



实验五 C++程序的结构

清华大学



实验目的

- 观察程序运行中变量的作用域、生存期和可见性；
- 学习类的静态成员的使用；
- 学习多文件结构在C++程序中的使用；





实验任务（1）

- 1.运行下面的程序，观察变量x、y的值
 - 程序见实验书





实验任务（2）

- 2.实现客户机（CLIENT）类。声明字符型静态数据成员ServerName，保存其服务器名称；整型静态数据成员ClientNum，记录已定义的客户数量；定义静态函数ChangeServerName()改变服务器名称。在头文件client.h中声明类，在文件client.cpp中实现，在文件test.cpp中测试这个类，观察相应的成员变量取值的变化情况。





实验提示（1）

- 运行lab5_1程序，观察程序输出。全局变量的作用域为文件作用域，在整个程序运行期间有效，但如果在局部模块中声明了同名的变量，则在局部模块中，可见的是局部变量，此时，全局变量不可见；而局部变量的生存期只限于相应的程序模块中，离开相应的程序模块，局部变量x、y就不再存在，此时同名的全局变量重新可见。

```
Begin...  
x = 1  
y = 2  
Evaluate x and y in main()...  
x = 10  
y = 20  
Step into fn1()...  
x = 1  
y = 200  
Back in main  
x = 10  
y = 20
```





实验提示 (1)

- 尝试使用单步调试观察

```
Begin...  
x = 1  
y = 2  
Evaluate x and y in main()...  
x = 10  
y = 20  
Step into fn1()...  
x = 1  
y = 200  
Back in main  
x = 10  
y = 20
```





实验提示（2）

- 新建一个空的项目lab5_2，添加头文件client.h，在其中声明类CLIENT，注意使用编译预处理命令；再添加源程序文件client.cpp，在其中实现CLIENT类，注意静态成员变量的使用方法；再添加文件lab5_2.cpp，在其中定义main()函数，测试CLIENT类，观察相应的成员变量取值的变化情况。





实验提示（2）

- 注意编译预处理命令
- 多种静态函数的调用方式
- 静态变量的初始化方式
- .h中声明
- .cpp实现
- 静态变量的存活

```
//  
  
#include "client.h"  
  
char client::ServerName = '1';  
int client::ClientNum = 0;  
void client::ChangeServerName(char x){  
    ServerName = x;  
}
```

```
CMakeLists.txt x main.cpp x client.cpp x client.h x  
//  
// Created by Xiang Yu on 2020/3/15.  
//  
  
#ifndef LAB5_CLIENT_H  
#define LAB5_CLIENT_H  
  
class client {  
public:  
    static char ServerName;  
    static int ClientNum;  
    static void ChangeServerName(char x);  
};  
  
#endif //LAB5_CLIENT_H
```





实验演示



谢谢大家！

Q&A