

Лекция 14

Символьные константы и переменные, операции с символами

Строковые константы и переменные

- Компьютер работает не только с числами, но и с символами.
- Последовательность символов, используемых в программе, называется строкой.
- Символьные или строковые константы представляют собой набор произвольных символов, принадлежащих языку Бейсик и заключенных в кавычки .
- Языки программирования имеют специальные средства для работы со строками.

Особенности работы со строками

1. Для того, чтобы переменная могла *иметь строковое значение*, ее имя необходимо описать:

Dim name **As String** или **Dim** name **As String [10]**

или имя может заканчиваться символом доллара: имя\$. Верно и обратное: если имя заканчивается \$, то переменная должна иметь строковое значение.

2. *Значение строковой переменной* тоже можно задавать оператором присваивания, но в правой части должно стоять не арифметическое, а *строковое выражение*, т.е. выражение, значением которого являются строки.
3. В такое выражение могут входить переменные и *строковые константы*. В языке VBA такие константы заключаются в кавычки.

Особенности работы со строками

4. Операции со строками отличаются от действий, которые можно выполнять с числами. Нелепо говорить о сложении или вычитании строк, тем более об умножении или делении. Но можно говорить о слиянии строк. Такая операция называется конкатенацией \equiv слияние строк.
5. Она объединяет две строки вместе (2-я присоединяется к 1-й) и обозначается знаком «+» или «&».
6. В отличие от Qbasic на VBA использование постфикса \$ не выделяет в ОП отдельной ячейки, но постфикс позволяет менять тип переменной внутри одной программы
 - Например: **b\$ = "two": b = 2: MsgBox b\$ & b**



Примеры

- строковых констант

"А", "Заяц", "F(X)", "!#%^&*_+()\$"

- строковых выражений:

" А " + " В " \Rightarrow " АВ "

- оператора присваивания

A\$="ИВАНОВ_": B\$="ИВАН_": C\$="Иванович"

FIO\$=a\$+b\$+c\$' можно не использовать знак \$

FIO \Rightarrow ИВАНОВ ИВАН ИВАНОВИЧ

Особенности работы со строками

7. Неправильное написание операторов (будет обнаружено при выполнении программы)
 $c\$ = 2 + \text{"ТЕХТ"}$
нельзя «сливать» число и текстовую константу
8. При попытке присвоить текстовой переменной результат арифметической операции результат будет преобразован в строковое выражение числа, например:
 $\text{Stroka\$} = \text{Sin}(3) + 2$



Особенности работы со строками

9. Операции сравнения символьных переменных осуществляется **по их коду**.
10. Следует иметь в виду, что коды латинских букв и коды цифр упорядочены по возрастанию и по алфавиту в соответствии с кодами таблицы ASCII:
 $0 < 1 < \dots < 9 < A < B < C \dots < Z < a < b < c \dots < A < Б \dots$
11. Коды русских букв имеют более сложный порядок следования. Но, в любом случае, коды возрастают в алфавитном порядке.

ASCII

Для кодирования символов используется: **ASCII код** (American Standard Code for Information Interchange) – стандартный код для информационного обмена,

который введен в действие

Институтом стандартизации США – **ANSI** (American National Standard Institute).

Таблица 3

Символы с кодами 128–255 (Кодовая таблица 866 – MS-DOS)

Код	Символ	Код	Символ	Код	Символ	Код	Символ
128	А	160	а	192	Л	224	р
129	Б	161	б	193	л	225	с
130	В	162	в	194	Т	226	т
131	Г	163	г	195	т	227	у
132	Д	164	д	196	—	228	ф
133	Е	165	е	197	†	229	х
134	Ж	166	ж	198	‡	230	ц
135	З	167	з	199	‡	231	ч
136	И	168	и	200	£	232	ш
137	Й	169	й	201	ƒ	233	щ
138	К	170	к	202	£	234	ъ
139	Л	171	л	203	ƒ	235	ы
140	М	172	м	204	ƒ	236	ь
141	Н	173	н	205	=	237	э
142	О	174	о	206	ƒ	238	ю
143	П	175	п	207	£	239	я
144	Р	176	■	208	£	240	ѐ
145	С	177	■	209	ƒ	241	ё
146	Т	178	■	210	ƒ	242	є
147	У	179		211	ƒ	243	ё
148	Ф	180		212	£	244	ї
149	Х	181		213	ƒ	245	і
150	Ц	182		214	ƒ	246	ŷ
151	Ч	183		215	ƒ	247	ÿ
152	Ш	184		216	ƒ	248	•
153	Щ	185		217	ƒ	249	•
154	Ъ	186		218	ƒ	250	•
155	Ы	187		219	■	251	√
156	Ь	188		220	■	252	№
157	Э	189		221	■	253	¤
158	Ю	190		222	■	254	■
159	Я	191		223	■	255	■

