

***Работа с
символьной
информацией***

Вычислительные машины имеют дело не только с числами. Едва ли не больше времени они бывают заняты обработкой текста.

Типы данных, работающие с
символьной информацией

```
graph TD; A[Типы данных, работающие с символьной информацией] --> B[Символьный тип Char]; A --> C[Строковый тип String];
```

Символьный тип
Char

Строковый тип
String

Символьный тип - буквы, цифры, знаки препинания и специальные символы

- ✓ *Объявление символов как переменных*

var

y:char;

или констант

const

x='';*

- ✓ *Использование в теле программы*

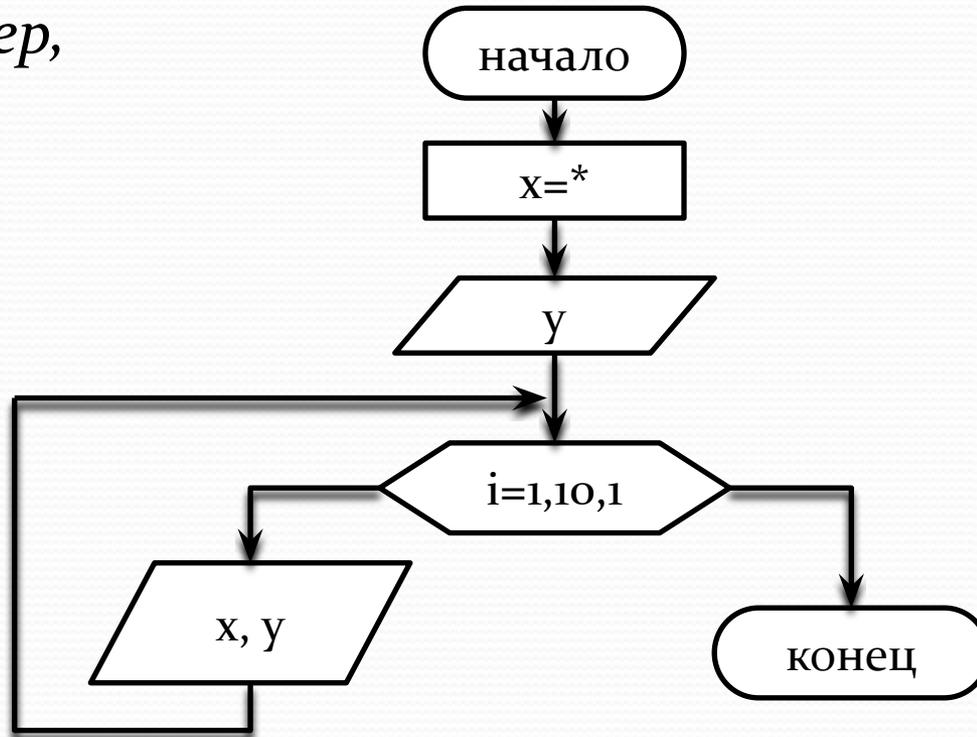
y:='з';

Задача: вывести на экран таблицу 2×10 , где в каждой строке первый элемент – «*», а второй произвольно вводимый с клавиатуры.

Например,

```
* #  
* #  
* #  
* #  
* #  
* #  
* #  
* #  
* #  
* #  
* #  
* #
```

Блок-схема



Листинг программы

```
Program chartype;  
uses crt;  
const  
    x='*';  
var  
    y:char;  
    i:byte;  
begin  
    writeln('Введите символ');  
    readln(y);  
    clrscr;  
    for i:=1 to 10 do  
        begin  
            writeln(x, ' ', y);  
        end;  
    end.
```



Строковый тип -

последовательность символов определенной длины
(экономит память!)

✓ Объявление строк

`var`

`s1: string[m];` // *m-максимально количество символов строки s1*

`s2: string;`

✓ Использование в теле программы

`s1 = 'Тише воды, ';`

Стандартные функции для работы со строковым типом данных

1. *Операция сложения символьных величин – «+»*

```
s1: = 'Тише воды, ';
```

```
s2:= 'ниже травы';
```

```
s3:=s1+''+s2;
```

2. *Длина строки – «Length»*

```
s2:= 'Семеро одного не ждут';
```

```
k2:=Length(s2).
```

3. *Копирование – «copy»*

```
s1:='паровоз';
```

```
s2:='123456';
```

```
s3:=copy(s1, 5, 3);
```

```
writeln(s3);
```

```
writeln(copy(s2, 3, 2));
```

4. *Удаление – «delete»*

```
s:='123456';  
delete(s, 3, 2);  
writeln(s);
```

5. *Замена (Вставка) – «insert»*

```
s1:='34':  
s2:='1256':  
insert (s1, s2, 3);
```

6. *Преобразование числа в строку*

```
n:=12;  
str(n,s1);
```

7. *Функции преобразования типов*

a) CHR – выводит символ соответствующий ASCII-коду –
используется, если символ отсутствует на клавиатуре

```
for i:= 0 to 255 do  
writeln( i, ' ', chr(i));
```

b) ORD – функция, обратная CHR

```
readln(s);  
writeln(ord(s));
```

8. Использование буфера обмена

- a) Readkey – записывает значение нажатой клавиши в буфер обмена, не выводя ее на экран.
- b) Uppcase – перевод строчных букв в заглавные.

```
uses crt;  
var  
  ch,gh:char;  
begin  
  writeln('Введите строчные буквы, для завершения нажмите Z');  
  repeat  
    ch:=readkey;  
    write(Uppcase(ch));  
  until ch='z';  
end.
```