

Початки алгоритмізації та процедурного програмування

Циклічні програми

Актуалізація опорних знань

Завдання 1

Які з перерахованих нижче розділів є обов'язковими в програмі, написаній мовою Паскаль?

1) Var; 2) Const; 3) Type; 4) Label; 5) Begin ... End.

Завдання 2

При допомозі якої з наведених команд змінній B надається значення виразу (може бути декілька вірних варіантів)?

$$\left(\frac{x+y}{x-y} - \frac{x-y}{x+y} \right) \cdot \left(\frac{x-y}{x+y} + \frac{x+y}{x-y} \right)$$

а) $A := (x + y) / (x - y); B := (A - 1 / A) * (1 / A + A);$

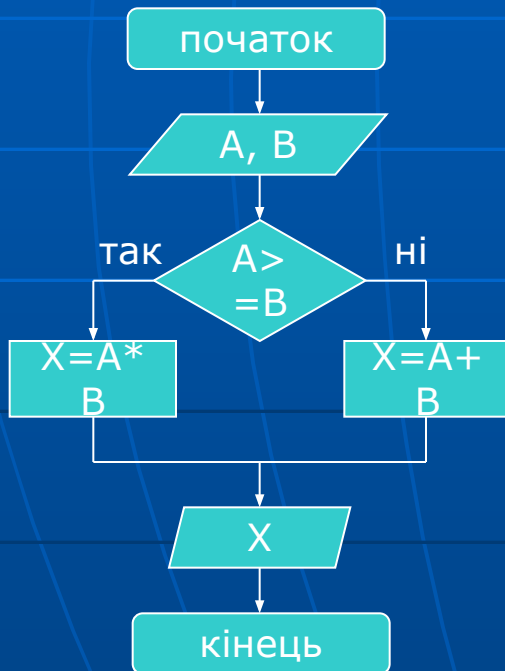
б) $A := (x + y) / (x - y); B := \text{Sqr}(A) - \text{Sqr}(1 / A);$

в) $A := (x - y) / (x + y); B := \text{Sqr}(1 / A) - \text{Sqr}(A);$

Актуалізація опорних знань

Завдання 3

Подивіться уважно на фрагмент алгоритму. Яким буде значення змінної X при $A = 5$, $B = 4$?



Завдання 4

Подивіться уважно на фрагмент алгоритму. Яким буде значення змінних p та d після його виконання?

```
k := 47; Case k Mod 9 Of
5: Begin d := k; p := True End;
0..2: Begin d := 2; p := False End;
8: Begin d := 1; p := False End
Else Begin d := 1; p := True End
End;
```

- 1) $p = \text{True}$, $d = 1$; 2) $p = \text{False}$, $d = 2$;
- 3) $p = \text{False}$, $d = 3$; 4) $p = \text{True}$, $d = 47$;
- 5) $p = \text{True}$, $d = 2$.

Вивчення нового матеріалу

Опис циклічних алгоритмів

Циклічним називається такий обчислювальний процес, який повторює одну і ту ж визначену послідовність операторів скінченну кількість разів.

В мові програмування Паскаль використовують 3 види циклів:

- цикл з лічильником (з параметром) – for;
- цикл з передумовою – while;
- цикл з післяумовою – repeat ... until.

Вивчення нового матеріалу

Опис циклічного алгоритму з лічильником

У випадках, коли наперед відома кількість повторень, використовують оператор циклу з лічильником.

for <змінна>:=<початкове_значення> **to** <кінцеве_значення> **do** <оператор> - загальний запис цикла мовою Паскаль.

Для цикла з лічильником у загальному вигляді ще присутній такий параметр як **крок**, але в мові Паскаль крок сталий $i = 1$.

