

ЛАБОРАТОРНА РОБОТА №8

Тема: Функції

Мета: Освоїти методи роботи з функціями

Час: 2 год.

Виконання роботи

- Надати викладачу, виконане завдання для самопідготовки в п. 4.2.
- Вивчити теоретичні відомості.
- Відповісти тестові завдання.
- Виконати самостійну роботу.

Завдання для самопідготовки

В процесі підготовки до заняття студент у обов'язковому порядку повинен виконати наступні завдання:

а) За допомогою конспекту лекцій і рекомендованої літератури розглянути суть таких питань:

- 1) Створення функцій
- 2) Використання функцій.

б) Занести в звіт такі дані:

- 1) номер практичної роботи;
- 2) тему і мету роботи;
- 3) короткий конспект основних теоретичних відомостей.

Практична частина

Контрольний приклад

До сих пір ми писали програми єдиним, функціонально неподільним, кодом. алгоритм програми перебував в головної функції, причому інших функцій в програмі НЕ було. ми писали маленькі програми, тому НЕ було потреби в оголошенні своїх функцій. для написання великих програм, досвід показує, що краще користуватися функціями. програма буде складатися з окремих фрагментів коду, під окремим фрагментом коду розуміється функція. Окремим, тому, що робота окремої функції НЕ залежить від роботи який - небудь інший. те є алгоритм в кожній функції функціонально достатній і НЕ залежний від інших алгоритмів програми. Одного разу написавши функцію, її можна, можливо буде з легкістю переносити в інші програми. Функція (в програмуванні) - це фрагмент коду або алгоритм, реалізований на якому - то мовою програмування, з метою виконання певної послідовності операцій. Отже, функції дозволяють зробити програму модульною, то є розділити програму на кілька маленьких підпрограм (функцій), які в сукупності виконують поставлену задачу. ще один величезний плюс функцій в тому, що їх можна, можливо багаторазово використовувати. Дана можливість

дозволяє багаторазово використовувати один раз написаний код, що в свою чергу,набагато скорочує Об`єм коду програми!

Крім того, що в C ++ передбачено оголошення своїх функцій, також можна, можливо скористатися функціями певними в стандартних заголовних файлах мовипрограмування C ++. щоб скористатися функцією, певної в заголовному файлі, потрібно його підключити. Наприклад, щоб скористатися функцією, яка зводитьдеякий число в ступінь, потрібно підключити заголовки файл `<cmath>` і в запусити функцію `pow ()` в тілі програми. розробимо програму, в якій запусимофункцію `pow ()`.

`/ Inc_func.cpp: визначає точку входу для консольного застосування.`

```
#include "stdafx.h"
// дію 1 - підключаємо заголовки <cmath> який містить прототипи основних
математичних функцій
#include <cmath>

int main (int argc, char * argv [])
{
    float power = pow (3.14,2); // дію 2 - запуск функції зведення числа в
ступінь
    return 0;
}
```

підключення заголовних файлів виконується так, як показано в рядку 5, т. е. оголошується препроцесорну директива `#include`, після чого всередині знаків `<>` пишеться ім'я заголовки файлу. коли підключений заголовки файл, можна використовувати функцію, що, і зроблено в рядку 9. функція `pow ()` зводить число 3.14 в квадрат і привласнює отриманий результат змінної `power`, де `pow` - ім'я функції; числа 3.14 і 2 - аргументи функції;

завжди після імені функції ставляться круглі скобочки, всередині яких, функції передаються аргументи, і якщо аргументів кілька, то вони відокремлюються один віддруга комами. аргументи потрібні для того, щоб функції передати інформацію. Наприклад, щоб звести число 3.14 в квадрат використовуючи функцію `pow ()`, потрібно як - то цієї функції повідомити, яке число, і в яку ступінь його зводити. ось саме для цього і придумані аргументи функцій, але бувають функції, в яких аргументи НЕ передаються, такі функції викликаються з порожніми круглими скобочки. Отже, для того, щоб скористатися функцією з стандартного заголовки файлу C ++ необхідно виконати два дії:

1. підключити необхідний заголовки файл;

2. запустити потрібну функцію.

