



# Информатика

Лекция 9

Символы и строки  
(типы и структуры данных)

# Символьный тип

- Значениями являются символы из множества ASCII (American Standard Code For Information) включающего 256 различных символов
- Для символов с графическим представлением используются апострофы '2' | 'a' | 'X' | ':' | '"' (сам апостроф)
- При отсутствии графического представления используется диес #  
#10 – код символа перевода строки  
#\$A – шестнадцатеричный код символа
- Управляющие символы также имеют форму: ^C  
^G - символ с кодом 8 (звонок)  
^[ - эквивалент #\$1B (#27 - ESC)

# Операции над символами

- Определение – `var a, b, x, y, u : char;`
- Ввод/вывод – `read(a); write(a);`
- Присваивание - `a := 'f'; a := b;`
- Функция `ord(b)` – порядковый номер символа  
`ord('b')` имеет значение 98  
`chr(i)` – *i*-ый символ множества ASCII  
`chr(98)` представляет строчный символ – *b*  
цифра 0 имеет номер 48 – `ord('0') = 48`
- Применимы операции сравнения/отношения  
`< | ≤ | = | ≠ | ≥ | >` – результаты булевский тип  
`pred(a) = chr(ord(a) - 1); succ(a) = chr(ord(a) + 1);`

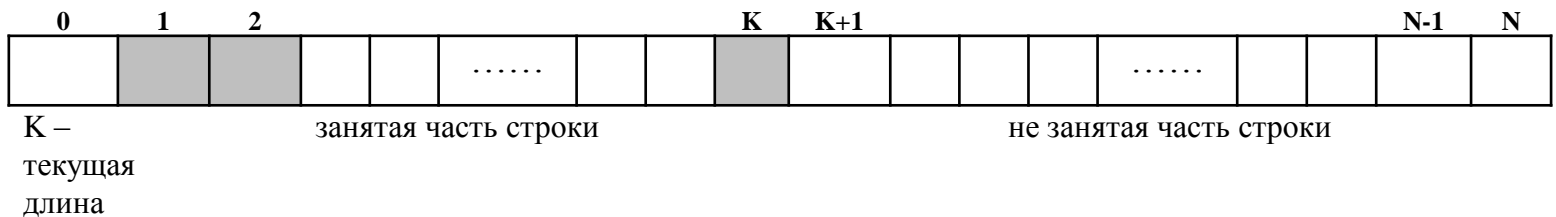
```
var c: char;  
for c := 'a' to 'z' do write (c);
```

# Строковый тип

- Строковый тип данных определяет множество символьных цепочек произвольной длины (от нуля символов до заданного их числа)
- Определение – **type** Line : string [80];  
**var** MyLine : Line;
- Присваивание значения –  
MyLine := "Пример строки";  
MyLine := AnotherLine;  
Readln ( F, MyLine);
- Указание максимальной длины можно опустить, по умолчанию максимальная длина строки имеет 255 символов

# Операция конкатенации

- Строки могут динамически изменять свою длину (присоединение строк)  
MyLine := "короткая строка";  
MyLine := MyLine + " стала длиннее";
- Для описания строковой переменной длиной N символов отводится N+1 байтов памяти:  
нулевой байт – длина строки,  
N – байтов хранят символы



# Ограничение длины

- Доступ к отдельным элементам строки через выражение целого типа – MyLine[номер байта] данная конструкция имеет символьный тип и является переменной, т.е. может находиться в левой части оператора присваивания
- Если строковой переменной присваивается выражение с длиной больше максимальной, то происходит укорачивание строки

```
var ShortStr : string [5];
```

```
begin
```

```
ShortStr := "очень длинная строка";
```

```
writeln (ShortStr);  {напечатает очень}
```

# Сравнение строк

- Элементы строк нумеруются целыми числами, начиная с 1 (нулевой байт – текущая длина строки):  
MyLine := ""; {пустая строка}  
writeln (Ord(MyLine[0])); {число 0}  
MyLine := "ABCD"  
writeln (Ord(MyLine[0])); {число 4}  
MyLine[0] := 2;  
writeln (MyLine); {строка AB}
- Над строками определены операции сравнения  $< | \leq | = | \neq | \geq | >$  (результата булевский тип):
  - более короткая – меньше более длинной;
  - для равных строк поэлементное сравнение символов с учетом лексикографической упорядоченности значений символьного типа

# Функции над строками

- Слияние (конкатенация)  
`concat( s1 [,s2, .. , sn]: string) : string`
- Возвращение подстроки  
`copy(var S : string; index, count : integer) : string`
- Удаление подстроки  
`delete(var S : string; index, count : integer) : string`
- Текущая длина строки  
`length(var S : string) : integer`
- Поиск позиции в строке  
`pos(substr, var S : string) : byte`
- Вставка подстроки  
`insert(source, var S : string, index : integer) : string`
- Преобразование символьного представления числа  
`val( s : string; var V, var Code : integer)`
- Преобразование числа в последовательность символов  
`str( X [: M [: N]]; var S : string)`



