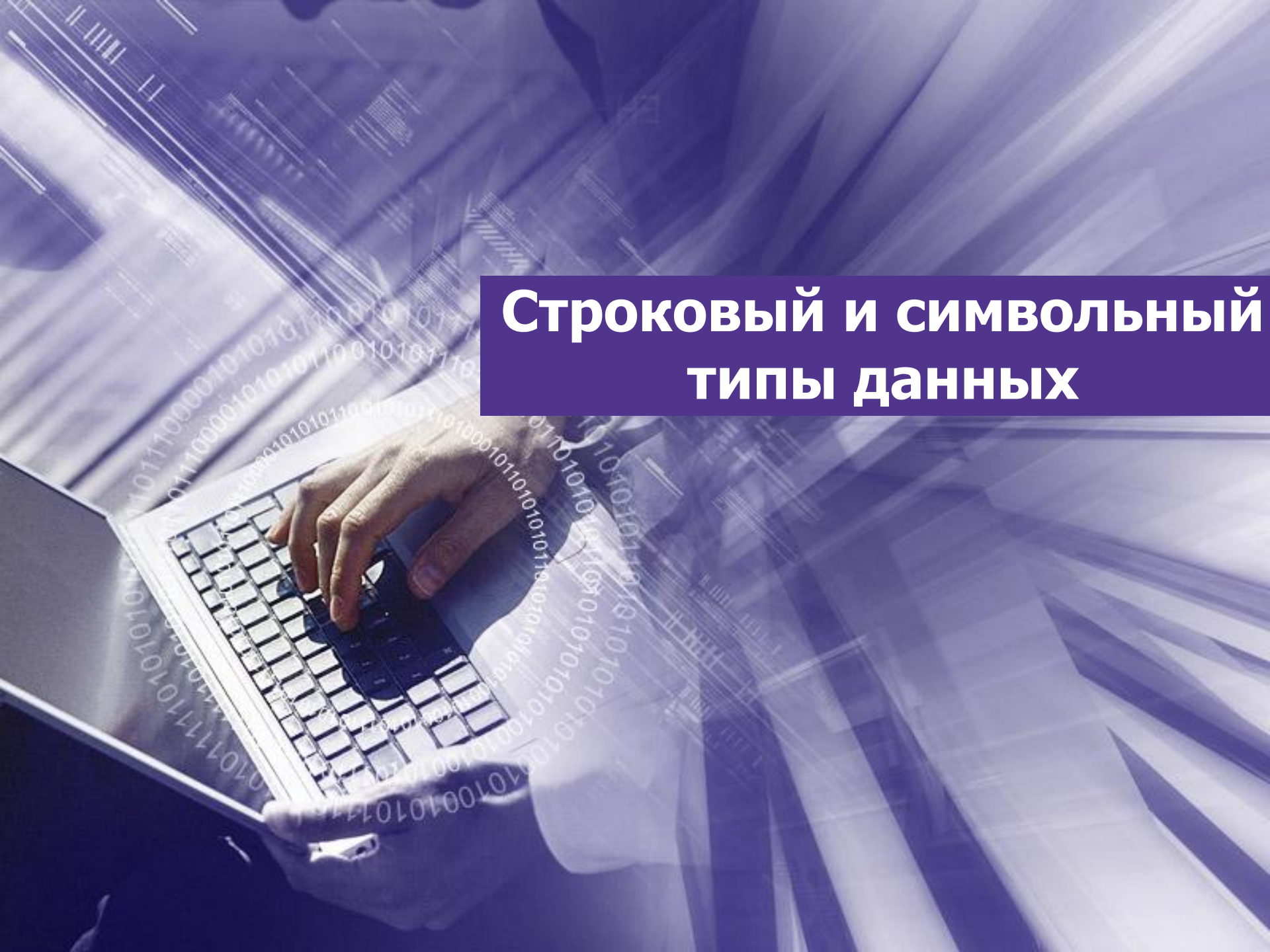
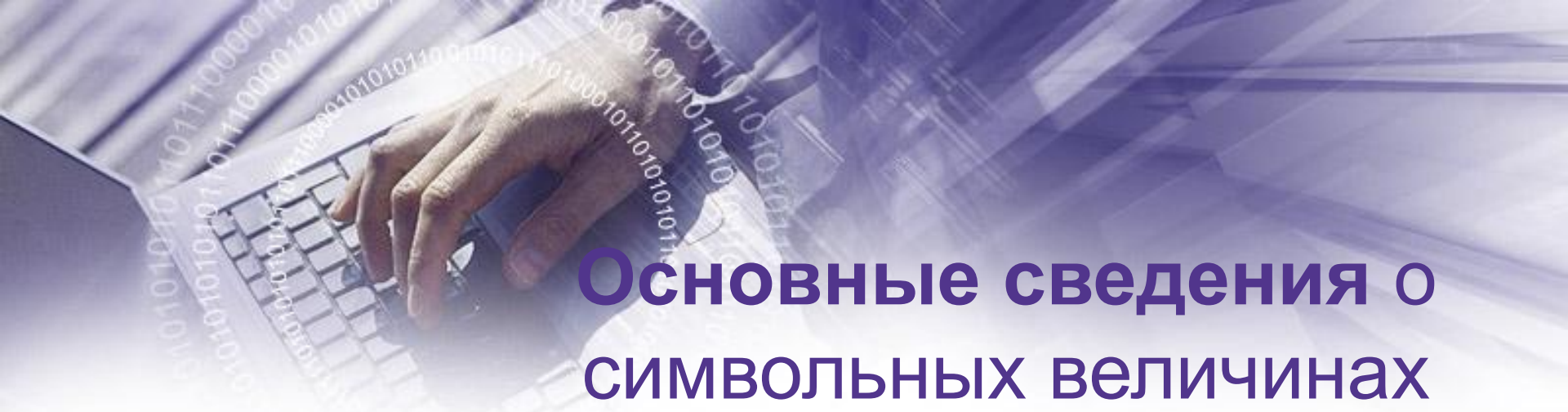