Таблица 2

Символы с кодами 32–127

Символ	Код	Символ	Код	Символ	Код	Символ	Код
пробел	56	В	80	Р	104	Н	
!	57	9	81	Q	105	I	
"	58	:	82	R	106	J	
#	59	;	83	S	107	K	
\$	60	<	84	T	108	L	
&	61	=	85	U	109	m	
'	62	>	86	V	110	n	
(63	?	87	W	111	o	
)	64	@	88	X	112	p	
*	65	A	89	Y	113	q	
+	66	B	90	Z	114	r	
,	67	C	91	[115	s	
-	68	D	92	\	116	t	
.	69	E	93]	117	u	
/	70	F	94	^	118	v	
0	71	G	95	_	119	w	
1	72	H	96	`	120	x	
2	73	I	97	A	121	y	
3	74	J	98	b	122	z	
4	75	K	99	c	123	{	
5	76	L	100	d	124		
6	77	M	101	e	125	}	
7	78	N	102	f	126	~	
	79	O	103	g	127	del	

.asc .txt

Стандартные функции обработки символьных переменных

- При решении задач появляется необходимость выделить цепочку символов из строкового выражения: найти слово, часть слова, букву или цифру, любой символ в символьной переменной или символьном массиве.
- Для подобной обработки целесообразно использовать соответствующие функции обработки строковых данных.
- В дальнейшем договоримся, что при описании встроенных функций переменные **x\$** и **y\$** означают любые строковые выражения, а переменные **m,n** – выражения целого типа.

Стандартные функции обработки символьных переменных (2)

Len(x\$) – число символов в символьной строке, включая пробелы.

$Cells(1,1) = Len("ИНФОРМАТИКА") \Rightarrow 11.$

Instr([n],x\$,y\$) – осуществляет поиск подстроки y\$ в строке x\$, начиная с символа n

если n не указано, то поиск осуществляется с 1-го символа, например:

Name\$ = "ИНФОРМАТИКА"

$cells(1,2) = Instr(1,name$, "ФОРМА") \Rightarrow 3$

Обнаружив подстроку y\$ в x\$ функция *Instr* возвращает номер позиции 1-го символа подстроки. Если подстрока не найдена, то возвращается ноль.

Стандартные функции обработки символьных переменных (3)

Mid(x\$,n[,m]) – выделяет подстроку в m символов из строки $x\$$, начиная с символа n .

Если m опущено или кол-во символов $< m$, то выделяются символы, начиная с символа n до конца строки.

Если $m=0$ или $n > \text{длины}$ строки, то результатом выполнения функции будет пустая строка.

Пример: $a\$ = \text{Mid}(\text{"ПАЛИТРА"}, 3, 4)$

$b\$ = \text{MID}(\text{"«кон»"}, 4, 1)$

$a\$ = \underline{\text{ЛИТР}}$ $b\$ = \text{""}.$

Стандартные функции обработки символьных переменных (4)

Space(n) – формирует строку из n пробелов.

String(n,x\$) – формирует строку из n
одинаковых символов, равных первому
символу строки ***x\$***.

```
stroka$ = String(80, «*») + Space(10) + String(5, «abc»)
```

Строка будет иметь вид:

```
80 *** _____ ааааа
```

Стандартные функции обработки символьных переменных (4)

Lcase (x\$) – переводит все символы строки в строчные буквы

Ucase (x\$) – переводит все символы строки в прописные буквы.