Крім виклику функцій з стандартних заголовних файлів, в мовою програмування C++ передбачена можливість створення власних функцій. В мовою програмування C++ є два типу функцій:

1. Функції, які НЕ повертають значень
2. Функції, які повертають значення
3. Функції, які не повертають значення, завершивши свою роботу, ніякого відповіді програмі НЕ дають. Розглянемо структуру оголошення таких функцій.

```
// Структура оголошення функцій не повертають значень
void / * ім'я функції * / ( / * параметри функції * / ) // Заголовок функції
{
// Тіло функції
}
```

сроки 2 починається з зарезервованого слова `void` - це тип даних, який НЕ може зберігати будь - які дані. Тип даних `void` говорить про тому, що дана функція НЕ повертає ніяких значень. `void` ніяк по - іншому НЕ використовується і потрібен тільки для того, щоб компілятор міг визначити тип функції. після зарезервованого слова `void` пишеться ім'я функції. відразу за ім'ям функції ставляться дві круглі скобочки, що відкривається і закривається. якщо потрібно функції передавати які - то дані, то всередині круглих скобочек оголошуються параметри функції, вони відокремлюються один від друга комами. Рядок 2 називається заголовком функції. після заголовка функції пишуться дві фігурні скобочки, всередині яких знаходиться алгоритм, званий тілом функції. розробимо програму, в якій оголосимо функцію знаходження факторіала, причому функція НЕ повинна повертати значення.

```
// Struct_func.cpp: визначає точку входу для консольного застосування.

#include "stdafx.h"
#include <iostream>
using namespace std;

// Оголошення функції знаходження n!
void faktorial (int numb) // заголовок функції
{
    int rezult = 1; // Инициализируем змінну rezult значенням 1
    for (int i = 1; i <= numb; i++) // цикл обчислення значення n!
        rezult * = i; // Накопичуємо твір в змінної rezult
```

```

        cout << numb << " ! =" << Rezult << endl;    // Друк значення n!
    }

int main (int argc,  char * argv [])
{
    int digit;    // Змінна для зберігання значення n!
    cout << "Enter number:";
    cin >> digit;
    faktorial (digit); // запуск функції знаходження факторіала
    system ( "pause");
    return 0;
}

```

після того, як були підключені Усе необхідні заголовні файли, можна оголошувати функцію знаходження факторіала. під оголошенням функції мається на увазі вибір імені функції, визначення параметрів функції і написання алгоритму, який є тілом функції. після виконання цих дій функцію можна, можливо використовувати в програмі. так як функція НЕ повинна повертати значення, то тип повертаються даних повинен бути void. ім'я функції - faktorial, всередині круглих скобочкооголошена змінна numb типу int. ця змінна є параметром функції faktorial (). Таким чином, всі оголошення в рядку 8 в сукупності складають Заголовок функції. Рядки 9 - 14 складають тіло функції faktorial (). усередині тіла в рядку 10 оголошена змінна rezult, яка буде зберігати результат знаходження n! після чого, в рядках 11-12 оголошено оператор циклу for для знаходження факторіала. В рядку 13 оголошений оператор cout, з допомогою якого значення факторіала буде друкуватися на екрані. Тепер, коли функція оголошена можна, можливо скористатися нею. В рядку 21 запускається функція faktorial (digit), всередині скобочек функції передається аргумент, т. е. значення, що міститься в змінній digit. результат роботи програми

Самостійна робота

1. Створити функції знаходження модулю числа та піднесення у ступінь. Розрахувати математичне рівняння, використовуючи ці функції.
2. Створити функції сортування та знаходження мінімуму та максимуму. Проаналізувати довільний масив.
3. Створити програму – імітацію "однорукого бандиту";
4. Створити програму, що буде тестувати знання школярів таблиці множення.
5. Написати програму, яка відображає у лівого краю екрана порожній трикутник з зірочок, висота якого вказана цілим параметром side. Наприклад, якщо side = 4 на екран має виводитися

```
      *  
    * *  
  *   *  
*****
```

6. Гра «кубики». Умова: є два гральних кубика зі значеннями від 1 до 6. Гра відбувається з комп'ютером, кубики кидаються по черзі. Перемагає той, у кого сума випали очок за підсумками п'яти кидків більше. Передбачте можливість отримання першого ходу людиною або комп'ютером. Кубики відображаються за допомогою символів. В кінці гри необхідно виводити середню суму за кидками для обох учасників.
7. Написати функцію, яка отримує в якості аргументів ціле позитивне число і систему числення, в яку це число повинне перекладається і повертає значення.

Вимоги до оформлення звіту

Звіт повинен містити:

- Короткий конспект теоретичних відомостей;
- Результати виконаних дій.