



燕山大学
YANSHAN UNIVERSITY

C++面向对象程序设计 实验指导书

(4) 数据共享与保护

燕山大学软件工程系

目 录

实验 4 数据共享与保护	1
1.1 时间安排	1
1.2 实验目的和要求	1
1.3 实验报告的撰写要求	1
1.4 实验内容	1

实验 4 数据共享与保护

1.1 时间安排

本实验安排 2 个实验课时。

1.2 实验目的和要求

1. 观察程序运行中变量的作用域、生存期。
2. 学习类的静态成员的使用。
3. 理解类的友元函数、友元类。
4. 学习多文件结构在 C++ 程序中的使用。

1.3 实验报告的撰写要求

将实验内容中红色字体的题目的构思过程、源码、运行结果（截图）、心得体会等内容按要求填写，详见实验报告模板。

1.4 实验内容

(1) 运行下面的程序，观察变量 `x`、`y` 的值，并体会标识符的作用域、同名隐藏规则等。

```
#include <iostream>
using namespace std;
void fn1(); //函数原型声明
int x = 1, y = 2;
int main()
{
    cout << "Begin..." << endl;
    cout << "x = " << x << endl;
    cout << "y = " << y << endl;
    cout << "Evaluate x and y in main()..." << endl;
    int x = 10, y = 20;
    cout << "x = " << x << endl;
    cout << "y = " << y << endl;
    cout << "Step into fn1()..." << endl;
    fn1(); //函数调用
    cout << "Back in main" << endl;
    cout << "x = " << x << endl;
    cout << "y = " << y << endl;
    return 0; }
void fn1()
{
    int y = 200;
    cout << "x = " << x << endl;
    cout << "y = " << y << endl; }
```



(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；声明整型静态数据成员 **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。

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。

体会静态成员的定义方法、工作机理及其适用场合。

```
////////////////////////////////client.h////////////////////////////////
#include <iostream>
using namespace std;

class Client
{
public:
    Client();
    ~Client();
    static int GetClientNum();    //获得客户机数量
    static void ChangeServerName(char sn); //设置服务器名
    static char GetServerName();    //获得服务器名
    friend void PrintServerNameAndClientNum(); //屏幕输出服务器名和客户机数目
protected:
    static char ServerName; //服务器名
    static int ClientNum;    //客户机数量
};

#endif

////////////////////////////////Client.cpp////////////////////////////////
#include "client.h"
#include <iostream>
using namespace std;

Client::Client()
{
    ClientNum++;
}

Client::~Client()
{
    ClientNum--;
}

int Client::GetClientNum() //获得客户机数量 {
return ClientNum;
}

int Client::ClientNum = 0 ;
char Client::ServerName = 'N';
```



```
void Client::ChangeServerName(char sn) //设置服务器名
{
    ServerName=sn;
}

char Client::GetServerName() // 获得服务器名
{
    return ServerName ;
}

void PrintServerNameAndClientNum()
{
    cout<<"服务器名: "<<Client::ServerName<<" "
        <<"客户机数: "<<Client::ClientNum<<endl;
}

////////////////////////////////////////test.cpp////////////////////////////////////////
#include <iostream>
#include "client.h"
using namespace std;
void PrintServerNameAndClientNum();
void main ()
{
    Client C1;
    Client C2;
    cout<<"服务器名称为: "<<C1.GetServerName()<<endl;
    cout<<"客户机数量为: "<<C1.GetClientNum()<<endl;
    C1.ChangeServerName('A');
    cout<<"服务器名为: " << C1.GetServerName()<<endl;
    PrintServerNameAndClientNum();
}
```

(3) 编写一类 **Cpoint** 表示一个点的信息。在此基础上编写一个表示三角形的类 **tria**，顶点为其对象成员。编写完整的程序输出三角形的面积。

要求用用友元函数来实现三角形面积的输出。分析：友元函数的使用步骤：

1.先声明使用友元的类； 2.定义友元函数所属的类； 3. 友员成员函数必须现在其所属的类里声明；但不能定义函数体；

4.定义使用友元函数的类的类体； 5.在两个类之外定义友元函数函数体

注意：类类型的参数的传递；体会友元函数、常引用

```
#include <iostream>
#include <cmath>
using namespace std;
```

```

class Cpoint;
////////// tria //////////
class tria
{
    public:
        void set(Cpoint &a,Cpoint &b,Cpoint &c);
        void show(const Cpoint &a, const Cpoint &b, const Cpoint &c);
        double area()
        {
            double p=(s+t+q)/2;
            return sqrt(p*(p-s)*(p-t)*(p-q));  }
    private:
        double s,t,q,p;
};
////////// Cpoint //////////
class Cpoint
{
    public:
        Cpoint(float i,float j) {x=i; y=j; }
        friend void tria::set(Cpoint &a,Cpoint &b,Cpoint &c);
        friend void tria::show(const Cpoint &a, const Cpoint &b,const Cpoint &c);
    private:
        float x,y;
};

void tria::set(Cpoint &a,Cpoint &b,Cpoint &c)
{
    s=sqrt((a.x-b.x)*(a.x-b.x)+(a.y-b.y)*(a.y-b.y));
    t= sqrt((c.x-b.x)*(c.x-b.x)+(c.y-b.y)*(c.y-b.y));
    q=sqrt((a.x-c.x)*(a.x-c.x)+(a.y-c.y)*(a.y-c.y));
}
void tria::show(const Cpoint &a,const Cpoint &b,const Cpoint &c)
{
    cout<<"a的坐标为: ("<<a.x<<","<<a.y<<)"<<endl;
    cout<<"b的坐标为: ("<<b.x<<","<<b.y<<)"<<endl;
    cout<<"c 的坐标为: ("<<c.x<<","<<c.y<<)"<<endl;
    cout<<"三个边长为: "<<endl;
    cout<<s<<"\t"<<t<<"\t"<<q<<endl;
}
////////// main //////////
void main()
{
    float x1,y1,x2,y2,x3,y3;
    cout<<"?"<<endl;

```



```
cin>>x1>>y1>>x2>>y2>>x3>>y3;
Cpoint a(x1,y1),b(x2,y2),c(x3,y3);
tria tr;
tr.set( a,b, c);
tr.show(a,b,c);
cout<<"三角形的面积为"<<tr.area()<<endl;
}
```

(4) 定义一个 **Girl** 类和一个 **Boy** 类，这两个类中都有表示姓名、年龄的私有成员变量，都要定义构造函数、析构函数、输出成员变量信息的公有成员函数。

1. 根据要求定义相应的类；

2. 将 **Girl** 类作为 **Boy** 类的友元类，在 **Girl** 类的成员函数 **visitboy(boy &)** 中访问 **boy** 类的私有成员，观察程序运行结果；

3. 在 **boy** 类的某成员函数 **visitgirl(girl &)** 中试图访问 **girl** 类的私有成员，观察编译器给出的错误信息，理解友元的不可逆性；

4. 主函数中正确定义两个类的对象，调用各自的成员函数实现相应的功能；

5. 再将 **boy** 类作为 **girl** 类的友元类，在 **boy** 类的某成员函数 **visistgirl(girl &)** 中访问 **girl** 类的私有成员，观察编译器给出的信息；

6. 删除两个类中的函数 **visitgirl(girl &)**, **visitboy(boy &)**，定义一个顶层函数 **visitboygirl(boy &, girl &)**，作为以上两个类的友元，通过调用该函数输出男孩和女孩的信息。