

15.05.2020

ОБЪЕДИНЕНИЕ «ПРОГРАММИРОВАНИЕ»,

2-й год обучения

Тема:

«Работа со строковыми функциями в Visual Basic»

Цели:

1. Познакомиться с понятием строковые функции
2. Научить обучающихся использовать строковые функции в программном коде.
3. Развивать у учащихся способность к алгоритмическому мышлению
4. Развивать умение анализировать взаимосвязь между различными объектами проекта.
5. Формировать устойчивое внимание

Теоретические сведения:

<https://sitesaid.ru/VB/vb%20strok.html>

Для работы в Visual Basic со строками используется оператор объединения, называемый также оператором конкатенации, и встроенные функции. Список наиболее часто используемых функции для работы со строками приведен ниже:

Функции Str и Val

Функции, удаляющие пробелы.

Выделение подстроки.

Преобразование строки.

Определение положения строки.

Объединение строк.

Функция	Назначение
As C	Возвращает ASCII-код символа
Chr	Преобразовывает ASCII-код в символ
InStr, InStrRev	Осуществляют поиск одной строки в другой
LCase	Изменяет регистр букв исходной строки на нижний
Left	Возвращает указанное количество символов с начала строки
Len	Возвращает количество символов в строке

LTrim, RTrim, Trim	Удаляют пробелы, расположенные соответственно в начале, в конце и с обеих сторон символьной строки
Mid	Возвращает заданное количество символов из произвольного места строки
Right	Возвращает указанное количество символов с конца строки
Str, CStr	Преобразовывают числовое выражение в строку
StrReverse	Изменяет порядок следования символов в строке на обратный
StrConv	Изменяет регистр букв символьной строки
Val	Преобразовывают строку в числовое выражение
UCase	Изменяет регистр букв исходной строки на верхний

Функции Str и Val

Функция Str преобразовывает численное значение в символьное представление. Синтаксис функции следующий:

Str (число)

Функция Val () преобразовывает символьную строку в численное значение.

Синтаксис функции:

Val (символьноеВыражение)

При преобразовании строки символов в число учитываются все цифровые символы, расположенные в строке слева направо. Пробелы, находящиеся в начале и конце символьной строки, игнорируются. Пробелы внутри строки недопустимы. Если первый символ выражения не является цифрой, функция Val возвратит значение ноль.

Рассмотрим следующий пример. Зададим в окне Immediate две символьные строки a и b следующего вида:

a="10"

b="12"

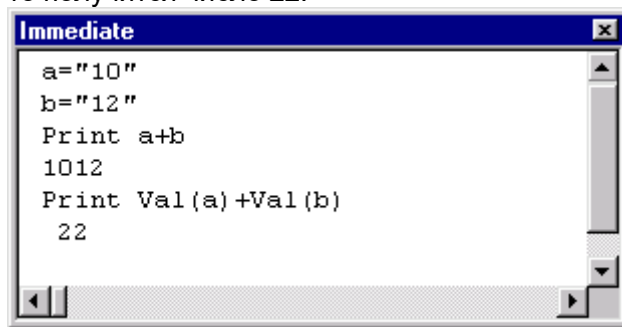
При сложении этих двух строк получается символьная строка

"1012"

Если сложить две символьные строки, предварительно преобразовав их в числа, и распечатать с помощью команды,

Print Val (a) + Val (b)

то получится число 22.



Функции, удаляющие пробелы в символьной строке

Функции LTrim, Rtrim и Trim используются для удаления пробелов в символьной строке.

Функция	Назначение
LTrim	Удаляет пробелы, расположенные в начале символьной строки
RTrim	Удаляет пробелы, расположенные в конце символьной строки
Trim	Удаляет пробелы, расположенные в начале и в конце символьной строки

Пример использования этих функции приведен ниже:

```
cCoinment= " Удаление пробелов "  
Print LTrim(cComment) ' Возвращает "Удаление пробелов "  
Print RTrim(cCornment) ' Возвращает " Удаление пробелов"  
Print Trim(cComrnent) ' Возвращает "Удаление пробелов"
```

Выделение подстроки

Вы можете выделить подстроку заданной символьной строки, используя функции Left, Right и Mid. Функции Left и Right выделяют строку начиная с крайнего левого или крайнего правого символа, а функция Mid позволяет выбрать любую подстроку.
Синтаксис функций:

```
Left(выражение, числоСимволов)  
Right(выражение, числоСимволов)  
Mid(выражение, номерПозиции [, числоСимволов ])
```

