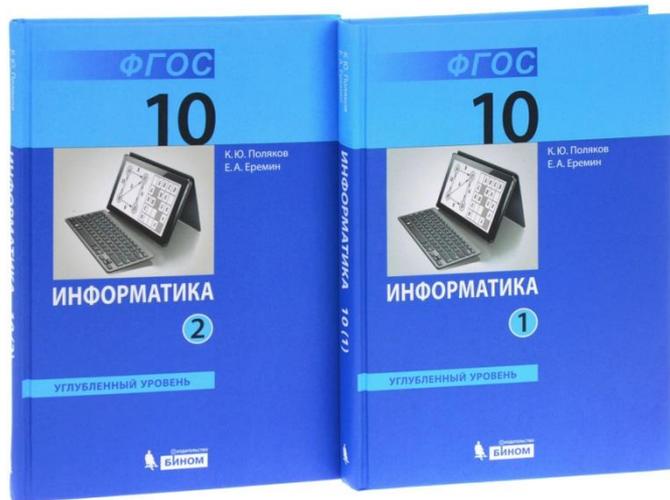


# Программирование на языке Паскаль **СИМВОЛЬНЫЕ СТРОКИ**



**Автор:**  
учитель информатики  
МБОУ г. Иркутска СОШ № 38  
Любина Ольга Николаевна

## Символьный тип

В FreePascal, Turbo Pascal и др. символьный тип *char* занимает **1 байт**.

В PascalABC.NET символьный тип *char* занимает **2 байта**.

# Строковый тип

---

*string* – строковый тип, состоит из набора последовательно расположенных символов *char* и используются для представления текста.

*В компиляторах FreePascal, Turbo Pascal строка имеет максимальную длину 255.*

*В PascalABC.NET строки могут иметь произвольную длину.*

К символам в строке можно обращаться, используя индекс: *s[i]* обозначает *i*-ый символ в строке, нумерация начинается с единицы.

# Строковый тип

---

*string* – тип данных, который используется для описания строк заданного конечного размера.

*var s: string;* { символьная строка }

Операция **+** для строк означает **конкатенацию** (слияние) строк, её результат имеет строковый тип.

К строкам применимы все операции сравнения **<**, **>**, **<=**, **>=**, **=**, **<>**.

# Функции `ord` и `chr`

---

***chr(n)*** - функция, возвращающая символ с кодом *n* в кодировке Windows.

## Пример программы

***ord(c)*** - функция, возвращающая значение типа *byte*, представляющее собой код символа *c* в кодировке Windows.

## Пример программы

# Символьные строки

---

Присваивание:

```
s := 'Вася пошёл гулять' ;
```

```
var s: string;
```

Ввод с клавиатуры:

```
readln(s) ;
```

Вывод на экран:

```
writeln(s) ;
```

Отдельный символ:

```
s[4] := 'a' ;
```

---

Длина строки:

```
var n: integer;  
...  
n := Length(s);
```

Пример программы

Массив символов

```
var s: array[1..50] of char;
```

Пример программы

# Операции со строками

---

Объединение (конкатенация) :

```
s1 := 'Привет' ;  
s2 := 'Вася' ;  
s  := s1 + ', ' + s2 + '!' ;
```

'Привет, Вася!'

## Пример программы

Срез:

```
s := '123456789' ;  
s1 := сору(s, 3, 5) ;      { '34567' }
```

откуда

с какого  
символа

СКОЛЬКО  
СИМВОЛОВ

5

## Пример программы

# Операции со строками

---

Удаление:

```
s := '123456789';  
Delete(s, 3, 6); { '129' }
```

с какого  
символа

сколько  
символов

Пример программы

Вставка:

```
s := '123456789';  
Insert('ABC', s, 3); { '12ABC3456789' }
```

что

куда

с какого  
символа

Пример программы

# Поиск в строках

---

```
s := 'Здесь был Вася.';
```

что где

```
n := Pos('с', s);
```

```
if n > 0 then
```

```
    write('Номер символа ', n)
```

```
else
```

```
    write('Символ не найден.');
```

[Пример программы](#)

# Пример обработки строк

---

**Задача:** Ввести имя, отчество и фамилию. Преобразовать их к формату «фамилия-инициалы».

**Пример:**

Введите имя, отчество и фамилию:

**Василий Алибабаевич Хрюндиков**

**Результат:**

**Хрюндиков В.А.**

Алибабаевич Хрюндиков

**Алгоритм:**

- найти первый пробел и выделить имя
- удалить имя с пробелом из основной строки
- найти первый пробел и выделить отчество
- удалить отчество с пробелом из основной строки
- «сцепить» фамилию, первые буквы имени и фамилии, точки, пробелы...

Хрюндиков

Хрюндиков В.А.

# Пример обработки строк

---

```
program FIO;
var s, name, name2: string;
n: integer;
begin
write('Введите имя, отчество и фамилию: ');
readln(s);
n:= Pos(' ', s); {находит кол-во символов включая 1-ый пробел}
name:= Copy(s, 1, n-1); {взять имя}
Delete(s, 1, n); {удаляет имя с пробелом из основной строки}
n:= Pos(' ', s); {в новой строке находит количество символов
включая первый пробел}
name2:= Copy(s, 1, n-1); {взять отчество}
Delete(s, 1, n); {удаляет отчество с пробелом из последней
строки, осталась фамилия}
s:= s + ' ' + name[1] + '.' + name2[1] + '.';
{берёт фамилию, пробел, первую букву от имени, ставит точку,
первую букву от отчества, ставит точку}
writeln(s) {выводит строку s}
```