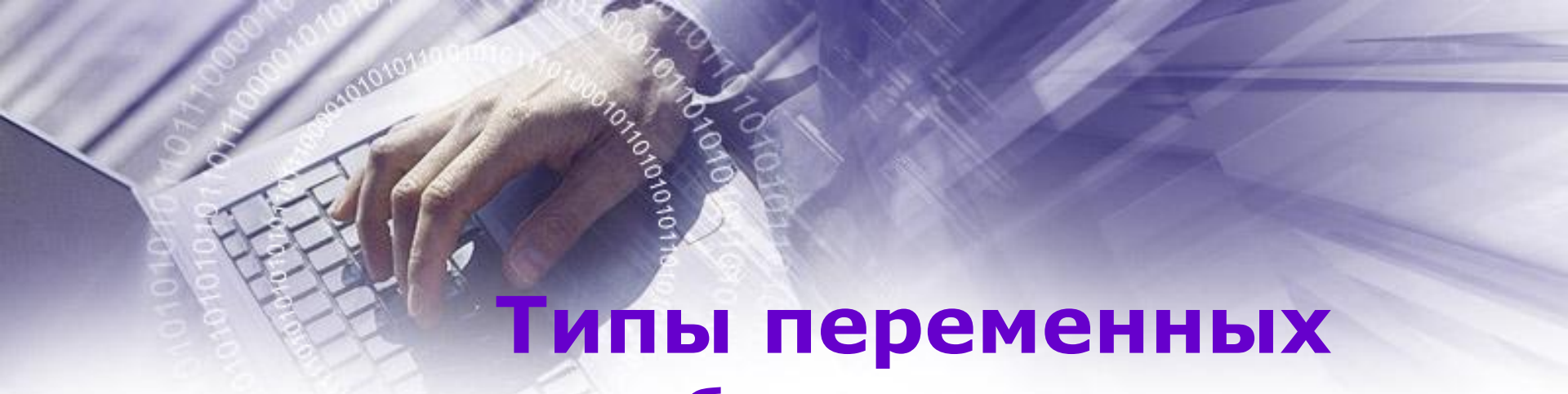


