



Базовые функции Delphi

Любая программа, которая успешно прошла тестирование, устарела.

8.1. Функции и процедуры работы со строками

Особенностью пакета delphi является то, что вся информация на экране является символьной, поэтому при работе необходимо переводить данные в тот формат, который соответствует задаче. Для перевода используются функции преобразования типов:

Inttostr(n) – переводит целочисленный тип в строку, где n – целое число.

Floattostr(a) – переводит вещественный тип данных в строку, a – вещественное число.

Strtoint(s) – строку переводит в целое число, где s – строка символов.

Strtfloat(s) – строку переводит в вещественное число, где s – строка СИМВОЛОВ.

Floattostrf(a,f,r,m) – вещественное число переводит в строку с учетом формата представления данных.

Где a – вещественное число;

f – формат представления числа;

r – точность (общее количество цифр);

m – количество цифр после запятой.

Параметры r и m зависят от выбранного формата представления числа. Описание форматов представлены в таблице 8.1.

Константы, определяющие формат вывода данных.

Константа	Формат
Ffgeneral	Общий цифровой формат. Число изображается с десятичной точкой, если количество цифр слева от точки меньше или равно заданной точности (r) или если число больше или равно 0,00001. В противном случае число изображается в «научном формате» и количество цифр экспоненты.
Ffexponent	Научный формат. Число изображается в виде d.ddd...Esddd, где s – знак «плюс» или «минус»; d – десятичная цифра. Общее количество цифр числа, включая число перед десятичной точкой, равно заданной точности. Параметр m задает минимальное количество цифр представления экспоненты (0..4)
Fffixed	Формат с десятичной точкой. Число преобразуется к виду: ddd.ddd... Если число отрицательное, строка, являющаяся изображением числа, начинается знаком «минус». Перед десятичной точкой всегда находится как минимум одна цифра. Параметр m задает количество цифр после десятичной точки (0..18). Если количество цифр целой части больше, чем заданная точность, то число изображается в научном формате.
Ffnumber	Числовой формат. Числовой формат похож на формат с десятичной точкой (ffixed), только в изображении числа используется разделители групп разрядов. Например. Число 45278,34 будет изображено так: 45 278,34.
ffcurrency	Денежный формат. Используется в изображении числа в денежном формате разделители целой и дробной частей, разделители групп разрядов, изображение денежной единицы определяется настройкой Windows. Параметр m задает количество цифр после десятичной точки.

Функция **length(строка)** возвращает длину строки, указанной в скобках. Значение функции (целое число) – количество символов, из которых состоит строка, включая пробелы.

Процедура **delete(строка,p,n);**

Позволяет удалить часть строки. Где строка – переменная строкового типа, p – номер символа, с которого начинается удаляемая подстрока, n – длина удаляемой подстроки.

Например:

s:=’Город Санкт-Петербург’;

Delete(s,7,6);

В результате переменная s будет равна «Город Петербург».

Функция **pos(подстрока,строка)**, где подстрока – строковая константа или переменная, которую надо найти в строковой константе или переменной строка.

Например, в результате выполнения команды

P:=pos(’Пе’,’Санкт-Петербург’);

Значение переменной p будет равно 7. Если в строке нет искомой подстроки, то функция возвращает ноль.

Функция **copy(строка,p,n)**. Позволяет выделить фрагмент строки. Где строка – выражение строкового типа, содержащее строку, фрагмент которой надо получить, p – номер первого символа в строке строка, с которого начинается выделяемая подстрока, n – длина выделяемой подстроки.

Например,

st:=’Инженер Иванов’;

Fam:=copy(st,9,6);

Значение переменной s будет равно «Иванов».

Функция **Chr (n)** возвращает символ, соответствующий указанному ASCII-коду. N – целое число.

Функция **AnsiCompareStr(str1,str2)**. Сравнение не затрагивает длину строки, оно выполняется по буквам. Но более длинная строка больше чем короткая. Сравнение чувствительно к регистру. Результат будет отрицательным, если str1 меньше str2, равен нулю, если строки равны и больше нуля, если str1 больше str2.

Функция **CompareText(str1,str2)**. Аналогична предыдущей функции, но при сравнении не учитывает регистр.

Функция **Concat(str1, str 2, ...,str n)**. Объединяет несколько строк в одну. Если длина результирующей строки превышает 255 символов, то лишние символы отсекаются.

