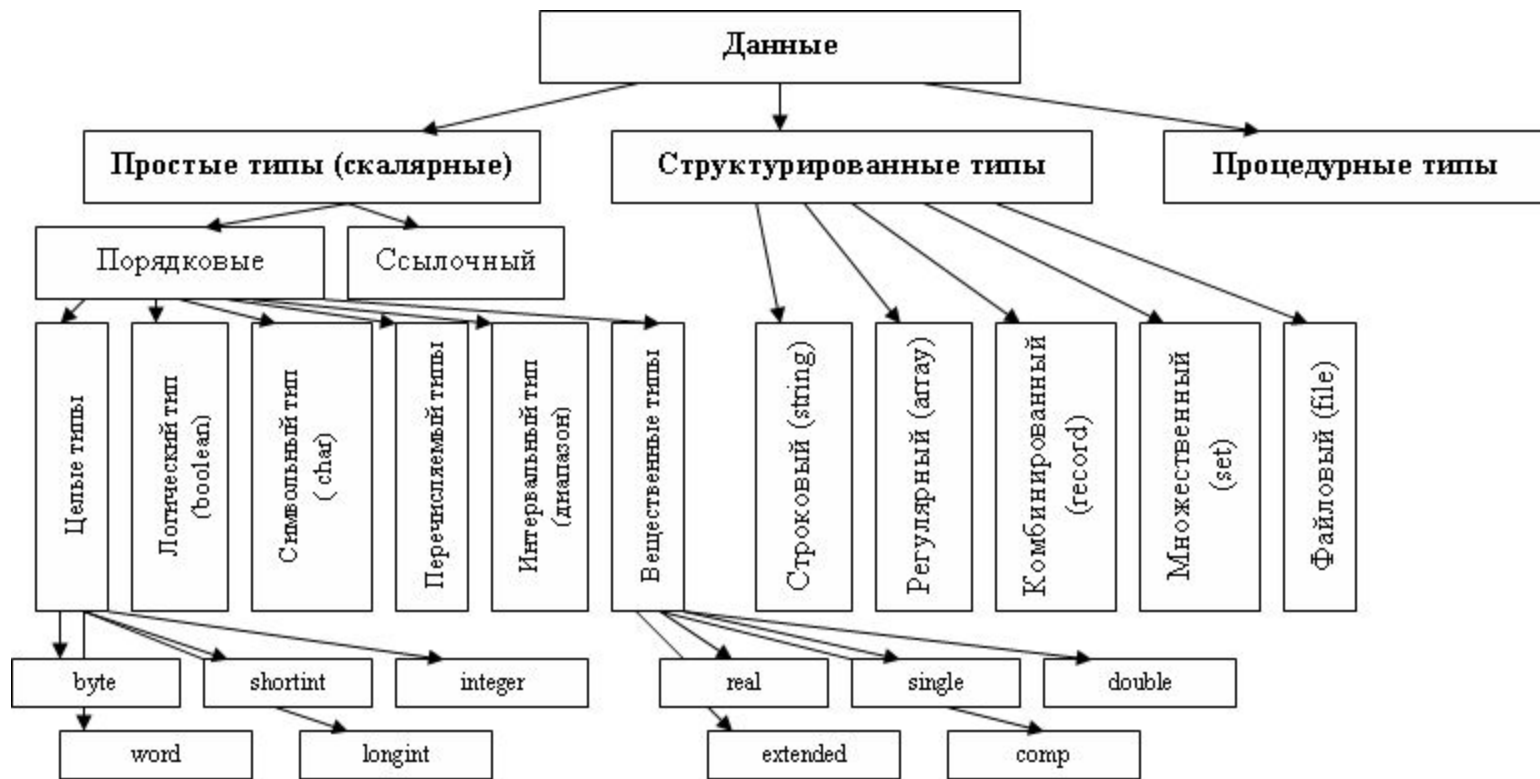


Строковый тип данных



Основные сведения о символьных величинах

1. Текст – это произвольная последовательность символов некоторого алфавита.
2. Алфавитом может служить любое множество символов, например, (0,1...9), (А,Б..., а,...), (А,В..., а, b,..).
3. Строкой символов, или символьной (строковой, текстовой) константой, называется последовательность символов, заключенных в кавычки. Максимальная длина – 255 символов.
4. Строка, не содержащая ни одного символа называется пустой строкой или строкой нулевой длины.