Left (x\$,n) – выбираются *n* символов из строки слева

Right (x\$,n) – выбираются *n* символов из строки справа

Str (числовое выражение) – возвращает строковое представление числа (для положительных чисел слева добавляется пробел)

Val (строковое выражение) – превращает строковое представление числа в числовое. Если строка начинается с нецифрового знака (_ + - цифра), то результатом выполнения является нуль.

Пример:

”Поздравляем с Новым ” & Str(Val(Right(Date,4))+1) & ” Годом!”

- Date= «13-12-2014» => 2014=> 2014+1=>
- «Поздравляем с Новым 2015 Годом!»

Некоторые стандартные (не математические) функции VBA

- Функция **DATE** возвращает текущую системную дату компьютера в формате дд-мм-гггг
- Функция **TIME** возвращает текущее системное время компьютера. Оператор TIME\$ устанавливает текущее системное время в Вашем компьютере.

Cells(1,4)=TIME Будет выведено: 8:01:31

TIME = "08:00:58" 'Замечание: Новое системное время остается в силе, пока Вы снова не измените его.

cells(2,4)= "Установленное время" +TIME



Пример 1. Чему будет равен результат работы функции?

Function text (a as string, b as string) as

nl = Len(a)

For i = 1 To nl

i = InStr(i, a, b)

If i <> 0 Then GoTo 20

Next i

20 text = Right(a, nl - i + 1)

End Function

Поиск номера символа, равного b="U"

Если такой символ найден, то выход из цикла к метке с номером 20
В данном случае I = 25

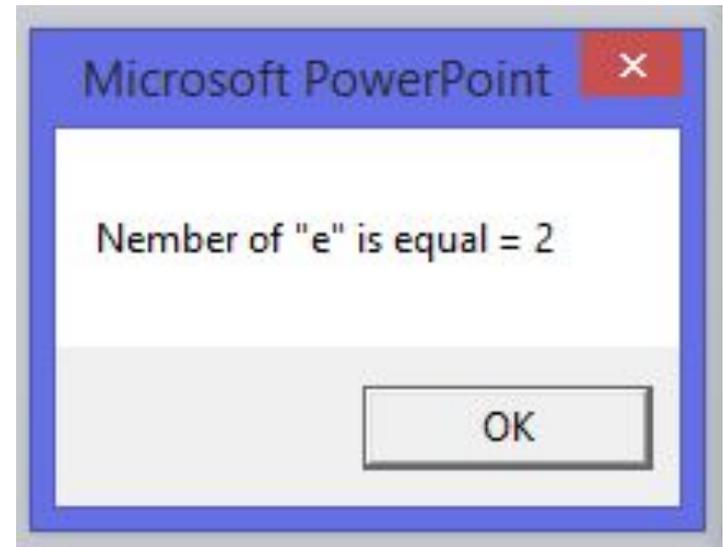
Выбираем справа из «a» 10 символов



Пример 2: Ввести фразу «Thy Will be done in Earth as it is in Heaven!»

Сколько букв "e" в фразе стоит на четных местах?

```
Sub test2()  
Dim text As String, nt As Byte  
text = "Thy Will be done in Earth as it is in Heaven!"  
nt = Len(text)  
ne = 0  
For i = 1 To nt  
ie = InStr(i, text, "e")  
If ie = 0 Then Exit For  
If ie Mod 2 = 0 Then ne = ne + 1  
i = ie  
Next i  
MsgBox "Number of ""e"" is equal = " & ne  
End Sub
```

