

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

У мові програмування C функції використовуються для організації коду в окремі блоки, які можна викликати для виконання певних операцій. Це дозволяє робити програму більш зрозумілою, організованою та зручною для повторного використання. Ось основні аспекти роботи з функціями в C:

### 1. Оголошення функції

Оголошення функції включає в себе визначення типу її результату, назву та параметри (якщо вони є). Оголошення функції має бути розташоване перед її використанням у програмі або в заголовочному файлі.

#### Синтаксис оголошення функції:

тип\_повернення ім'я\_функції(тип\_аргументу1 ім'я\_аргументу1, тип\_аргументу2 ім'я\_аргументу2, ...);

- **Тип повернення** — визначає тип значення, яке функція повертає після виконання (наприклад, int, float, void — якщо функція нічого не повертає).
- **Ім'я функції** — назва функції.
- **Аргументи (параметри)** — це змінні, які передаються функції для обробки.

#### Приклад оголошення функції:

```
int add(int a, int b); // Функція, що повертає суму двох чисел
```

### 2. Визначення функції

Після оголошення функції її необхідно визначити. У визначенні ви вказуєте, що саме має виконувати функція. Це може бути блок коду, який виконується при виклику функції.

#### Синтаксис визначення функції:

```
тип_повернення ім'я_функції(тип_аргументу1 ім'я_аргументу1, тип_аргументу2 ім'я_аргументу2, ...) {  
    // Тіло функції  
    // Операції з аргументами та інші дії  
    return значення; // Якщо функція має повертати значення  
}
```

#### Приклад визначення функції:

```
int add(int a, int b) {  
    return a + b;  
}
```

### 3. Виклик функції

Для виклику функції використовується її ім'я та передача необхідних аргументів. Після виклику функція виконується, і її результат повертається (якщо це передбачено).

#### Синтаксис виклику функції:

```
ім'я_функції(аргумент1, аргумент2, ...);
```

#### Приклад виклику функції:

```
int result = add(5, 3); // Викликається функція add, результат буде збережено в змінній result
```

## 4. Передача аргументів функціям

У С існують два способи передачі аргументів функціям: **передача за значенням** та **передача за посиланням** (через вказівники).

#### Передача за значенням

Коли аргументи передаються за значенням, функція отримує копії значень змінних, і зміни в цих значеннях не впливають на вихідні змінні.

#### Приклад:

```
void increment(int x) {  
    x = x + 1; // Зміни в x не впливають на зовнішню змінну  
}  
  
int main() {  
    int a = 5;  
    increment(a);  
    printf("%d", a); // Виведе 5, оскільки x в функції increment - це копія a  
    return 0;  
}
```

#### Передача за посиланням (через вказівники)

Якщо потрібно змінити значення змінної, переданої в функцію, можна передавати її за допомогою вказівника. Це дозволяє функції працювати безпосередньо з оригінальними даними.

#### Приклад:

```
void increment(int *x) {  
    *x = *x + 1; // Зміна значення за допомогою вказівника  
}  
  
int main() {  
    int a = 5;  
    increment(&a); // Передача адреси змінної a  
    printf("%d", a); // Виведе 6, оскільки значення a було змінено через вказівник  
    return 0;  
}
```

## 5. Повернення значень з функцій

Функції в С можуть повертати значення після виконання. Для цього в тілі функції використовується ключове слово `return`, яке дозволяє повернути значення в точку виклику функції.

#### Синтаксис:

return значення;

- **Значення**, яке функція повертає, повинно відповідати типу функції (наприклад, для функції, що повертає тип `int`, треба використовувати ціле число).

#### Приклад:

```
int add(int a, int b) {  
    return a + b; // Повертає суму a і b  
}  
  
int main() {  
    int result = add(2, 3); // result буде містити 5  
    printf("%d", result);  
    return 0;  
}
```

#### Підсумок

1. **Оголошення функції** визначає її тип, ім'я та параметри.
2. **Визначення функції** включає реалізацію коду, що виконується при виклику функції.
3. **Виклик функції** передбачає використання її імені та передачу необхідних аргументів.
4. **Передача аргументів** може бути за значенням або за посиланням (через вказівники).
5. **Повернення значення** з функції здійснюється через ключове слово `return`.

Ці механізми дозволяють створювати модульний код, який легко підтримувати і повторно використовувати.