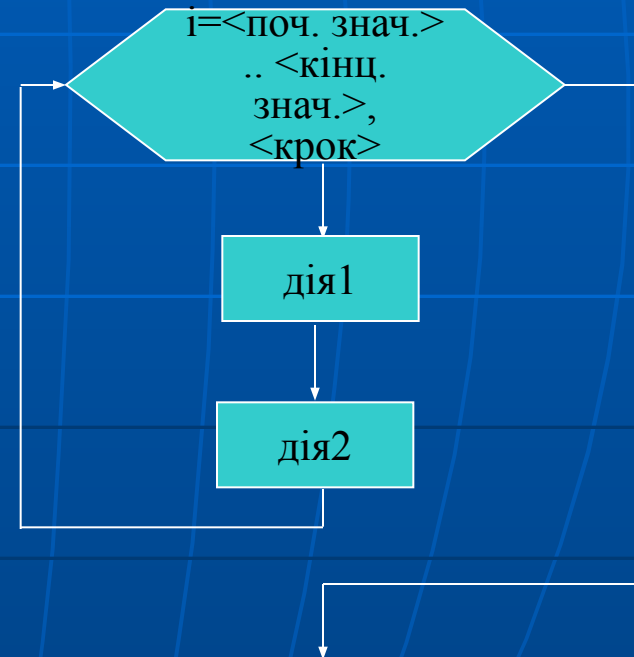
Замість ключового слова **to** можна записати інше ключове слово – **downto**, при цьому початкове значення повинно бути більше за кінцеве.

Якщо в тілі циклу виконуються більше однієї дії, то вони беруться в операторні дужки:

```
begin
```

```
...
```

```
end;
```



Вивчення нового матеріалу

Опис циклічного алгоритму з передумовою

У випадках, коли наперед невідома кількість повторень, використовують цикл з передумовою або післяумовою.

while <умова> **do** <оператор> - загальний запис цикла мовою Паскаль.

При виконанні цього оператора спочатку перевіряється умова (деякий логічний вираз) і, у випадку, коли це значення правдиве, виконується оператор.

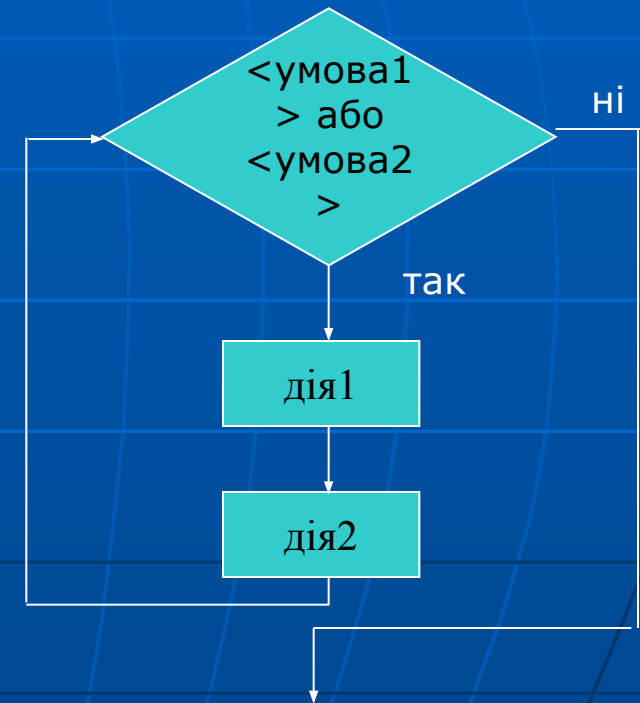
Можливий варіант, коли для виконання тіла циклу недостатньо перевірки однієї умови, тоді записуються декілька умов, між ними ставляться логічні сполучники “і”, “або”, а самі умови беруться в дужки.

Якщо в тілі циклу виконується більше однієї дії, то вони беруться в операторні дужки:

```
begin
```

```
...
```

```
end;
```



Вивчення нового матеріалу

Опис циклічного алгоритму з післяумовою

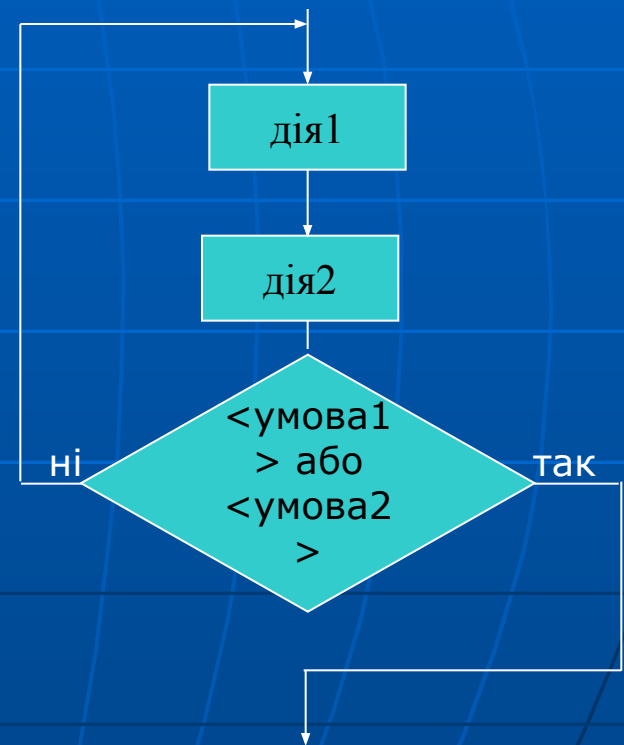
У випадках, коли наперед невідома кількість повторень, використовують цикл з передумовою або післяумовою.