# Задача

*Содержательная постановка.* Даны алфавит и строка (цепочка) символов . Определить «правильность» строки. Строка считается правильной, если каждый символ строки является символом данного алфавита:

$$A = \{a, b, c\}$$

$S_1 = \text{"aacbac"};$  – строка правильная

$S_2 = \text{"axbucz"};$  – строка неправильная

*Формальная постановка.*

Дан алфавит

$$A = \{a_1, \dots, a_n\}$$

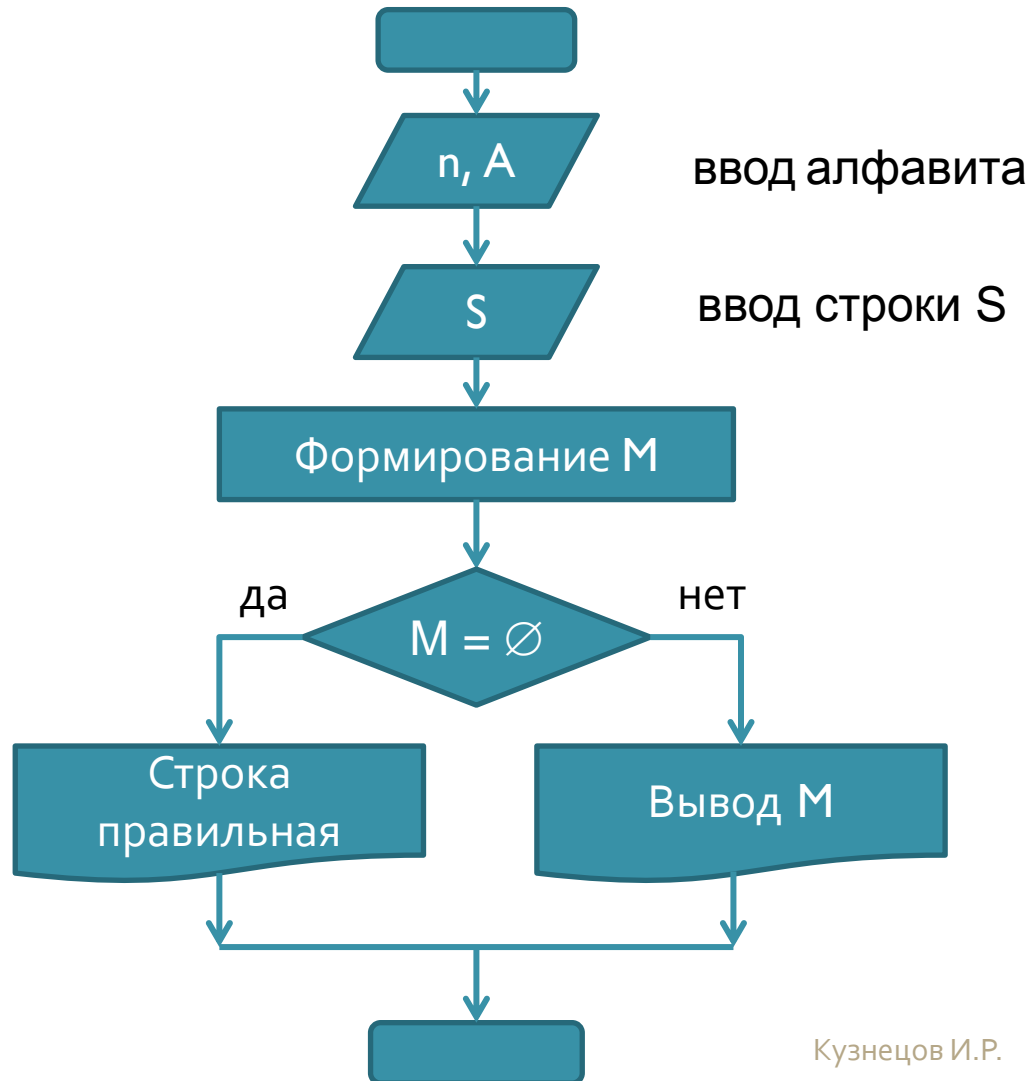
Дана строка символов

$$S = s_1 s_2 \dots s_k$$

Построим множество  $M = \{ i \mid s_i \notin A, i = 1, \dots, k \}$

Тогда, если  $M = \emptyset$  – строка  $S$  правильная, иначе – строка  $S$  неправильная

# Алгоритмизация



Кузнецов И.Р.

# Текст программы (Pascal)

- program analysis;  
const n1=255;  
type natural = 1 .. n1;  
var i, n, k : natural; A: set of char; a1 : char;  
s : string; M : set of natural;  
begin  
write('число элементов алфавита ='); readln(n);  
writeln('ввод символов алфавита');  
for i := 1 to n do begin  
write(символ = ); readln(a1);  
A := A + [a1]  
end;  
write('задайте строку s= '); readln(s);  
k := length(s); M := [];  
for i := 1 to k do if (not(s[i] in A)) then M := M + [i];  
if M=[] then writeln('строка правильная')  
else begin writeln ('номера недопустимых символов ');  
for i := 1 to k do if (i in M) then writeln (i)  
end;  
end.  
end.

**Цикл лекций подготовлен в 2013/2014уч. году  
Кузнецовым Игорем Ростиславовичем,  
доцентом кафедры радиоэлектронных средств  
Санкт-Петербургского  
Государственного электротехнического  
университета «ЛЭТИ»**

Прочитан в дисциплине  
«Информатика»