

Основи програмування (повторення)

Об'єктно-орієнтоване
програмування. Лекція №0
Викладач: Заспа Г.О.

[Змінні]

Змінна є одним з центральних понять в програмуванні. Змінна – це область пам'яті, призначена для збереження інформації. Змінна характеризується: 1) ім'ям; 2) значенням; 3) типом.

а 12

Значення – це інформація, яка знаходиться в змінній. В змінній завжди є якесь значення.

Ім'я – це ідентифікатор, який однозначно ідентифікує якусь змінну. Його задача – дати можливість розрізняти змінні при зверненні до них.

[Тип. Опис змінних]

Тип – це обмеження на можливе значення змінної. Обмеження встановлюються за видом значення та можливим діапазоном в рамках виду. Види значень: цілі, дійсні, символльні, логічні (булеві). Типи бувають прості та складні. Прості типи: integer (а також short, long, byte), real (а також float, double), char, boolean. Складні типи: масиви, рядки, записи, множини тощо.

В мовах з сильною типізацією (Паскаль, С, С++, Java) всі змінні повинні бути описані (оголошені) перед їх використанням. Опис змінної складається як мінімум з ім'я змінної та її типу. Приклад з Паскаля:

```
var  
  i:integer;
```

Приклад з мови С:

```
int i;
```

[Оператори]

Оператор – це дія, яка виконується програмою.

Оператори є активними складовими програми – вони щось роблять, на відміну від змінних, які є пасивними складовими – з ними щось роблять.

Оператори:

- Присвоєння значення
- Умови
- Циклів (з передумовою, з постумовою, з параметром)
- Варіанту (case, switch)
- Goto
- Повернення значення (return)
- Виклику підпрограми
- Інші

Оператор присвоєння значення

Найбільш вживаним оператором є **оператор присвоєння значення**. Його призначення – записати значення в змінну. В мові Паскаль він позначається :=, в мові С та похідних від неї =. Загальна форма запису оператору присвоєння значення:

змінна:=вираз;

Вираз – це сукупність констант, змінних та викликів функцій, об'єднаних знаками операцій та дужками. Константа – це просте значення, незмінна величина. Приклади констант: 5, 3.7. Приклади знаків операцій: +, -, *, /. Вираз завжди обчислюється і в результаті обчислення виходить одне значення. Типи змінної в лівій частині оператора присвоєння і значення, обчисленого з виразу в правій частині оператора присвоєння (тип виразу), повинні бути однаковими (а точніше, сумісними).

Приклади виразів:

1	X+2	1+x+cos(x)	X>2
---	-----	------------	-----

Питання: що можна робити зі змінними

Питання: якого типу може бути значення, обчислене з виразу (або, іншими словами, якого типу може бути вираз)

Оператор умови

Оператор умови є одним з операторів розгалуження, поряд з операторами варіанту та циклів.

Паскаль	С
if умова then <i>оператор</i> ; або if умова then <i>оператор1</i> else <i>оператор2</i> ;	if (умова) <i>оператор</i> ; або if (умова) <i>оператор1</i> ; else <i>оператор2</i> ;

На місці умови, якщо говорити точніше, повинен бути логічний вираз.

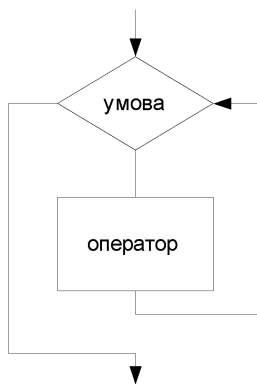
Оператори циклів

Операторів циклів є 3: цикл з передумовою, цикл з постумовою та цикл з параметром. Принциповою особливістю оператору циклу з постумовою є те, що в ньому тіло циклу обов'язково виконується хоча б один раз, а в операторі циклу з передумовою та операторі циклу з параметром може не виконатись ні разу.

Цикл з передумовою:

`while (умова)`

`оператор (тіло циклу);`

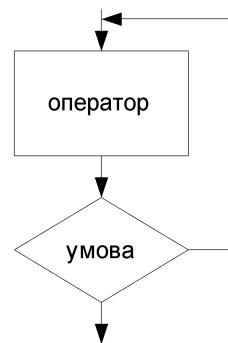


Цикл з постумовою:

`do`

`оператор (тіло циклу);`

`while (умова);`

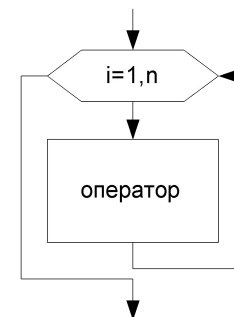


Цикл з параметром:

`for (int i=початкове значення;`

`умова продовж. циклу; i++)`

`оператор (тіло циклу);`



[Підпрограми]

Підпрограма – це окремо оформлена частина програми.

Говорячи про підпрограми, потрібно розглядати два питання:
опис підпрограм та виклик підпрограм.

Опис підпрограми задає, що вона може робити. Опис складається з заголовка підпрограми та тіла підпрограми. В заголовку вказуються ім'я підпрограми і формальні параметри, якщо вони є. Тіло підпрограми задає те, що вона повинна робити і складається з операторів. Приклад підпрограми з мови Паскаль:

```
procedure p1(i:integer); {заголовок}  
begin                    {тіло}  
  i:=i+10;  
  writeln('i=',i);  
end;
```


[Підпрограми]

Після того, як зроблено опис підпрограми, можна робити її виклик. Виклик підпрограми – це команда виконати тіло підпрограми. Він складається з ім'я підпрограми та фактичних параметрів, якщо для цієї підпрограми були задані формальні параметри (кількість та типи фактичних та формальних параметрів повинні співпадати). Приклад:

p1(9);

Формальний параметр – це місце в описі підпрограми, на яке буде підставлено фактичний параметр при виклику підпрограми. Формальний параметр має характер змінної і описується в заголовку підпрограми як змінна. Фактичний параметр – це завжди вираз.

Процедури та функції

Підпрограми діляться на два види – процедури та функції. Різниця між ними одна – функція повертає значення, а процедура – ні. Повернення значення – це властивість функції, яка дає можливість викликати її в рамках виразу, підставляючи на місце виклику функції повернене значення при обчисленні виразу. При описі функції обов'язково вказується тип значення, що повертається. Приклад:

```
program function_example;
var k:integer;
function square(x:integer):integer;
begin
  square:=x*x;
end;
begin
  k:=1+square(4);
  writeln('k=',k);
end.
```

Локальні та глобальні змінні

В програмуванні існує принцип вкладеності блоків коду, тобто один блок коду може бути вкладений в інший. Найяскравіше це проявляється при створенні підпрограм, які є "вкладеними" у програму. В свою чергу можуть існувати підпрограми всередині підпрограм (підпрограми, вкладені у підпрограми). Змінні, описані всередині вкладених блоків, називаються локальними по відношенню до тих блоків, в які вони вкладені. Змінні називаються глобальними по відношенню до всіх блоків, які вкладені в той, в якому ці змінні описані. Локальні змінні є доступними тільки в тому блоці, в якому вони описані. Глобальні змінні є доступними у всіх блоках, вкладених в той, в якому вони описані.