Ниже приведены примеры использования этих функций и возвращаемые ими значения:

```
cComment= "Выделение подстроки"  
Print Left(cComment,3) ' Возвращает "Выд"  
Print Right (cComment, 6) ' Возвращает "строки"  
Print Mid (cComment, 11,3) ' Возвращает "под"
```

Преобразование строки

Функции UCase () и LCase () используются в Visual Basic для преобразования строчных символов в заглавные и заглавных в строчные. Помимо этого, в Visual Basic имеется функция StrConv, которая преобразовывает выражение к имени собственному, начинающемуся с заглавной буквы.

Функции UCase () и LCase () возвращают значения, имеющие тип Variant. Для того чтобы возвращаемое значение имело тип string, необходимо использовать функции UCase\$ () и LCase\$ ().

Функция UCase

Функция ucase преобразует все строчные буквы в символьной строке в заглавные. Синтаксис функции: UCase (символьнаяСтрока) Например: cComment = "вывод" Print UCase(cComment) ' Возвращает "ВЫВОД" Print UCase\$(cComment) ' Возвращает "ВЫВОД"

Функция LCase

Функция LCase возвращает заданную символьную строку, в которой все заглавные буквы преобразованы в строчные. Синтаксис функции: LCase(символьнаяСтрока) Например: cComment= "ВЫВОД" Print LCase(cComment) ' Возвращает "вывод" Print LCase\$(cComment) ' Возвращает "вывод"

Функция StrConv

Функция StrConv преобразовывает выражение, написанное строчными или заглавными буквами, в имя собственное. Например: cComment = "Петр Петрович Петров" Print StrConv(cComment,vbProperCase) ' Возвращает "Петр Петрович Петров" Аналогичный результат будет получен и в следующем случае: cComment = "ПЕТР ПЕТРОВИЧ ПЕТРОВ" Print StrConv(cComment, vbProperCase) ' Возвращает "Петр Петрович Петров"

Определение положения строки в другой строке

Visual Basic содержит две функции, позволяющие осуществлять поиск символьной строки в другой: InStr () и InStrRev (). Эти функции отличаются тем, что InStr () осуществляет поиск с начала строки и до ее конца, а InStrRev () проводит поиск в обратном направлении, то есть от конца строки к началу.

Функция InStr () имеет следующий упрощенный синтаксис:

InStr (исходнаяСтрока, строкаПоиска)

В результате проведенного поиска функция возвращает число, указывающее номер позиции первого вхождения строки.

Рассмотрим такой пример. Введите в окне Immediate следующую команду:

```
Print InStr ("Сегодня прекрасная погода", "погода")
```

В результате будет возвращено число 20.

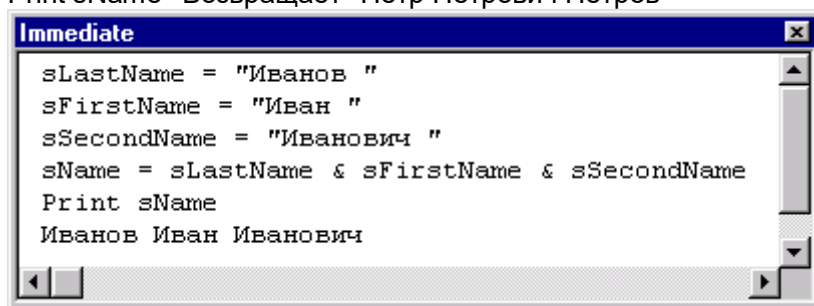
Объединение строк

В Visual Basic для работы со строками можно использовать только один оператор — оператор объединения. С помощью данного оператора можно объединять несколько строк в одну. Этот оператор обозначается символом амперсанда (&).

В предыдущих версиях Visual Basic для объединения строк использовался символ "плюс" (+). В Visual Basic 6 этот оператор также поддерживается.

Например, объединение строк удобно использовать при формировании полного адреса, если известен индекс, город и улица. В следующем примере и на рис. 5.7 показан результат объединения фамилии, имени и отчества в окне Immediate:

```
sLastName = "Петр "  
sFirstName = "Петрович"  
sSecondName = "Петров"  
sName = sLastName & sFirstName & sSecondName  
Print sName ' Возвращает "Петр Петрович Петров"
```



Практическое задание:

Разобрать вышеприведенные функции в среде программирования Visual Basic