Строковый и символьный типы данных



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,…), (A,B…, a,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



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’.

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 (конкатенация)

- 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;
```

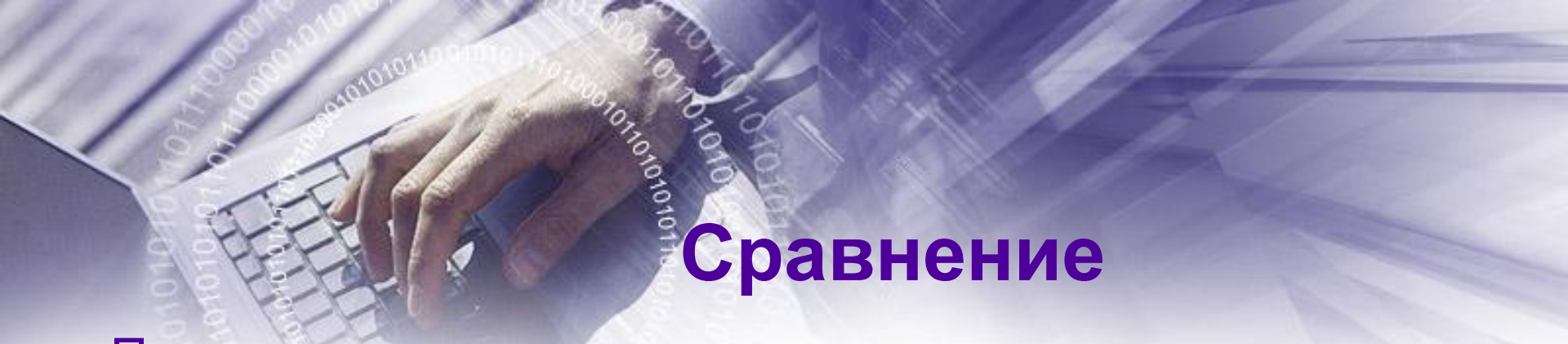
В результате значение строки str3: «У Егорки всегда отговорки», но…

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 всегдаотговорки»,

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

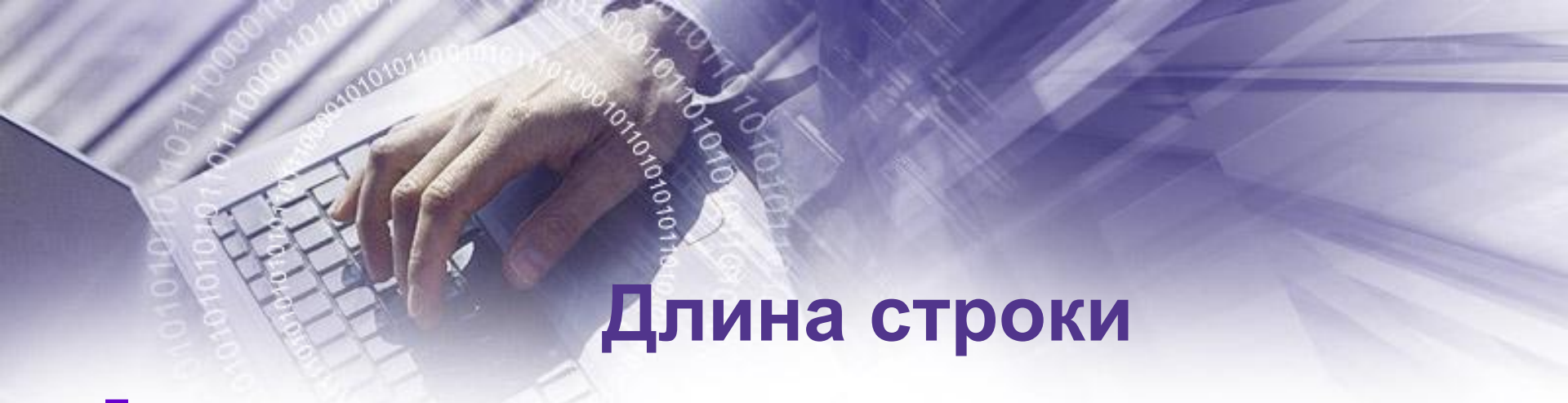


Сравнение

- Происходит посимвольно слева направо: сравниваются коды соответствующих символов до тех пор, пока не нарушится равенство, при этом сразу делается вывод о знаке неравенства.
- 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.

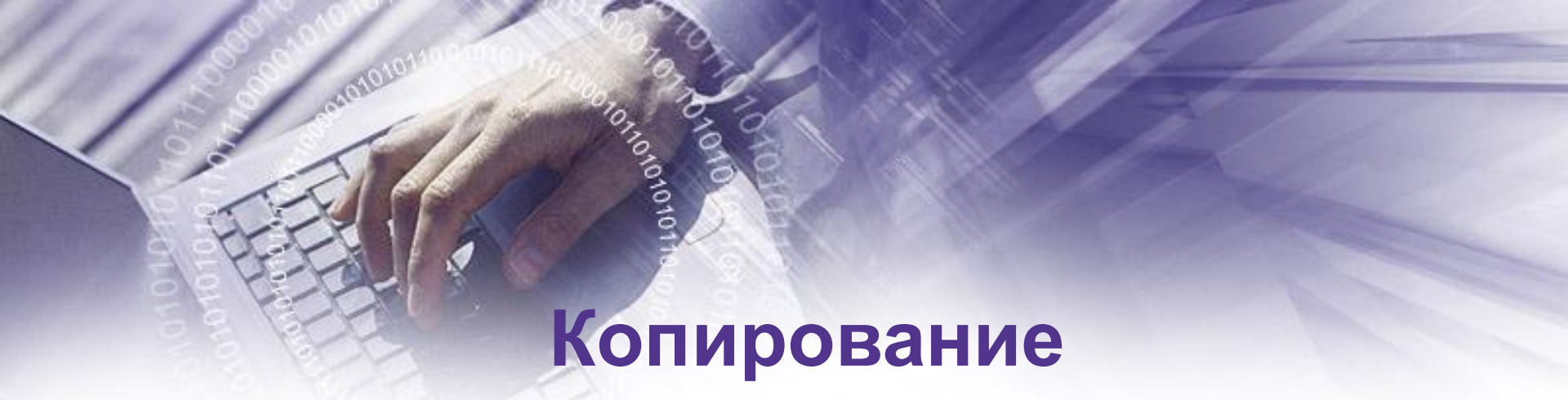


Длина строки

- 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);
```

