

Лабораторна робота № 5

Тема: Програми з циклом із лічильником

Мета: Навчитись створювати програми з циклом і з лічильником

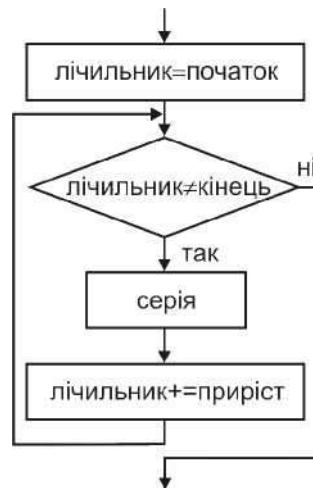
Теоретичні відомості

Цикли. Цикл із лічильником

Види циклів

Часто при виконанні алгоритмів необхідно багаторазово повторювати одні і ті ж дії. Наприклад, щоб зобразити ромашку, потрібно намалювати багато однакових пелюсток; щоб викопати яму, потрібно багато разів повторити подібні рухи лопатою тощо. У таких випадках використовують циклічну структуру або структуру «повторення».

Цикл - це форма організації дій, при якій одна і та ж послідовність дій виконується кілька разів доти, поки виконується деяка умова.



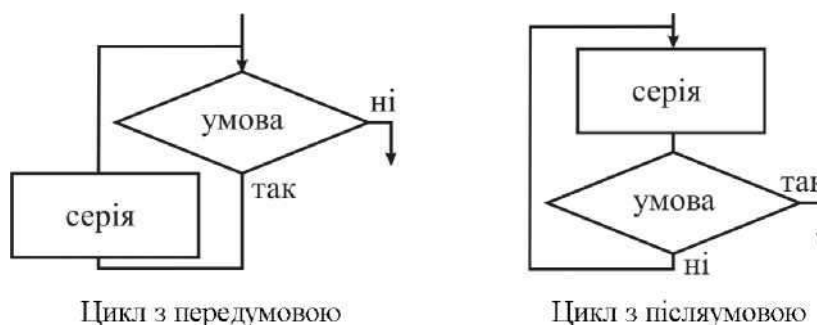
Серія команд, що повторюється без змін при кожному проході циклу (ітерації), називається **тілом циклу**.

Першим розглянемо **цикл з лічильником** (див. блок-схему). Такий цикл використовується, коли заздалегідь відомо, скільки разів треба виконати тіло циклу. Лічильник набуває початкового значення, перевіряється умова, і якщо вона істинна, то виконується серія операторів, а також лічильник змінюється на величину приросту. Потім все повторюється доти, поки умова не перестане виконуватися.

Є ще два типи повторення: з **передумовою** та з **післяумовою**. Такі цикли зручно використовувати, коли заздалегідь не відомо, скільки разів буде виконуватися тіло циклу.

У першому випадку спочатку перевіряється умова, і якщо вона істинна, то тіло циклу виконується черговий раз, якщо ж ні – повторення серії операторів припиняється.

У випадку повторення з післяумовою, спочатку відбувається виконання вказаної дії, а після цього визначається, чи є потреба виконувати її знову. Причому, в цьому випадку повторення відбувається в разі, якщо умова не виконується (див. блок-схеми).



Можливі ситуації, коли тіло циклу з передумовою не виконається жодного разу. Це відбувається в тому випадку, коли при першій перевірці умови значення логічного виразу є хибним.

Якщо при повторенні циклу умова незмінно залишається істинною, то цикл може повторюватися нескінченно.

Цикл з післяумовою обов'язково виконається хоча б один раз, тому що спочатку виконується серія операторів, а потім перевірка умови.

Цикл з лічильником for

Розглянемо синтаксис оператора циклу з лічильником у C++:

```
for (оператор1; вираз1; вираз2)  
    оператор_тіла_циклу;
```

Послідовність його виконання така:

крок 1: виконується оператор1;

крок 2: обчислюється вираз1, і якщо він істинний, то виконується оператор тіла циклу. Якщо хибний, то виконання циклу припиняється і

виконується наступний оператор;

крок 3: обчислюється вираз2;

крок 4: повторюються кроки 2-4.

