

Тема: Операції.

Мета: Познакомитися із правилами побудови, обрахунку, використання різноманітних виразів – арифметичних, логічних, рядкових.

1. Поняття операції та виразу. Арифметичні та рядкові операції.

Основне призначення введених у комп'ютер даних – їх опрацювання із наступним виведенням і / або збереженням. Дії над даними задаються виразами. Вирази визначають порядок виконання дій над даними і складаються із операндів, круглих дужок і знаків операцій. Операнди являють собою константи, змінні і виклики функцій. Операція – це дія, яка виконується над операндами.

Операції в Delphi поділяються на арифметичні, операції відношення, логічні (булеві), рядкові, операції отримання адреси тощо. Вирази відповідно називаються арифметичними, відношення, логічними (булевими), рядковими тощо в залежності від того, якого типу операнди та операції в них використовуються.

У складних виразах послідовність обрахунків визначається дужками, а якщо підряд йде декілька операцій, не розділених дужками, то послідовність їх обчислення визначається старшинством та асоціативністю операцій. За старшинством (пріоритетом) операції поділяються на чотири рівні.

Пріоритет операцій на мові Delphi:

Операція	Пріоритет
@, not	Перший (вищий)
*, /, div, mod, and, shl, shr, as	Другий
+, -, or, xor	Третій
=, <>, >, <, >=, <=, in, is	Четвертий (нижчий)

Операції із більш високим пріоритетом виконуються раніше, ніж операції із більш низьким пріоритетом. Якщо ж підряд йде декілька операцій із рівним пріоритетом, то послідовність виконання обрахунків визначається асоціативністю, яка у всіх операцій однакова – зліва направо.

Арифметичні операції.

Найчастіше у програмах зустрічаються арифметичні операції; вони виконують арифметичні дії над значеннями операндів цілочисельного чи дійсного типів даних.

Операція	Дія	Тип операндів	Тип результатів
+	Додавання	Цілий, дійсний	Цілий, дійсний
-	Віднімання	Цілий, дійсний	Цілий, дійсний
*	Множення	Цілий, дійсний	Цілий, дійсний
/	Ділення	Цілий, дійсний	Дійсний
div	Цілочисельне ділення	Цілий	Цілий
mod	Залишок від ділення	цілий	Цілий

Операції додавання, віднімання та множення відповідають аналогічним операціям у математиці. На відміну від них, операція ділення має три види: звичайне ділення (/), цілочисельне ділення (div), залишок від ділення (mod).

Результат, отриманий при використанні різних операцій ділення:

Вираз	Результат
-12 / 4	-3.0

Вираз	Результат
25 div 7	3
25 mod 7	4
-10 div 4	-2
12 / 5	2.4
6 div 8	0
6 mod 8	6
20 mod 5	0

Операції з рядками.

Рядковий тип даних є також одним із найбільш розповсюджених у візуальному програмуванні. Рядок – це послідовність символів. У Delphi існує декілька рядкових типів. Основні з них:

Тип рядка	Максимальна довжина	Розмір пам'яті	Сфера застосування
ShortString	255 символів	Від 2 до 256 байт	Погана сумісність, зберігання невеликих рядків, швидка обробка
AnsiString	Близько 2^{31} символів	Від 4 байт до 2 Гб	8-бітні символи (ANSI), DBCS, ANSI, MBCS ANSI ...
WideString	Близько 2^{30} символів	Від 4 байт до 2 Гб	Юнікод-символи – багатокористувацькі сервери, мультимовні додатки ...

Для більшості потреб підходить тип AnsiString.

Стандартні функції опрацювання рядків.

1. Функція Length (Str: String) – повертає кількість символів (довжину рядка).

```

Var
Str: ansistring;
L: integer;
Begin
Str:='Добрий день!';
L:=length(Str); {результат L=15}
End;
```

2. Процедура SetLength (Str: String; NewLength: Integer) – може зменшувати довжину рядка. Якщо рядок містить більшу кількість символів, ніж задано у другому параметрі процедури, то «зайві» символи ігноруються.

```

Var
Str: ansistring;
Begin
Str:='Добрий день!';
Setlength (Str, 6); {Результат Str='Добрий'}
End;
```

3. Функція Pos (SubStr, Str: String) – повертає позицію підрядка в рядку. Нумерація символів розпочинається із одиниці. При відсутності підрядка в рядку функція повертає значення 0.

```

Var
Str1, Str2: ansistring;
P: integer;
```

```
Begin  
Str1:='Добрий день!';  
Str2:='об';  
P:=pos(Str2,Str1); {результат P=2}  
End;
```

4. Функція Copy (Str: string; Start, Length: integer) – повертає підрядок (частину рядка) розпочинаючи із символу Start довжиною Length. Обмежень на Length немає, проте якщо числове значення перевищує кількість символів, починаючи із позиції Start, то рядок копіюється до самого кінця.

```
Var  
Str1, Str2: ansistring;  
Begin  
Str1:='Добрий день!';  
Str2:=Copy(Str1,3,4); {Результат Str2='брий'}  
End;
```

5. Процедура Insert (SubStr: string; Str: string; Pos: integer) – вставляє в рядок Str підрядок SubStr в позицію Pos; починаючи із позиції вставки літери вхідного рядка розсуваються.

```
Var  
Str: ansistring;  
Begin  
Str:='Добрий день!';  
Insert ('новий', Str, 8); {Результат Str='Добрий новий день!'}  
End;
```

6. Процедура Delete (Str: string; Start, Length: integer) – вилучає із рядка Str символи, розпочинаючи із позиції Start довжиною Length (якщо вказаний параметр виходить за межі рядка – вилучає символи до кінця рядка).

```
Var  
Str: ansistring;  
Begin  
Str:='Добрий день!';  
Delete (Str, 7,5); {Результат Str='Добрий!'}  
End;
```

7. Функції AnsiUpperCase (Str: string) та AnsiLowerCase (Str: string) – перетворює рядок відповідно у верхній та нижній регістри.

```
Var  
Str1, Str2, Str3: ansistring;  
Begin  
Str1:='Задача';  
Str2:=ansilowercase (str1); {str2='задача'}  
Str3:=ansiuppercase (str1); {str3='ЗАДАЧА'}  
End;
```

Рядки можна порівнювати між собою за допомогою звичайних операторів порівняння:

```
Var  
Str1, Str2, Str3: ansistring;  
Logic1, logic2: Boolean;
```

Begin

Str1:='cat';

Str2:='cat';

Str3:='dog';

Logic1:=(str1=str2); {результат logic1=true}

Logic2:=(str1=str3); {результат logic1=false}

End;

Якщо рядки однакові, логічний вираз стане рівним true.

2. Теоретичні питання для самоконтролю.

1. Дайте означення поняття «вираз».
2. Що собою являють операції та операнди?
3. Які типи операцій використовуються у Delphi?
4. Перерахуйте операції за пріоритетом у мові Delphi.
5. Наведіть приклади математичних операцій у Delphi.
6. Назвіть основні рядкові типи Delphi.
7. Перерахуйте стандартні функції опрацювання рядків.