Типы переменных для работы с символами

1. Символьный тип данных: Char.

- Описание Var S:char.
- Диапазон значений: любой символ (один!) – буквы, цифры, знаки препинаний и специальные символы. Каждому символу соответствует индивидуальный числовой код от 0 до 255.
- Значения для переменных типа char задаются в апострофах, например: ch:='*'; a:='3'; letter:='G'; rus:='ф'.

Типы переменных для работы с символами

2. Строковый тип данных: *String*.

- Строкой называется последовательность символов определенной длины.
- Описание

```
Var Str1:string[30]; Str2: string.
```

**Стандартные функции
для работы с
СИМВОЛЬНЫМИ
величинами**

Операция сложения (склеивания)

- Позволяет строить из двух символьных строк третью, состоящую из символов первой строки, за которой следуют символы второй строки.

1. Обозначение: знаком «+»

- *Пример:*

```
var str1, str2, str3: string(20);  
begin  
  str1:='У Егорки';  
  str2:='всегда отговорки';  
  str3:=str1+' '+str2;
```

В результате значение строки str3: «У Егорки всегда отговорки».

Операция сложения (склеивания)

2. Функция `concat(str1,str2,...strn)`

(где n – произвольное количество) – *функция сцепления строк*. Аргументами могут быть как имена строк, так и сами строки.

Пример:

а) Результат `concat(str1,str2)` будет строка :
«У Егорки всегда отговорки»,

б) `concat('ab','cd','ef')` - получится строка 'abcdef'

Что получится в результате выполнения программы?

```
program slovo;  
    uses crt;  
    var a, b, c, d, k, l, m, n: string;  
  
begin  
    clrscr;  
    a:= 'пар';  
    b:= 'ом';  
    c:= 'с';  
    d:= 'ад';  
    k:= a + b;   l:= c + b;   m:= c + d;   n:= a + d;  
    write (k, ' ', l, ' ', m, ' ', n );  
    readln  
  
end.
```

Сравнение

- Сравнение строк выполняется посимвольно в соответствии с их кодами до первого несовпадения. Если одна из строк закончилась до первого несовпадения, то она считается меньшей. Пустая строка меньше любой строки.
- Две строки называются **равными**, если они равны по длине и совпадают посимвольно.
- 'Balkon' < 'balkon'
- 'balkon' > 'balken'
- 'balkon' > 'balk'

Можно использовать любые сравнения и их комбинации в условных операторах (>, >=, <, <=, =, <>).

Доступ к отдельному символу

- Для доступа к отдельному символу в строке необходимо указать имя строки и в квадратных скобках номер позиции элемента (символа) в строке.
- По отношению к отдельному символу строки возможны все те же операции, что и по отношению к переменной типа Char.

Длина строки

- **Длина строки** – это количество введенных символов, не может превышать максимально возможной длины, указанной в описательной части.
- Значение длины определяется при помощи функции ***Length(...)***, результат которой целое число, равное количеству символов.

• *Пример:*

```
Str1:='ABCDEFGH';
```

```
Str2:='Мама мыла раму';
```

```
k1:= Length(str1);
```

```
K2:= Length(str2);
```

```
Результат: k1=8; K2=14.
```

Копирование

- Функция *copy(str,n,m)* – копирует *m* символов строки *str*, начиная с *n*-го символа. При этом исходная строка не меняется.

Результат можно присваивать другой строке или сразу выводить на экран.