Наприклад, у програмі може зустрітися такий цикл з лічильником:

```
for (int i = nc; i <= kc; i++)  
    cout<<i;
```

for - ключове слово, що означає початок циклу; **i** - змінна циклу, якій присвоюється початкове значення лічильника **nc**;

kc - значення лічильника, після досягнення якого цикл завершується;

i++ - зміна лічильника циклу. Якщо крок не дорівнює одиниці, можна, наприклад, написати: **i=i+5**, чи **i=i+0.25**. Можливе скорочення запису: **i+=5** у першому випадку та **i+= 0.25** у другому. Не забувайте, що у другому випадку змінна циклу повинна мати тип **float!**

В даному випадку виконання циклу здійснюється за такою схемою:

1. Параметр **i** одержує значення **nc**.
2. Робиться перевірка, чи не перевищує значення змінної циклу кінцевого значення лічильника **kc**.
3. Якщо не перевищує, то виконуються оператори у фігурних дужках. Якщо оператор у циклі один, його можна у дужки не брати.
4. Цикл закінчує роботу, як тільки умова **i<=kc** стане хибною.

Розглянемо приклад:

```
#include<iostream.h>                //Програма 3.1  
#include<conio.h>  
int main()  
{  
    int num;  
    for (num = 1; num<=100; num++)  
        cout<<num<<" ";  
    getch();return 0;  
}
```

Як бачите, у цій програмі в операторі **for** змінна **num** спочатку отримує значення 1. Потім здійснюється перевірка, чи виконується умова **num<=100**. Якщо це так, то виконується відповідний оператор тіла циклу (**cout<<num<<" "**;) і **num** збільшується на 1 (**num++** або **num = num + 1**). Потім

перевірка і подальші кроки повторюються.

Складемо програму, при виконанні якої комп'ютер «просить» увести число, при якому цикл повинен завершитися, а потім роздрукує всі числа від нуля до уведеного числа:

```
#include<iostream.h> //Програма 3.2
#include<conio.h>
int main()
{
int i,n;
cout<<"Vvedit' chislo: "; cin>>n;
for (i = 0; i <= n; i++)
    cout<<i<<" ";
getch();return 0;
}
```

Тілом циклу **for** може бути складений оператор. Наступна програма підсумує всі цілі числа від 1 до 10, вивівши при цьому покроковий коментар:

```
#include<iostream.h> //Програма 3.3
#include<conio.h>
int main()
{
int i,sum=0;
for (i = 1; i <= 10; i++)
{
    cout<<"Dodayu "<<i<<" do "<<sum; //тіло циклу
    sum = sum + i; // - складений
    cout<<"- oderguyu: "<<sum<<endl; //оператор
}
getch();return 0;
}
```

Лічильник циклу можна не тільки збільшувати, а й зменшувати. Заголовок циклу може бути, наприклад, таким:

```
for (i = 10; i >= 1; i--)
```

Внесіть зміни у попередню програму і випробуйте її. У цьому випадку змінна циклу з кожним кроком зменшується на одиницю.

Змінимо програму так, щоб підсумовування цілих чисел від **1** до **n (n>1)** здійснювалося доти, поки значення суми не перевищить уведене значення **Smax**. У результаті на екран буде виведено кількість проведених операцій додавання:

```

... //Програма 3.3.1
int i, Smax, n, sum=0;
int k=0; //лічильник операцій додавання
cout<<"Vvedit n:"<<cin>>n;
cout<<"Vvedit Smax:"<<cin>> Smax;
    for (i = 1; i <= n; i++)
        if (sum<=Smax)
            {
                sum = sum + i;
                k++; //приріст лічильника на одиницю
            }
cout<<" Wikonano: "<<k<<" operaciy"<<endl;
...

```

Основний цикл програми 3.3.1 можна записати ще коротше:

```

... //Програма 3.3.2
for (i = 1; sum<=Smax; i++)
    sum += i;
cout<<" Wikonano: "<<--i<<" operaciy"<<endl;
...

```

*Зверніть увагу, що між заголовком циклу **for** та тілом циклу не ставиться крапка з комою (;)!!!*

Як бачите, перевірка умови **sum<=Smax** тут винесена в заголовок циклу. Крім того, замість змінної **k** в кінці виводиться зменшене на одиницю значення змінної **i**. Отже, такий варіант є кращим з двох причин:

економія пам'яті, завдяки відмові від змінної **k**; **більша швидкодія**, за рахунок зменшення кількості операцій, які повторюються в тілі циклу, з 4-х (<=,=,+,++) до 2-х (+=).

Особливості використання циклу *for*

Як вже було сказано, іноді, через помилки програмування, повторення циклу не припиняється зовсім. Тоді кажуть, що програма «зациклилася». Причиною «зациклення» є те, що умова припинення циклу не може стати істинною. Уникайте таких помилок.

