IVXL
CDM

Числовая система определяет набор значений, которые используются для представления чисел. Компьютер использует двоичную систему исчисления. Двоичная и шестнадцатеричная система исчисления (используется для представления значений [*номера и адреса памяти*
] в компьютерных системах) являются важными для программирования. Двоичные числа важны, потому что компьютер работает с двоичными числами - номера состоят из двух цифр 1 и 0. Шестнадцатеричные числа удобны, потому что они позволяют легко обрабатывать двоичные числа.

Delphi предоставляет множество полезных функций для преобразования целых чисел (в десятичной системе) строковые значения (строка это способ представления шестнадцатеричных и двоичных чисел), и наоборот.

Давайте посмотрим, какие функции можно использовать для преобразования чисел из одной системы в другую, каких функций не хватает и как легко реализовать их в **Object Pascal. IntToHex**
,
ObjectPascal.HexToInt
.

В модуле **SysUtils** есть функция **IntToHex**, которая возвращает шестнадцатиричное представление числа.

```
function IntToHex (Value: Integer; Digits: Integer): String;
```

Хорошо, у нас есть возможность преобразовать целое число в шестнадцатеричное, а где функция **HexToInt**? Функция **HexToInt** не существует в Delphi?.

Тем не менее, Delphi позволяет использовать шестнадцатиричные выражения (*используя префикс*
$
). Вот простая функция
HexToInt

:

function HexToInt(HexNum: string): LongInt; begin Result:=StrToInt('$' + HexNum) ; end; **Функции IntToBin, BinToInt**

Если Вы не знакомы с двоичными числами, то можете почитать *Введение в двоичную арифметику* .

Таким образом, как организовать функции **IntToBin** и **BinToInt** в Delphi. Вот один из примеров:

function IntToBin (Value: LongInt; Digits: integer): string; begin result := StringOfChar ('0', Digits) ; while Value > 0 do begin if (Value and 1) = 1 then result [Digits] := '1'; dec (Digits) ; Value := Value shr 1; end; end; function BinToInt(Value: String): LongInt; var i: Integer; begin Result:=0; // Удаляем ведущие нули while Copy(Value, 1, 1) = '0' do Value:=Copy(Value, 2, Length(Value) - 1) ; // преобразовываем for i:= Length(Value) downto 1 do if Copy(Value, i, 1) = '1' then Result:=Result + (1 shl (Length(Value) - i)) ; end; **Римские**

В большинстве случаев сегодня используется арабская система исчисления. Десять однозначных цифр от **0** до **9**, составляют символы нашей системы исчисления. Примером другой системе счисления является система римских цифр, которые могут представлять все числа от **1** до **1000000**, используя только семь символов: *I = 1, V = 5, X = 10, L = 50, C = 100, D = 500, M = 1000* . Пример: 1999 (арабская) является MCMXCIX (римская).

А вот функция для перевода целых арабских чисел в римские:

function IntToRoman(Value: LongInt): String; const Arabics: Array[1..13] of Integer = 1,4, 5,9,10,40,50,90,100,400,500,900,1000) ; Romans: Array[1..13] of String = ('I','IV', 'V','IX','X','XL','L','XC','C','CD','D','CM','M') ; var j: Integer; begin for j := 13 downto 1 do while (Value >= Arabics[j]) do begin Value := Value - Arabics[j]; Result := Result + Romans[j]; end; end;

По материалам About.com