

Тема 4. Цикли (for, while, do-while).

Цикли в C++ дозволяють виконувати певний блок коду кілька разів, поки виконується задана умова. У мові є три основних види циклів: for, while, do-while. **Давайте розглянемо їх детальніше.**

1. Цикл for

Використовується, коли відома кількість повторень.

for (ініціалізація; умова; оновлення) {

 // Тіло циклу

}

Ініціалізація: задає початкове значення змінної циклу.

Умова: перевіряється перед кожною ітерацією; якщо умова хибна, цикл завершується.

Оновлення: виконується після кожної ітерації.

Приклад:

for (int i = 0; i < 5; i++) {

 std::cout << "Ітерація: " << i << std::endl;

}

2. Цикл while

Використовується, коли кількість ітерацій заздалегідь невідома, але залежить від виконання умови.

while (умова) {

 // Тіло циклу

}

Умова перевіряється перед кожною ітерацією. Якщо умова хибна, цикл завершується.

Приклад:

int i = 0;

while (i < 5) {

 std::cout << "Ітерація: " << i << std::endl;

 i++;

}

3. Цикл do-while

Цей цикл гарантує, що тіло циклу виконається хоча б один раз, оскільки умова перевіряється після виконання тіла.

```
do {  
    // Тіло циклу  
} while (умова);
```

Приклад:

```
int i = 0;  
  
do {  
    std::cout << "Ітерація: " << i << std::endl;  
    i++;  
} while (i < 5);  
  
---
```

Ключові слова для управління циклами:

1. break: вихід із циклу.

```
for (int i = 0; i < 10; i++) {  
    if (i == 5) break;  
    std::cout << i << " ";  
}
```

2. continue: перехід до наступної ітерації.

```
for (int i = 0; i < 10; i++) {  
    if (i % 2 == 0) continue;  
    std::cout << i << " ";  
}
```

3. return: завершує виконання функції, в якій знаходиться цикл.

Загальні поради:

Використовуйте for, якщо кількість ітерацій відома заздалегідь.

Використовуйте `while`, якщо цикл має виконуватися, поки дотримується умова.

Використовуйте `do-while`, якщо хоча б одна ітерація потрібна за будь-яких умов.

Цикли є потужним засобом для автоматизації обчислень, особливо при роботі з великими обсягами даних.