Приклад зациклення:

```

for (i = 0; i < 100; value++)
    cout<<i;

```

Цей цикл мав би припинитись при досягненні змінною циклу **i** значення **100**. Але ні у виразах заголовку, ні у тілі циклу значення **i** не змінюється. Як наслідок, значення змінної **i** ніколи не стане рівним 100, і програма буде

працювати нескінченно!

Як вже було сказано, цикли **for** не обмежуються використанням в якості лічильника циклу змінних типу **int**. Наприклад, наступна програма використовує змінну циклу типу **char (letter)** для виведення букв латинського алфавіту у першому циклі й змінну типу **float (value)** для виведення чисел із плаваючою крапкою в іншому циклі:

```
#include<iostream.h> //Програма 3.4
#include<conio.h>
int main()
{
char letter;
float value;
    for (letter = 'A'; letter<='Z'; letter++)
        cout<<letter;
    cout<<endl;
    for (value = 0.0; value<=1.0; value+=0.1)
        cout<<value<<" ";
getch();
return 0;
}
```

Хід роботи

Завдання 1.

1. Випробуйте програму 3.1. Експериментуючи із цією програмою, змініть значення 100 на 10, 30 і навіть 4000.
2. Випробуйте програму 3.2. Що відбудеться при $n = -1$?
3. Випробуйте програму 3.3. Експериментуючи із програмою, замініть 10 іншими значеннями. Потім замість $i++$ уведіть $i+=5$.

Переробіть програму, як у прикладі 3.3.1. Експериментуйте, уводячи різні значення n та S_{max} .

Завдання 2.

Варіант 1

1. Написати програму, яка виводить на екран ваше ім'я 10 разів.
2. Надрукувати всі натуральні числа від 1 до введеного з клавіатури n та їх квадрати у вигляді таблиці:

1 1
2 4
3 9 і т.д.

3. Написати програму, яка виводить на екран таблицю вартості, наприклад, яблук в діапазоні від 100 г до 1 кг з кроком 100 г. Нижче наведено рекомендований вигляд екрану програми під час її роботи (дані, введені користувачем, виділені напівжирним шрифтом).

Введіть ціну одного кілограма і натисніть <Enter> (копійки від грн відокремлюйте крапкою) **16.50**

Вага (гр)	Вартість (грн.)
<i>100</i>	<i>1.65</i>
<i>200</i>	<i>3.30</i>
<i>300</i>	<i>4.95</i>
<i>400</i>	<i>6.60</i>
<i>500</i>	<i>8.25</i>
<i>600</i>	<i>9.90</i>
<i>700</i>	<i>11.55</i>
<i>800</i>	<i>13.20</i>
<i>900</i>	<i>14.85</i>
<i>1000</i>	<i>16.50</i>

4. Напишіть програму, яка виводить на екран зображення шахової дошки. Чорні клітини відображати "зірочкою", білі - пропуском. Рекомендований вигляд екрана під час виконання програми наведено нижче.

```
* * * *  
 * * * *  
* * * *  
 * * * *  
* * * *  
 * * * *  
* * * *  
 * * * *
```

Варіант 2

1. Написати програму, яка виводить на екран ваше прізвище 10 разів.

2. Капосний папуга навчився висмикувати у дідуся Івана волосся, яке ще залишилось у того на голові. Почавши з однієї волосини, він кожен день

збільшував порцію вдвічі. Через скільки днів дідусеві не знадобиться гребінець, якщо на початку в нього було аж N волосин.

3. Написати програму, яка виводить таблицю степенів двійки від нульової до десятої. Нижче наведено рекомендований вигляд екрану під час роботи програми.

Таблиця ступенів двійки

0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	1024

4. Написати програму, яка виводить таблицю квадратів перших десяти цілих позитивних чисел. Нижче наведено рекомендований вигляд екрану під час роботи програми.

Таблиця квадратів.

<i>Число</i>	<i>Квадрат</i>
1	1
2	4
3	9
4	16
5	25
6	36
7	49
8	64
9	81
10	100

Завдання 3

1. Збережіть програми у папці з назвою лабораторної роботи.
2. Дайте відповіді на контрольні запитання.

Контрольні запитання:

1. Що таке «цикл»?
2. Як виконується цикл з лічильником?
3. Як виконується цикл з передумовою?
4. Чи може тіло циклу з передумовою не виконатися жодного разу?
5. Як виконується цикл з післяумовою?
6. Чи може тіло циклу з післяумовою не виконатися жодного разу?
7. Що таке «зациклення»?
8. Чи можливо у циклі for в якості змінних циклу використовувати змінні типу char; типу float?