

Тема: АРГУМЕНТИ ОСНОВНОЇ ФУНКЦІЇ `main()`: `argc` і `argv`

1. Аргументи основної функції `main()`: `argc` і `argv`

Іноді виникає потреба передати інформацію програмі при її запуску. Як правило, це реалізується шляхом передачі аргументів командного рядка функції **`main()`**. Аргумент командного рядка є інформацією, що вказується в команді (командному рядку), призначеною для виконання операційною системою, після імені програми. Наприклад, C++-програми можна компілювати шляхом виконання наступної команди:

```
cl prog_name
```

У цьому записі елемент *prog_name* – ім'я програми, яку ми хочемо скомпілювати. Ім'я програми передається C++-компіляторові як аргумент командного рядка.

Аргумент командного рядка є інформацією, що задається в командному рядку після імені програми.

У мові програмування C++ для основної функції **`main()`** визначено два вбудованих, але необов'язкових параметри, **`argc`** і **`argv`**, які набувають свої значення від аргументів командного рядка. У конкретному операційному середовищі можуть підтримуватися і інші аргументи (таку інформацію необхідно уточнити по документації, що додається до Вашого компілятора).

Формально для імен параметрів командного рядка можна вибирати будь-які ідентифікатори, проте імена `argc` і `argv` використовуються за домовленістю вже протягом декількох років. Тому є сенс не вдаватися до інших імен ідентифікаторів, щоб будь-який програміст, якому доведеться розбиратися у Вашій програмі, зміг швидко ідентифікувати їх як параметри командного рядка.

Параметр **`argc`** має цілочисельний тип і призначений для зберігання кількості аргументів командного рядка. Його значення завжди не менше від одиниці, оскільки ім'я програми також є одним з аргументів, що враховуються.

Параметр **`argv`** є покажчик на масив символічних покажчиків. Кожен

показчик в масиві **argv** посилається на рядок, що містить аргумент командного рядка. Елемент **argv[0]** вказує на ім'я програми; елемент **argv[1]** - на перший аргумент, елемент **argv[2]** - на другий і т.д. Всі аргументи командного рядка передаються програмі як рядки, тому числові аргументи необхідно перетворити у програмі у відповідний внутрішній формат.

Важливо правильно оголосити параметр **argv**, а саме:

```
char *argv[];
```

Доступ до окремих аргументів командного рядка можна отримати шляхом індексації масиву **argv**. Як це зробити, показано у наведеному нижче коді програми. Під час її виконання на екран виводиться вітання («Привіт»), а за ним Ваше ім'я, яке повинно бути першим аргументом командного рядка.

Код програми 7.12. Демонстрація механізму доступу до окремих аргументів командного рядка

```
#include <iostream>          // Потокове введення-виведення
using namespace std;         // Використання стандартного простору імен

int main(int argc, char *argv[])
{
    if(argc !=2) {
        cout << "Ви забули ввести своє ім'я" << endl;
        return 1;
    }
    cout << "Привіт, " << argv[1] << endl;

    getch(); return 0;
}
```

Припустимо, що Ви називаєтеся Тимур і що Ви назвали цю програму іменем name. Тоді, якщо запустити цю програму, ввівши команду name Тимур, результат її роботи повинен виглядати так: Привіт, Тимур. Наприклад, Ви працюєте з диском A, і у відповідь на пропозицію введення команди Ви повинні ввести згадану вище команду і отримати такий результат:

```
A>name Тимур
Привіт, Тимур
A>
```

У мові програмування C++ точно не визначено, як мають бути

представлені аргументи командного рядка, оскільки середовища виконання (операційні системи) мають тут великі відмінності. Проте найчастіше використовують таку домовленість: кожен аргумент командного рядка повинен бути відокремлений пропуском або символом табуляції. Як правило, коми, крапки з комою і подібні до них знаки не є допустимими роздільниками аргументів. Наприклад, рядок

один, два і три

складається з чотирьох рядкових аргументів, тоді як рядок

один, два, три

містить тільки два, оскільки кома не є допустимим роздільником.

Якщо необхідно передати як один аргумент командного рядка набір символів, який містить пропуски, то його потрібно помістити в лапки. Наприклад, цей набір символів буде сприйнятий як один аргумент командного рядка:

«це тільки один аргумент»

Представлені тут приклади застосовуються до широкого діапазону середовищ мови програмування C++, але це не означає, що Ваше середовище входить до їх числа.

