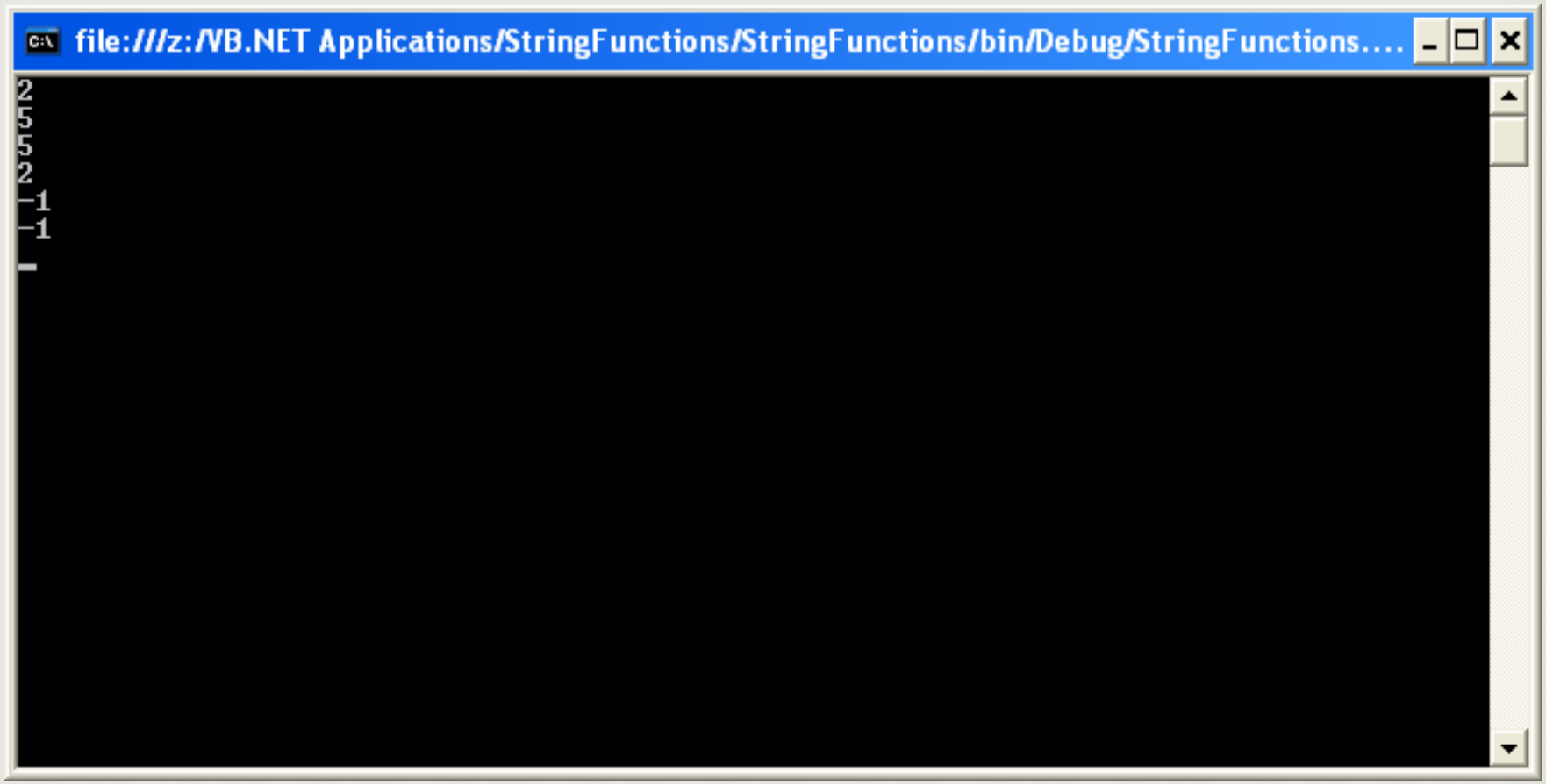
- **arg** is of type Char or String specifying text to look for in the String
- **start** is optional, specifying from where to begin the search
- **length** is optional, specifying how many characters to search

Example

```
Dim s As String = "mississippi"
Dim res As Integer
res = s.IndexOf("ss")
Console.WriteLine(res)
res = s.LastIndexOf("ss")
Console.WriteLine(res)
res = s.IndexOf("ss", 3)
Console.WriteLine(res)
res = s.LastIndexOf("ss", 5)
Console.WriteLine(res)
res = s.IndexOf("ss", 3, 2)
Console.WriteLine(res)
res = s.LastIndexOf("ss", 5, 2)
Console.WriteLine(res)
```

0	1	2	3	4	5	6	7	8	9	10
m	i	s	s	i	s	s	i	p	p	i

Output


file:///z:/VB.NET Applications/StringFunctions/StringFunctions/bin/Debug/StringFunctions....
2
5
5
2
-1
-1

Insert()

- This method inserts text into a string at a specified position and returns the new string. The original string is not changed.
- **Syntax: <variable>=str.Insert(position, arg)**
  - *variable* is a string
  - *str* is the string where insertion will be done
  - *position:* is the position where *arg* will be inserted

Example

```
Dim s As String = “V  Basic"
Dim res As String
res = s.Insert(1, “isual”)
Console.WriteLine(res)
```

OUTPUT: Visual Basic

Thankyou