Пример: Str1:=‘ABCDEFGH’;

Str2:=‘abcdefgh’;

Str3:=copy(str1,4,3);

Writeln(str3); *Результат:* str3=‘DEF’

Writeln(copy(str2,4,3)); *Результат:* ‘def’

Что получится в результате выполнения программы?

```
program primer;  
    var a, b, c, d: string;  
begin  
    a:= 'лекарство';  
    b:= copy (a, 8, 1) + copy (a, 4, 1) + copy (a, 1, 2) + copy (a,  
7, 1);  
    c:= copy (a, 3, 1) + copy (a, 5, 1) + copy (a, 2, 1) + copy (a,  
6, 1) + copy (a, 1, 1) + copy (a, 9,1);  
    d:= copy (a, 6, 1) + copy (a, 2,1) + copy (a, 7, 1) + copy (a,  
3, 2);  
    write (b, ' ', c, ' ', d);  
end.
```

Удаление

- Используется процедура *Delete(str,n,m)*, которая вырезает из строки *str* *m* СИМВОЛОВ, начиная с *n*-го; при этом сама строка ИЗМЕНЯЕТСЯ.

• *Пример:*

Str1 := 'ABCDEFGH';

Delete(str1,3,4);

Результат: Str1 = 'ABGH'

Замена (вставка)

- Вставку строки *Str1* в строку *Str2*, начиная с *n*-го символа осуществляет процедура *Insert(Str1,Str2,n)*, при этом первая строка не изменяется, а вторая получает новое значение.

Пример: Str1:=‘ABCDEFGH’;
Str2:=‘abcdefgh’;
Insert(str1,str2,3);

Результат: Str2=‘abABCDEFGHcdefgh’

Что делает данная программа? Каков результат её выполнения?

```
program primer1;  
var  a, b, c, d : string;  
begin  
    a:= 'лекарство';  
    delete (a, 3, 5);  
    writeln (a);  
    delete (a, 3, 1);  
    b:= 'т';  
    insert (b, a, 3);  
    writeln (a);  
    c:= 'кот';  
    d:= 'мпо';  
    insert (d, c, 3);  
    writeln (c);  
end.
```

Подстрока

- Позицию подстроки *str1* в строке *str2* определяет функция *pos(str1, str2)*.
- Результат – целое число, которое определяет номер первого элемента, с которого начинается первое вхождение подстроки в строку. Если такой подстроки нет, то значение функции равно 0.

Пример: Str1:='CDE';
Str2:='ABCDEFGH';
K1:=pos(str1, str2);
K2:=pos(str2, str1);

Результат: K1=3; K2=0

Числа и строки

1. Процедура ***Str(N, str1)*** переводит числовое значение N в строковое и присваивает результат строке ***str1***, причем можно переводить любые числа.

• *Пример:* `Str(1234, str1);` *Результат* `str1:=‘1234’`

2. Функция ***val(str, N, K)*** переводит строковое значение в числовое.

Если строка **действительно** является записью числа, то значение $K=0$, а N – это число; иначе K будет равно номеру символа, в котором встречается первое нарушение записи числа N .

• *Пример:*

`val(‘1234’, N, K);` $N=1234$, $K=0$

`val(‘12d34’, N, K);` $N=0$, $K=3$

Функции преобразования типов

1. Функция определения символа по числовому коду в ASCII – **chr(x)**.

X – тип integer (byte), результат – тип char.

Пример: `chr(65) = 'A'`

2. Функция определения числового кода символа в ASCII – **ord(x)**.

X – тип char, результат – тип integer.

РЕШЕНИЕ ЗАДАЧ

Задача 1

- В заданной строке A заменить выделенную подстроку на подстроку B. Для поиска позиции, с которой надо произвести замену, воспользоваться функцией поиска подстроки.
- A:= 'Выдать премию в размере месячного оклада'
- B:= 'годового'

Задания для самостоятельной работы.

Выполнить преобразования слов:

- А) груша – гроза – проза – проба – сдоба – свара
- Б) бисер – бомба – балка – палка – палец
- В) метка – сетка – седло – сопло – совет – кювет
- Г) нитка – нерка – корка – кирка – книга – пурга