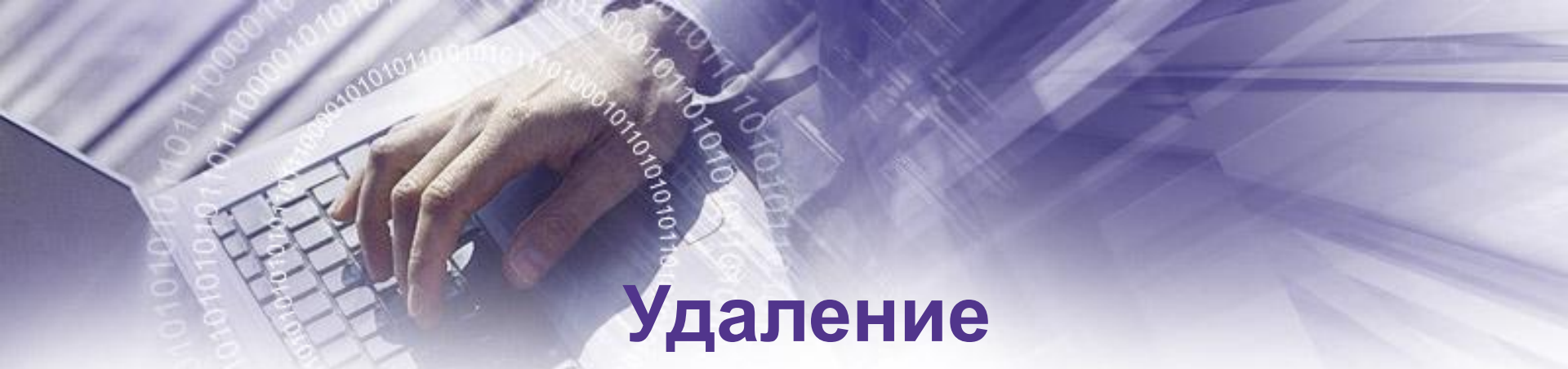
Результат: k1=8; K2=14.



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'
```



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)

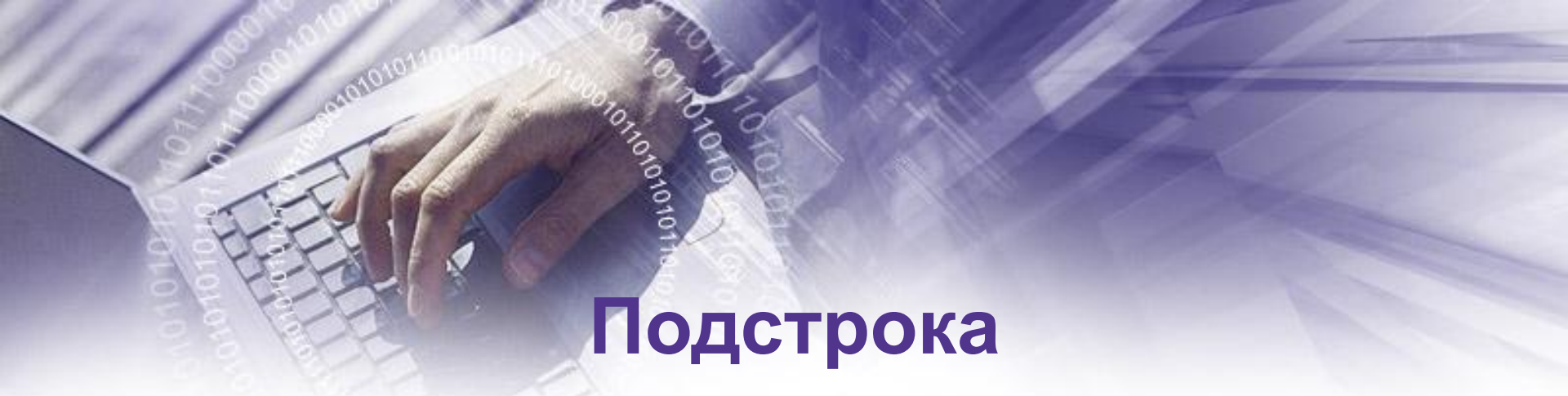
- Вставку строки ***Str1*** в строку ***Str2***, начиная с ***n***-го символа осуществляет процедура ***Insert(Str1,Str2,n)***, при этом первая строка не изменяется, а вторая получает новое значение.
- Пример:

Str1:=‘ABCDEFGH’;

Str2:=‘abcdefgh’;

Insert(str1,str2,3);

Результат: Str2=‘abABCDEFGHcdefgh’



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.