

Лекція 1

Основи мови програмування Object Pascal

Типи даних:

Integer- ціле число; **Byte** - число в діапазоні 0..255

Real, double, single, extended - будь-яке реальне число (з плаваючою точкою);

Boolean - Логічний тип приймає значення істина або брехня (true, false);

Char - символьний тип ('A', 'F');

String, ANSIString - строковий тип даних ('Hello World');

Змінні

Оголошує наступним чином:

var ім'я змінної: тип;

наприклад:

var c: integer;

Масиви

Ініціалізація відбувається наступним чином:

Type ім'я масиву = **array** [кількість елементів, кількість елементів ...] **of** тип;

Type arr = **array** [1..10] **of** real;

.....

var a: arr;

Звертатися до елементів масиву потрібно наступним чином:

a [1]: = 10;

a [2]: = 12;

...

константи

const PI = 3.14; E: Real = 2.87;

Записи

type ім'я запису = **record**

змінні різних типів;

end;

наприклад:

```
// Налаштування
type optionsrec = record
  n: string;
  dt: string;
  niter: string;
end;
.....
var opt: optionsrec;

opt.n: = 'MyString';
opt.dt: = 'Hello !!!';
```

Безлічі

Зарезервоване слово set (безліч) визначає безліч не більше ніж з 256 порядкових значень:

Set of тип (0..255)

Розглянемо наступний приклад

```
type f = set of byte;
.....
var h: f; // змінна безлічі
.....
    i: = StrToInt (Edit1.Text);

    h: = [1..100];

if i in h then

    ShowMessage ( 'Входить!!!')

else

    ShowMessage ( 'Не входить!!!' + s);
```

ОПЕРАЦІЇ І ВИРАЗИ

За характером виконуваних дій операції поділяються на такі групи:

- 1. Арифметичні операції: +, -, *, /, div, mod**
- 2. Операції відносин: =, <>, <,>, <=,> =**
- 3. Булеві (логічні) операції: not, and, or, xor**
- 4. Строкова операція (конкатенація): + 'Hello' + 'World' => 'Hello World'**
- 5. Операції над множинами: +, -, *, in, <=,> =**

Основні операції порівняння даних

= - Так само
<> - Чи не само
> - Більше
< - Менше
> = - Більше або дорівнює
< = - Менше або само

математичні вирази

+ - Додавання
- - віднімання
***** - Множення
/ - Ділення
div - Ділення націло
mod - Залишок від ділення лівого операнда на правий

Логічні операції

And - Логічне І. Результат істина, якщо обидва операнда істинні, інакше помилковий
Or - Логічне АБО. Результат істина, якщо хоча б один з операторів істина, в іншому випадку брехня.
Xor - Що виключає Або. Результат істина якщо операнди не рівні один одному. Інакше брехня.
Not - Логічне заперечення.

Оператор присвоєння в Паскалі : =

Наприклад: a: = 10;

Функції перетворення типів

Конвертація цілого числа в рядок (string) і навпаки

IntToStr (k) k- integer;

StrToInt (s) s - string;

Конвертація дійсного числа в рядок (string) і навпаки

FloatToStr (k) k- integer;

StrToFloat (s) s - string;

Конвертація системного часу і дати в рядок (TDateTime)

TimeToStr (t) t - TDateTime;

DateToStr (d) d - TDateTime;

Конвертація рядки string в Java String.

StringToJString (s) s - string;

Конвертація за форматом

FloatToStrF (n, f, k, m) - рядок, що є відображенням речового n. При виконанні функції вказують: f - формат (спосіб відображення); k - точність (потрібна загальна кількість цифр); m - кількість цифр після десяткової точки.

Round (n) - округлення до найближчого цілого числа. (n - дійсне число)

Використання власних підпрограм

Функції (повертає якесь значення)

```
function ім'я функції (вхідні параметри): тип;  
begin  
    тіло функції  
end;
```

приклад:

```
function f (x, y: real): real;  
begin  
    f: = y * x;  
end;
```

Її виклик в основній програмі:

```
...  
x: = 10;  
y: = 20;  
z: = f (x, y);
```

Процедури (не повертає значень)

```
procedure ім'я процедури (вхідні параметри);  
begin  
    Тіло процедури;  
end;
```

приклад

```
procedure Input (var x, y: real; s1, s2: string);  
begin  
    x: = StrToFloat (s1);  
    y: = StrToFloat (s2);  
end;
```

```
...  
Input (x, y, Edit1.Text, Edit2.Text);
```

УМОВНИЙ ОПЕРАТОР

If умова then
 оператор
Else
 оператор;

приклад

If $a > 0$ then

 $x := a$
Else
 $x := 0;$

У разі, коли необхідно виконати декілька операторів використовуються операторні дужки begin end.

If $a < 0$ then
begin
 $x := a;$
 $z := 0;$
end
Else
begin
 $x := 0;$
 $z := a;$
end;

Приклад використання логічних операцій $a \in [-10, 10]$

if $(a > -10)$ and $(a < 10)$ then
 $y := a * a$
else
 $y := a;$

ЦИКЛИ

Цикл з передумовою:

while (умова) do Оператор;

while (умова) do
begin
 Оператори;

end;

приклад:

```
x: = 0;

while (100> x) do x: = x + 1;
.....
x: = 0;

while (100> x) do
begin
    x: = x + 1;
    Edit1.Text: = IntToStr (x);
end;
```

Цикл з лічильником

```
for початкове значення to кінцеве значення do оператор;

for початкове значення to кінцеве значення do
begin
    Оператори;
end;
```

приклад:

...

```
For i: = 1 to 100 do
Begin
    Memo1.Lines.Add (FloatToStr (a [i]));
End;
```

...

Генератор псевдовипадкових чисел

Служить для генерації числа в певному діапазоні. Для генерації використовується функція Random (n) генерує число 0..n-1. Для ініціалізації генератора випадкових чисел потрібно викликати процедуру Randomize;

приклад: **Randomize;** // ініціалізація
 n: = random (10); // n приймає значення 0..9

Обробка виняткових ситуацій

Структура обробника виняткової ситуації:

on Деякий-Виняток do Обробка_винятку;

Конструкцію on-do можна використовувати тільки в рамках обробника винятків:

```
try
    оператор (оператори); except on Виняток do Обробка_Винятку; on Другий_Виняток
do Його_Обробка;
```

Приклад:

```
...
try
    n: = StrToInt (Edit1.Text);

except on EConvertError do
    begin
        ShowMessage ( "Невірне введення! ");
        exit;
    end;
end;
...
```

Оператор вибору case

Служить для вибору дій в залежності від значення цілочисельний змінної.

Структура наступна:

Case <значення> of

1:

Begin

 <Оператор1>;

 <Оператор2>;

End;

2: <оператор1>;

.....

Else <Оператор>;

End;

Приклад використання:

```
case ColorSelected of // ColorSelected - Integer
```

```
1: RadioButton1.IsChecked: = true;
```

```
2: RadioButton2.IsChecked: = true;
```

```
3: RadioButton3.IsChecked: = true;
```

```
4: RadioButton4.IsChecked: = true;
```

```
5: RadioButton5.IsChecked: = true;
```

```
end;
```