

## Тема 3. Умовні конструкції (if, else, switch).

Умовні оператори в C++ дозволяють виконувати різні дії в залежності від виконання певної умови.

**Основними умовними операторами є:**

### 1. if

Цей оператор використовується для виконання блоку коду, якщо умова є істинною.

```
if (умова) {  
  
    // Код виконується, якщо умова істинна  
  
}
```

Приклад:

```
int x = 10;  
  
if (x > 5) {  
  
    std::cout << "x більше за 5" << std::endl;  
  
}
```

### 2. if-else

Додає альтернативний блок коду, який виконується, якщо умова хибна.

```
if (умова) {  
  
    // Код виконується, якщо умова істинна  
  
} else {  
  
    // Код виконується, якщо умова хибна  
  
}
```

Приклад:

```
int x = 3;  
  
if (x > 5) {  
  
    std::cout << "x більше за 5" << std::endl;  
  
} else {  
  
    std::cout << "x не більше за 5" << std::endl;  
  
}
```

### 3. if-else if-else

Дозволяє перевіряти кілька умов.

```
if (умова1) {
```

```

    // Код виконується, якщо умова1 істинна
} else if (умова2) {

    // Код виконується, якщо умова2 істинна
} else {

    // Код виконується, якщо жодна з умов не є істинною
}

```

Приклад:

```

int x = 8;

if (x > 10) {

    std::cout << "x більше за 10" << std::endl;

} else if (x > 5) {

    std::cout << "x більше за 5, але не більше за 10" << std::endl;

} else {

    std::cout << "x не більше за 5" << std::endl;

}

```

#### 4. Тернарний оператор (?:)

Це скорочена форма оператора if-else.

результат = (умова) ? вираз1 : вираз2;

Приклад:

```

int x = 7;

std::string результат = (x > 5) ? "Більше за 5" : "Не більше за 5";

std::cout << результат << std::endl;

```

#### 5. switch

Використовується для перевірки значення змінної на відповідність одному з кількох варіантів (case).

```

switch (змінна) {

    case значення1:

        // Код для значення1

        break;

    case значення2:

        // Код для значення2

```

```
        break;

    default:

        // Код за замовчуванням
}

```

Приклад:

```
int x = 2;

switch (x) {

    case 1:

        std::cout << "x дорівнює 1" << std::endl;

        break;

    case 2:

        std::cout << "x дорівнює 2" << std::endl;

        break;

    default:

        std::cout << "x не дорівнює 1 чи 2" << std::endl;

}

```

#### **Важливі моменти:**

Усі умови в if, else if або switch повинні бути булевими (true або false).

В switch потрібно використовувати break для уникнення "провалювання" до наступного блоку.

Умови тернарного оператора слід використовувати для простих умов.

Ці оператори є базовими для контролю потоку виконання програми в C++.