

## Лабораторна робота №3. Циклічні структури

**Тема.** Програмування алгоритмів циклічної структури. Документування коду.

**Мета.** Придбати практичні навички з розроблення циклічних програм, використання операторів *for*, *while*, *do-while*.

**Теми для попереднього опрацювання:**

- оператор *sizeof*;
- оператори циклу *for*, *while*, *do-while*;
- оператори переходу *break*, *continue*.

**Загальні відомості**

**Циклічний алгоритм** – це алгоритм, який містить такі фрагменти (послідовність операторів), які багаторазово виконуються при різних значеннях проміжних даних. Група дій, що повторюються в циклі, називається його тілом. Однократне виконання циклу називається його кроком.

Основою таких алгоритмів є оператори циклу. Число повторень цих операторів може бути задане в явній формі (цикл із відомим заздалегідь числом повторень) або в неявній формі (цикл із невідомим заздалегідь числом повторень).

**Оператори циклів.** Розрізняють:

- ітераційні цикли;
- арифметичні цикли.

В ітераційних циклах умова виконання циклу відома.

**1. Цикл з передумовою:**

```
while (вираз-умова)
{
    оператор;
}
```

Як <вираз-умова> найчастіше використовується відношення або логічний вираз. Якщо результат виразу відмінний від 0 (*true*), тіло циклу виконується; якщо вираз-умова стане *false* – цикл закінчується.

Наприклад,

```
int a=0, sum = 0;
while (a!= 10)
{
    sum+=a++;
}
```

**2. Цикл із післяумовою:**

```
do
{
    оператор;
}
while (вираз-умова);
```

Тіло циклу виконується, поки вираз-умова дійсний (*true*).

Наприклад,

```
int a=0, sum = 0;
do
{
    sum+=a++;
}
while (a <= 10);
```

**3. Цикл з параметром:**

```
for ( вираз_1; вираз-умова; вираз_3)
```

```

{
    оператор;
}

```

Вираз\_1 та вираз\_3 можуть складатися з декількох виразів, розподілених комами. Вираз\_1 задає початкові умови для циклу (ініціалізація). Вираз-умова визначає умову виконання циклу: якщо вона відмінна від 0, цикл виконується, а потім обчислюється значення виразу\_3. Вираз\_3 задає зміну параметра циклу або інших змінних (корекція). Цикл триває доти, поки вираз-умова не стане дорівнювати 0.

```
for ( int a=0, sum = 0; a <= 10 ; a++)s+=a;
```

Будь-який вираз може бути відсутнім, але поділяючи їх « ; » повинні лишатися завжди.

Наприклад:

```
int sum = 0;int a = 0;
for ( ; a <= 10 ; a++)s+=a;
```

**Оператори переходу.** Виконують безумовну передачу керування:

**break** – оператор, що перериває виконання операторів *while*, *do-while*, *for* та *switch*. Управління передається наступному оператору за перериванням. Як що *break* перериває виконання вкладеного циклу, то управління передається циклу, який охоплює перериваний. Використовується, якщо умову продовження ітерацій треба перевіряти в середині циклу;

**continue** – перехід до наступної ітерації циклу *while*, *do-while* *for*. У циклах *while*, *do-while* наступна ітерація починається з обчислення умовного виразу, у циклі *for* – з обчислення виразу для зміни параметра циклу, а потім умовного виразу.

Текст програми необхідно супроводжувати *Doxygen* коментарями.

## **ОСНОВНЕ ЗАВДАННЯ**

Реалізувати програму відповідно до індивідуального завдання за допомогою трьох типів циклів: *for*, *while-do*, *do-while* (отримати три однакових результати).

### **Індивідуальні завдання.**

1. Визначити значення  $10 * n!$  ( $10 * \text{факторіал числа } n$ ).
2. Для заданого цілого числа визначити подвійний факторіал. Наприклад,  $6!! = 2 * 4 * 6 \rightarrow 48$ ,  $7!! = 1 * 3 * 5 * 7 \rightarrow 105$ .
3. В заданому цілому числі визначити кількість розрядів та суму його цифр. Наприклад, число 123456 має 6 розрядів, сума його цифр – 21.
4. Визначити найбільший спільний дільник для двох заданих чисел.
5. Визначити зворотне число для заданого цілого числа. Кількість розрядів від самого початку не визначено. Наприклад, зворотне число для 12334  $\rightarrow$  43321.
6. Визначити, чи є задане ціле число простим.
7. Для заданого парного числа визначити подвійний факторіал. Наприклад,  $6!! = 2 * 4 * 6 \rightarrow 48$ .
8. Визначити, чи є ціле  $6$  значне число «щасливим» квитком («щасливий квиток» – квиток, в якому сума першої половини чисел номера дорівнює сумі другої половини. Наприклад, 102300  $\rightarrow$  1+0+2=3+0+0).
9. Визначити кількість щасливих квитків, номери яких складаються з 4 цифр.
10. Визначити суму чисел, які кратні 3 та 5 одночасно, із заданого діапазону. Діапазон задається двома змінними (початок діапазону, кінець діапазону включно).
11. Визначити кількість цифр у заданому цілому числі.
12. У заданому діапазоні цілих чисел визначити останнє число, що ділиться на 7 без залишку.
13. Визначити кількість парних (2, 4, 6, 8...) чисел у заданому діапазоні.
14. Обчислити загальну суму вкладу через  $N$  років із заданою початковою сумою вкладу та заданою процентною ставкою.

15. Знайти добуток усіх чисел із заданого діапазону, що є парними та діляться на 3 без залишку.
16. Знайти найближче просте число, що більше заданого.
17. Для заданого непарного числа визначити подвійний факторіал. Наприклад,  $7!! = 1 * 3 * 5 * 7 \rightarrow 105$ .
18. Знайти найближче просте число, що менше за задане.
19. Знайти найбільше 4 значне число, сума цифр якого дорівнює заданому числу.
20. Якою мінімальною кількістю купюр можна набрати потрібну суму  $S$  за умови, що в наявності є купюри номіналом 1, 2, 5 та 10 грн.?
21. Число, яке дорівнює сумі своїх дільників, називається досконалим числом. Наприклад,  $6 = 1+2+3$ . Визначити, чи є задане число досконалим.

*Додаткові умови виконання завдання:*

- звіт має бути виконаний згідно з вимогами до оформлення лабораторних робіт.

*Контрольні питання.*

1. Як записується і як працює оператор *for*?
2. У чому відмінність оператора *while* від оператора *do while*?
3. Як програмуються циклічні алгоритми з явно заданою кількістю повторень циклу?
4. Як програмуються циклічні алгоритми із заздалегідь невідомим числом повторень циклу?
5. Напишіть оператор циклу, який не виконується жодного разу.
6. Напишіть оператор циклу, який виконується необмежену кількість раз.
7. Замініть фрагмент програми з оператором *for* рівнозначним фрагментом програми з оператором *while*.
8. Замініть фрагмент програми з оператором *for* рівнозначним фрагментом програми з оператором *do while*.
9. Як можна перервати виконання оператора циклу?
10. Яке призначення операторів *break* і *continue*?