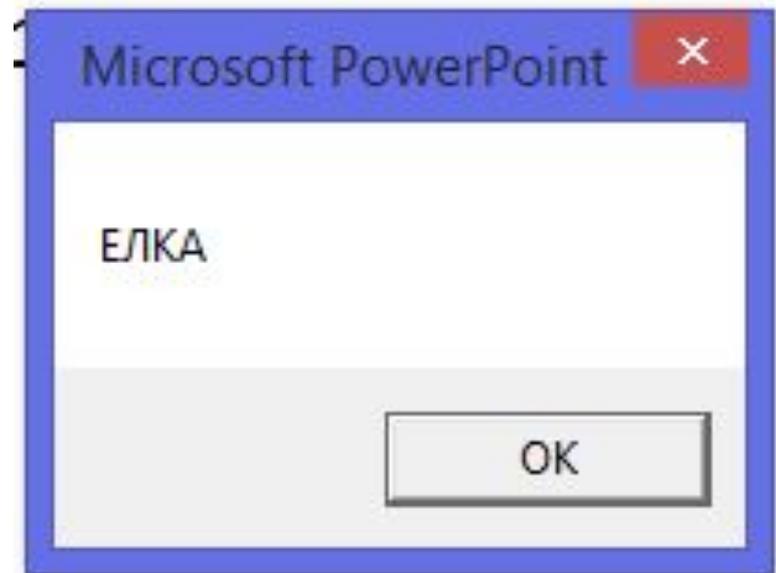


Пример 3:Расшифровать слово. Ключ к шифру: буквы закодированного слово стоят на каждой третьей позиции, начиная с конца закодированного слова.

- **Пример: Берем слово: «ЕЛКА». Вставим его по указанному ключу в слово «Новогодний»:**
- **ноАвоКгоЛднЕий**

```
Function decod_word(word As String) As String
Dim dw As String
dw = ""
nw = Len(word)
For i = nw - 2 To 1 Step -3
dw = dw & Mid(word, i, 1)
Next i
decod_word = dw
End Function
```

```
Sub main()
Dim cod_word As String
cod_word = "НОАВОКГОЛДНЕИЙ"
dec_word = decod_word(cod_word)
MsgBox dec_word
End Sub
```

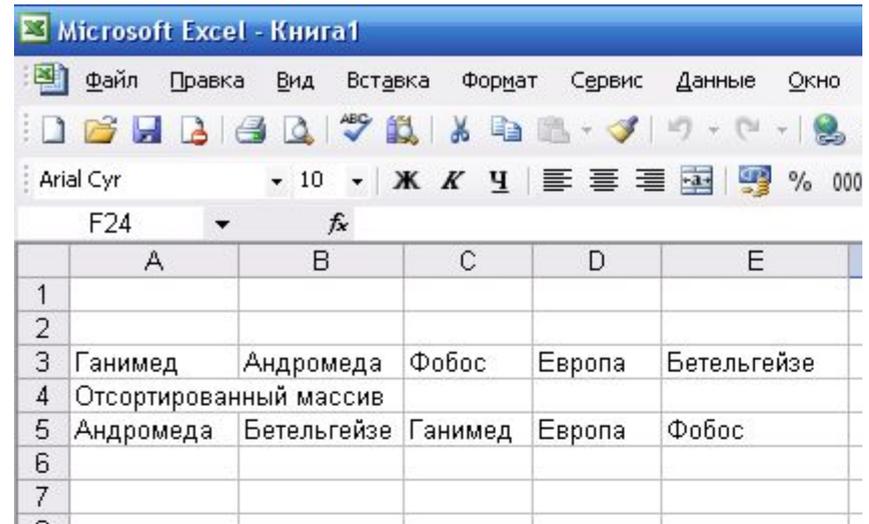


Пример 2. Программа сортировки СИМВОЛЬНОГО МАССИВА.

```
(General) | sorts
Sub main()
Dim a$( )
n = 5
ReDim a$(n)
Call reada(a$( ), n, 3)
Call sorts(a$( ), n)
Cells(4, 1) = "Отсортированный массив"
For i = 1 To n
Cells(5, i) = a$(i)
Next i
End Sub

Sub reada(a$( ), n, k)
For i = 1 To n
a$(i) = Cells(k, i)
Next i
End Sub

Sub sorts(a$( ), n)
k = 0
m3: For i = 2 To n - k
If a$(i - 1) > a$(i) Then
c$ = a$(i)
a$(i) = a$(i - 1)
a$(i - 1) = c$
End If
Next i
k = k + 1
If k < n + 1 Then GoTo m3
End Sub
```



The screenshot shows the Microsoft Excel interface with a spreadsheet containing the following data:

	A	B	C	D	E
1					
2					
3	Ганимед	Андромеда	Фобос	Европа	Бетельгейзе
4	Отсортированный массив				
5	Андромеда	Бетельгейзе	Ганимед	Европа	Фобос
6					
7					

Данная программа демонстрирует не только возможности взаимодействия п/п друг с другом, но и правило передачи массивов в п/п: имя массива указывается в качестве формального параметра с пустыми скобками.

Лекция закончена