

## Лабораторная работа №14 по дисциплине "Программирование"

ТЕМА: Символы и строки.

-----

### СОЗДАНИЕ И ИСПОЛЬЗОВАНИЕ ПАПОК

-----

1. В Вашей папке создайте папку с именем П\_лр14, совпадающим с именем файла с этим заданием, и затем скопируйте файл с заданием в эту папку.
2. При выполнении задания создавайте в папке П\_лр14 для каждой программы папку с именем, по смыслу соответствующим действиям, выполняемым составляемой программой, и сохраняйте все файлы, имеющие отношение к этой программе, в этой созданной папке (имена сохраняемых файлов должны отличаться от имен, присваиваемых по умолчанию, и должны по смыслу соответствовать действиям, выполняемым составляемой программой).
3. ПРИ ВЫПОЛНЕНИИ СЛЕДУЮЩИХ ЛАБОРАТОРНЫХ РАБОТ ДЕЙСТВУЙТЕ АНАЛОГИЧНО ПРЕДЫДУЩЕМУ.

### СИМВОЛЫ

-----

4. Символьный тип данных, обозначаемый при объявлении как Char, предназначен для представления любого символа из набора допустимых символов. Для каждого символа отводится один байт памяти.  
Пример объявления величин символьного типа:  
**Var** a, s: char;  
К символам можно применять операции отношения (<, <=, =, >, >=, <>), при этом сравниваются коды символов.
5. Некоторые стандартные функции для символьных величин:

Функция	Описание	Тип	
		аргумента	результата
Ord(s)	Код (порядковый номер) символа, например, ord('b') => 98	char	integer
Chr(n)	Преобразование в символ, например, chr(98) => 'b'	integer	char
Pred(s)	Предыдущий символ, например, pred('b') => 'a'	char	char
Succ(x)	Последующий символ, например, succ('b') => 'c'	char	char

### СТРОКИ

-----

5. Строки представляют собой последовательности символов.  
При объявлении строки обозначаются ключевым словом String с возможным указанием максимальной длины строки, например:  
**Var** s: string;  
t: string[30];
6. Строки можно присваивать друг другу. Если максимальная длина результирующей строки меньше длины исходной строки, то лишние символы справа отбрасываются, например:

```

Var s1: string[3];
    s2: string[6];
...
s2:='Summer';
s1:=s2;      //s1:='Sum'

```

7. Строки можно сцеплять (объединять, склеивать) между собой с помощью операции конкатенации, обозначаемой знаком +, например:

```

s1:='Sum';
s2:='mer';
s3:=s1+s2;      //s3:='Summer'

```

8. Строки можно сравнивать между собой с помощью операций отношения, при этом справедливы, например, следующие соотношения:

```

'abc'>'ab'
'abc'='abc'
'abc'<'abc '

```

9. При вводе и выводе может использоваться имя строки, например:

```

readln(s1,s2);
writeln(s3);

```

При вводе в строку считывается количество символов, равное длине строки или меньшее, если клавиша Enter будет нажата раньше.

При выводе для строки отводится количество позиций, равное ее фактической длине.

10. К отдельному символу строки можно обращаться как к элементу массива символов, например, s1[4].

Один символ из строки совместим с величиной типа Char и их можно использовать в выражениях одновременно.

#### НЕКОТОРЫЕ ПРОЦЕДУРЫ И ФУНКЦИИ ДЛЯ РАБОТЫ СО СТРОКАМИ

-----

11. Для реализации действий со строками и фрагментами строк предусмотрены стандартные процедуры и функции.

ВНИМАНИЕ! Фрагмент строки часто называют подстрокой.

ВНИМАНИЕ! Фраза "функция возвращает" практически всегда используется вместо фразы "результатом выполнения функции является".

Процедура или функция с именем и примером перечня аргументов	Описание
Функция Concat(s1,s2,...,sN)	Возвращает строку, являющуюся слиянием строк s1,s2,...,sN. Действие функции аналогично операции конкатенации.
Функция Copy(s,start,len)	Возвращает подстроку длиной len, начинающуюся с позиции start строки s.
Процедура Delete(s,start,len)	Удаляет из строки s, начиная с позиции start, подстроку длиной len.
Процедура Insert(subs,s,start)	Вставляет подстроку subs в строку s, начиная с позиции start.
Функция Length(s)	Возвращает фактическую длину строки.
Функция Pos(subs,s)	Ищет вхождение подстроки subs в строку s и возвращает номер первого символа subs в s или нуль, если subs не содержится в s.