repeat <оператор> **until** <умова>- загальний запис цикла мовою Паскаль.

При виконанні цього оператора спочатку спочатку виконується оператор, тобто тіло циклу, потім перевіряється умова. У випадку, коли результат перевірки неправдивий, тоді виконується тіло циклу ще раз. В протилежному випадку управління передається наступному оператору, який розташований відразу після циклу, тобто відбувається вихід з циклу.

Можливий варіант, коли для виходу з тіла циклу недостатньо перевірки однієї умови, тоді записуються декілька умов, між ними ставляться логічні сполучники “і”, “або”, а самі умови беруться в дужки.

Якщо в тілі циклу виконується більше однієї дії, то вони не беруться в операторні дужки, так як **repeat** ... **until** їх замінюють.



Порівняння циклу з передумовою, циклу з післяумовою та циклу з лічильником

- В операторі **WHILE** цикл виконується при істинності умови, а цикл **REPEAT** – поки умова хибна. Ще одна відмінна риса – для циклу **REPEAT** при записі в тілі циклу декількох операторів немає необхідності брати їх в операторні дужки `begin...end`, оскільки цей оператор сам виступає в цій ролі: `repeat...until` їх замінює.
- В операторі **REPEAT** тіло циклу виконується хоча б 1 раз.
- При використанні як оператора **WHILE**, так і оператора **REPEAT**, потрібно мати на увазі те, що змінні чи вирази, які записані в умові циклу – між словами **WHILE** і **DO** та після **UNTIL**, в тілі циклу повинні змінювати своє значення, в іншому випадку отримаємо “вічний” цикл.
- При використанні циклу з лічильником значення параметру змінюється автоматично через 1 (або в сторону збільшення, або в сторону зменшення). Отже цикл з лічильником не може бути вічним.

Вивчення нового матеріалу

Завдання 5

Скласти блок-схему алгоритму та написати програму на мові програмування Паскаль для обчислення значення функції на проміжку [c; d] з кроком 1:

$$y = \frac{e^{3x}}{2a + b}$$



```
program obchislennya2;
uses dos, crt;
var a, b, y : real;
    c,d,i:integer;
begin
  clrscr;
  write('Введіть значення a: ');
  readln(a);
  write('Введіть значення b: ');
  readln(b);
  write('Введіть значення c: ');
  readln(c);
  write('Введіть значення d: ');
  readln(d);
  if (2*a+b)=0 then write('Введені некоректні дані!')
  else
    for i:=c to d do
    begin
      y:=exp(3*i)/(2*a+b);
      writeln('При x= ', i,'y= ',y:3:2);
    end;
  readln
end.
```

Завдання 6

Скласти блок-схему алгоритму та написати програму на мові програмування Паскаль для обчислення значення функції на проміжку [c; d] з кроком 0,5:

$$y = \frac{e^{3x}}{2a + b}$$

```

program obchislennya2;
uses dos, crt;
var a, b, y, c, d : real;
begin
  clrscr;
  write('Введіть значення a: ');
  readln(a);
  write('Введіть значення b: ');
  readln(b);
  write('Введіть значення c: ');
  readln(c);
  write('Введіть значення d: ');
  readln(d);
  i:=c;
  if (2*a+b)=0 then write('Введені
некоректні дані!')
  else
  while i<=d do
  begin
    y:=exp(3*i)/(2*a+b);
    writeln('При x= ', i,'y= ',y:3:2);
    i:=i+0.5;
  end;
  readln
end.
    
```