Щоб отримати доступ до окремого символу в одному з аргументів командного рядка, під час звернення до масиву **argv** додайте другий індекс. Наприклад, у процесі виконання наведеної нижче програми по-символьно відображаються всі аргументи, з якими вона була викликана.

Код програми 7.13. Демонстрація по-символьного виведення всіх аргументів командного рядка, з яких вона була викликана

```

#include <iostream>          // Потокowe введення-виведення
using namespace std;        // Використання стандартного простору імен

int main(int argc, char *argv[])
{
    for(int t=0; t<argc; ++t) {
        int i=0;
        while(argv[t][i]) {
            cout << argv[t][i];
            ++i;
        }
        cout << " ";
    }

    getch(); return 0;
}

```

Неважко здогадатися, що перший індекс масиву **argv** дає змогу отримати доступ до відповідного аргументу командного рядка, а другий - до конкретного символу цього рядкового аргументу.

Звичайно аргументи **argc** і **argv** використовуються для введення у програму початкових параметрів, початкових значень, імен файлів або варіантів (режимів) роботи програми. У мові програмування C++ можна ввести стільки аргументів командного рядка, скільки допускає операційна система. Використання аргументів командного рядка додає програмі професійний вигляд і дає змогу використовувати її в командному файлі (виконуваному текстовому файлі, що містить одну або декілька команд).

2. Передача програмі числових аргументів командного рядка

Як було зазначено вище, при передачі програмі числових даних як аргументів командного рядка, ці дані приймаються в рядковій формі. У програмі повинно бути передбачено їх перетворення у відповідний внутрішній формат за допомогою однієї із стандартних бібліотечних функцій, які підтримуються мовою C++. Наприклад, у процесі виконання наведеної нижче програми виводиться сума двох чисел, які вказуються в командному рядку після імені програми. Для перетворення аргументів командного рядка у внутрішнє представлення тут використовується стандартна бібліотечна функція

atof(). Вона перетворить число з рядкового формату в значення типу **double**.

Код програми 7.14. Демонстрація механізму знаходження суми двох числових аргументів командного рядка

```
#include <iostream>          // Потокове введення-виведення
#include <cstdlib>            // Використання бібліотечних функцій
using namespace std;         // Використання стандартного простору імен

int main(int argc, char *argv[])
{
    double a, b;
    if(argc !=3) {
        cout << "Використання: add число число \n";
        return 1;
    }
    a = atof(argv[1]);
    b = atof(argv[2]);
    cout << a + b;

    getch(); return 0;
}
```

Щоб додати два числа, використовується командний рядок такого вигляду (припускаючи, що ця програма має ім'я **add**).

```
C>add 100.2 231
```

3. Перетворення числових рядків у числа

Стандартна бібліотека C++ містить декілька функцій, які дають змогу перетворити рядкове представлення числа в його внутрішній формат. Для цього використовуються такі функції, як **atoi()**, **atol()** і **atof()**. Вони перетворюють рядок в цілочисельне значення (типу **int**), довге ціле (типу **long**) і значення з плаваючою крапкою (типу **double**) відповідно. Використання цих функцій (для їх виклику необхідно приєднати до програми заголовний файл **<cstdlib>**) продемонстровано у наведеному нижче коді програми.

Код програми 7.15. Демонстрація механізму використання функцій **atoi(), **atol()** і **atof()****

```

#include <iostream>           // Потокowe введення-виведення
#include <cstdlib>            // Використання бібліотечних функцій
using namespace std;         // Використання стандартного простору імен

int main()
{
    int c;

    long d;
    double f;

    c = atoi("100");
    d = atol("100000");
    f = atof("-0.123");
    cout << c << " " << d << " " << f;
    cout << endl;
    getch(); return 0;
}

```

Результати виконання цієї програми є такими:

```
100 100000 -0.123
```

Функції перетворення рядків корисні не тільки при передачі числових даних програмі через аргументи командного рядка, але і у деяких інших ситуаціях.

Контрольні запитання:

1. Опишіть аргументи основної функції `main()`: `argc` і `argv`.
2. Як передати програмі числові аргументи командного рядка?
3. Що таке перетворення числових рядків у числа?