Процедура Str(x, s)	Преобразует числовое значение x в строку s, при этом для x может быть задан формат, как в процедурах Write и Writeln, например Str(x:8:2, s).
Процедура Val(s, x, errcode)	Преобразует строку s, содержащую символьное представление числа, в значение числовой переменной x. В случае успешного преобразования значение переменной errcode равно нулю, при обнаружении ошибки в символьном представлении числа значение errcode будет содержать номер позиции первого ошибочного символа, а значение x не определено.
Функция IntToStr(n)	Преобразует целочисленное значение n в строку
Функция StrToInt(s)	Преобразует строку s в целое число
Функция FloatToStr(x)	Преобразует вещественное значение x в строку
Функция StrToFloat(s)	Преобразует строку s в вещественное число
Функция AnsiUpperCase(s)	Возвращает строку s, преобразованную к верхнему регистру.
Функция AnsiLowerCase(s)	Возвращает строку s, преобразованную к нижнему регистру.
Функция Trim(s)	Возвращает строку s с удаленными из нее пробелами и управляющими символами в начале и в конце строки
Функция TrimLeft(s)	Возвращает строку s с удаленными из нее пробелами и управляющими символами в начале строки
Функция TrimRight(s)	Возвращает строку s с удаленными из нее пробелами и управляющими символами в конце строки

Примеры выражений с использованием функций для работы со строками:

```
'Summ '+ FloatToStr(x)
20+StrToInt(s)
```

#### СОЗДАНИЕ ПРОГРАММ

-----

ВНИМАНИЕ! Все создаваемые здесь программы должны быть "зациклены", то есть после вывода результатов следует переходить снова на ввод данных.

#### ПРОГРАММА 14-1

-----

12. Создайте консольное приложение, при выполнении которого происходит ввод с клавиатуры одного любого символа и вывод его кода.

#### ПРОГРАММА 14-2

-----

13. Создайте консольное приложение, при выполнении которого происходит ввод с клавиатуры одного любого символа и целого числа строк, в которых затем будет выводиться этот символ.

После ввода должны выводиться строки, в каждой из которых должны находиться количество выводимых символов, пробел, введенный символ в количестве, указанном в начале этой же строки.

Количество выводимых символов должно меняться от 1 до введенного числа строк включительно (пример ввода и вывода см. ниже).

В программе должен быть предусмотрен вывод строки с соответствующим сообщением, если введенное число строк меньше или равно нулю.

Пример ввода и вывода для ПРОГРАММЫ 14-2

```
*
5
1 *
2 **
3 ***
4 ****
5 *****
```

ПРОГРАММА 14-3  
-----

14. Создайте консольное приложение, при выполнении которого происходит то же, что и при выполнении ПРОГРАММЫ 14-2, но вывод строк происходит в обратном порядке.

Пример ввода и вывода для ПРОГРАММЫ 14-3

```
*
5
5 *****
4 ****
3 ***
2 **
1 *
```

ПРОГРАММА 14-4  
-----

15. Создайте консольное приложение, выполняющее следующие действия:
- а) вывод слова "Месяц" на английском языке и затем ввод в этой же строке слова на английском языке, которое может совпадать с названием какого-либо месяца;
  - б) если введенный текст совпадает с названием какого-либо месяца, то вывод названия времени года, к которому относится этот месяц, на английском языке  
(если месяц январь, то в строке с названием времени года дополнительно вывод пожелания счастливого Нового года на английском языке)  
и затем переход на начало следующей строки;
  - если введенный текст не совпадает с названием какого-либо месяца, то вывод текста "----" и затем переход на начало следующей строки;
  - в) переход к пп."а)" на вывод слова "Месяц" на английском языке и ввод текста и т.д..

ПРОГРАММА 14-5  
-----

16. Создайте консольное приложение, при выполнении которого происходит ввод на английском языке в отдельных строках фамилии, имени и отчества, начинающихся с больших букв:

**Familia**

**Imia**

**Otchestvo**

и затем происходит формирование и вывод строки следующего вида:

**Familia I.O.**

ПРОГРАММА 14-6

-----

15. Создайте консольное приложение, при выполнении которого происходит ввод на английском языке в отдельных строках фамилии, имени и отчества, содержащих или не содержащих пробелы впереди, внутри и сзади в этих строках, начинающихся с больших или маленьких букв и содержащих большие или маленькие буквы:

**fAM iLia**

**Im iA**

**otcheStvo**

и затем происходит формирование и вывод строки следующего вида:

**Familia I.O.**

ОТЧЕТ О ВЫПОЛНЕНИИ РАБОТЫ

-----

16. Продемонстрируйте тексты созданных программ, а также их выполнение в среде программирования и независимо от нее.
-