

Тема 5. Функції: оголошення, виклик, передача аргументів, повернення значень.

У C++ функції є основним інструментом для організації коду, дозволяючи розділяти програму на логічно відокремлені частини. Розглянемо основні аспекти роботи з функціями.

1. Оголошення функції

Оголошення функції визначає її ім'я, тип поверненого значення, параметри (за необхідності) та тіло.

Синтаксис:

```
тип_повернення ім'я_функції(список_параметрів) {  
    // Тіло функції  
}
```

Приклади:

```
int add(int a, int b) { // функція приймає два параметри та повертає їх суму  
    return a + b;  
}
```

```
void printMessage() { // функція не приймає параметрів і не повертає значення  
    cout << "Hello, World!" << endl;  
}
```

2. Виклик функції

Функцію можна викликати, звертаючись до неї за ім'ям і передаючи відповідні аргументи, якщо це потрібно.

Приклад:

```
int result = add(5, 10); // Виклик функції add із аргументами 5 і 10  
printMessage(); // Виклик функції printMessage
```

3. Передача аргументів

а. За значенням

Параметри копіюються у функцію.

Зміни параметрів у функції не впливають на початкові змінні.

Приклад:

```
void modifyValue(int x) {  
    x = 42; // Локальна зміна
```

```
}  
  
int main() {  
  
    int a = 10;  
  
    modifyValue(a);  
  
    cout << a; // Виведе 10, бо x змінювався локально  
  
    return 0;  
  
}
```

b. За посиланням

Параметри передаються у функцію за посиланням, що дозволяє змінювати їх значення.

Приклад:

```
void modifyValue(int& x) {  
  
    x = 42; // Змінює значення змінної у викликаному контексті  
  
}  
  
int main() {  
  
    int a = 10;  
  
    modifyValue(a);  
  
    cout << a; // Виведе 42  
  
    return 0;  
  
}
```

c. За вказівником

Передаються адреси змінних, дозволяючи змінювати їх значення.

Приклад:

```
void modifyValue(int* x) {  
  
    *x = 42; // Зміна значення через вказівник  
  
}  
  
int main() {  
  
    int a = 10;  
  
    modifyValue(&a);  
  
    cout << a; // Виведе 42  
  
    return 0;
```

```
}
```

4. Повернення значень

Функція може повертати значення за допомогою ключового слова `return`.

Якщо функція повертає значення, його тип вказується перед ім'ям функції.

Якщо функція нічого не повертає, використовується тип `void`.

Приклади:

```
int square(int x) { // Повертає квадрат числа
```

```
    return x * x;
```

```
}
```

```
void display() { // Не повертає значення
```

```
    cout << "This is a void function!" << endl;
```

```
}
```

5. Перевантаження функцій

C++ дозволяє створювати кілька функцій із однаковим ім'ям, але різними параметрами.

Приклад:

```
int multiply(int a, int b) {
```

```
    return a * b;
```

```
}
```

```
double multiply(double a, double b) {
```

```
    return a * b;
```

```
}
```

```
int main() {
```

```
    cout << multiply(2, 3) << endl;    // Виклик версії з цілими числами
```

```
    cout << multiply(2.5, 3.5) << endl; // Виклик версії з числами типу double
```

```
    return 0;
```

```
}
```

6. Рекурсія

Функція може викликати сама себе, що називається рекурсією.

Приклад:

```
int factorial(int n) {  
    if (n == 0) return 1;  
    return n * factorial(n - 1); // Рекурсивний виклик  
}
```

```
int main() {  
    cout << factorial(5); // Виведе 120  
    return 0;  
}
```

Приклад програми

```
#include <iostream>  
  
using namespace std;  
  
// Функція для обчислення суми  
int add(int a, int b) {  
    return a + b;  
}  
  
// Функція для виведення повідомлення  
void printMessage() {  
    cout << "C++ Functions Example" << endl;  
}  
  
int main() {  
    printMessage();          // Виклик функції без параметрів  
    int sum = add(15, 25);    // Виклик функції з аргументами  
    cout << "Sum: " << sum << endl;  
    return 0;  
}
```