Процедура **Insert(sub, str, n)**. Вставляет подстроку *sub* в строку *str* с позиции *n*. Параметры имеют следующий тип: *sub*, *str* – string, *n* – integer.

Функция **LowerCase(str)**. Преобразовывает все ASCII символы строки в строчные (нижний регистр).

Процедура **Str(a, st)**. Преобразовывает числовое значение в его строковое представление. Параметры имеют следующий тип: *a* – integer или real, *st* – string.

Функция **StringOfChar(s, n)**. Возвращает строку, состоящую из заданного количества *n* одинаковых символов *s*. Параметры имеют следующий тип: *s* – char, *n* - integer.

Функция **Trim(str)**. Удаляет пробелы и управляющие символы из начала и конца строки.

Функция **TrimLeft(str)**. Удаляет пробелы и управляющие символы, стоящие в начале строки.

Функция **TrimRight(str)**. Удаляет пробелы и управляющие символы, стоящие в конце строки.

Функция **UpCase(s)**. Возвращает переданный символ в верхнем регистре. Параметр *s* имеет тип char.

Функция **UpperCase(str)**. Конвертирует все символы строки, лежащие в диапазоне от 'a' до 'z', в прописные (верхний регистр).

8.2. Математические функции и процедуры

Функция **Abs(x)**. Возвращает абсолютное значение аргумента. Параметр функции может иметь любой числовой тип.

Функция **Exp(x)**. Вычисляет значение экспоненту числа *x*. Параметр функции может иметь любой числовой тип.

Функция **Frac(x)**. Возвращает дробную часть аргумента. Аргумент и результат выполнения функции имеют действительный тип.

Например: $Frac(567.89)=0.89$.

Функция **Int(x)**. Возвращает целую часть аргумента. Аргумент и результат выполнения функции имеют действительный тип.

Например: *Int(567.89)=567.00*.

Функция **Max (a,b)**. Возвращает большее из двух чисел. Параметры функции могут иметь любой числовой тип.

Функция **Mean(имя_массива)**. Возвращает среднее число из набора значений массива. Массив имеет действительный тип.

Например:

Var

mas: array[1..5] of real;

sr : real;

...

sr := Mean(mas);

Функция **Min (a,b)**. Возвращает меньшее из двух чисел. Параметры функции могут иметь любой числовой тип.

Математическая константа **Pi**. Возвращает значение числа Пи.

Функция **Round(x)**. Округляет число к ближайшему целому. Аргумент функции имеет действительный тип, результат целочисленный.

Функция **Sqr(x)**. Вычисляет квадрат аргумента x. Аргумент может быть любого числового типа.

Функция **Sqrt(x)**. Вычисляет квадратный корень аргумента x. Аргумент может быть любого числового типа.

Функция **Trunc(x)**. Отсекает дробную часть числа. Аргумент функции имеет действительный тип, а результат выполнения функции - целочисленный.

Например: *trunc(567.89)=567*.

8.3. Функции работы с переменными типа дата

Функция **Date()**. Возвращает текущую дату.

Функция **Now()**. Возвращает текущую дату и время.

Функция **Time()**. Возвращает текущее время.

Функция **Tomorrow()**. Возвращает дату завтрашнего дня.

Функция **Yesterday()**. Возвращает дату вчерашнего дня.

Функция **Datetostr(dat)**. Преобразует дату, указанную в качестве параметра в строку.

Функция **Datetimetostr(dat_tim)**. Преобразует дату и время, указанные в качестве параметра в строку.

Функция **Strtodate(st)**. Преобразует строку, указанную в качестве параметра в дату.

Функция **Strtotime(st)**. Преобразует строку, указанную в качестве параметра во время.

Функция **Strtodatetime(st)**. Преобразует строку, указанную в качестве параметра в дату и время.

Функция **DaysBetween (dat2, dat1)**. Возвращает целое число, равное результату вычитания dat1 из dat2, т.е. возвращая разность в целых днях.

Например:

```
var
```

```
dat1, dat2 : TDateTime;
```

```
n: integer;
```

```
...
```

```
n:=DaysBetween(dat2, dat1);
```



Контрольные вопросы

1. Процедуры и функции для работы со строковыми переменными.
2. Процедуры и функции для работы с переменными типа дата.
3. Математические процедуры и функции.