

# 实验三 类与对象（三）——静态成员、常量成员、友元、对象成员

张林鹏\_2021032449

## 一、实验目的

1. 理解静态数据成员、静态成员函数的作用, 熟悉其应用;
2. 理解常量对象、常量数据成员、常量成员函数作用, 熟悉相互关系及应用;
3. 熟悉友元函数、友元类的定义及应用;
4. 熟悉对象成员 (容器类) 的应用;
5. 进一步熟悉类与对象的应用及编程.

## 二、实验内容

### 程序1: exp\_301.cpp

1. 运行该程序的输出结果为:
2. 由输出结果可知, 静态成员属于类, 在类中只有一份拷贝.
3. 程序中的 `int counter::count=100;` 的作用是: 给静态数据成员count赋初值为100

如果将其放在 `main( )` 函数中, 重新编译程序, 会出现 编译错误, 其原因是 静态成员的初值必须在类外部定义时给出.

### 程序2: exp\_302.cpp

4. 编译程序, 会出现 编译错误, 其出错原因是 头文件 `hhpoint.h` 和 `point.h` 中的类定义重复.
5. 将程序中的 (1) 改为注释, 再将 (2) 行的注释去掉, 再编译运行程序, 程序的输出结果为:

```
Before moving: p: x=20 y=40
center: x=100 y=100
```

6. 将程序中 (3) ~ (6) 行首的注释去掉, 程序的输出结果为:

```
Before moving: p: x=20 y=40
```

```
center: x=100 y=100
After moving: p: x=30 y=60
```

7. 将程序中 (7) 行首的注释去掉, 编译程序时无错误, 而运行时会出现错误, 其原因是 center是一个const对象, 它的成员函数move()没有被声明为const成员函数, 因此不能被const对象调用.

### 程序3: exp\_303.cpp

8. 编译程序出错的原因是 girl 类中的 name 指针未初始化.
9. 在 girl 类中加上 `friend void disp(girl &g);` 重新编译运行程序, 其输出结果为:

```
姓名:李小丫 年龄:12
姓名:王永兰 年龄:15
姓名:赵梦美 年龄:13
```

其中, `friend void disp(girl &g);` 表示函数 `disp( )` 为girl类的 友元函数.

### 程序4: exp\_304.cpp

1. 你的分析输出结果是:

```
(1): 15 (2): 30
```

2. 程序实际输出结果是:

```
(1): 15 (2): 30
```

### 程序5: exp\_305.cpp

1. 你的分析输出结果是:

```
女孩姓名:李小丫 年龄:12
男孩姓名:张海兵 年龄:15
```

2. 程序实际输出结果是:

```
女孩姓名:李小丫 年龄:12
男孩姓名:张海兵 年龄:15
```

## 程序6:

### 1. 你的分析输出结果是:

姓名:张小三 性别:男 出生日期:1985年12月15日  
姓名:李小丫 性别:女 出生日期:1986年3月9日

### 2. 程序实际输出结果是:

姓名:张小三 性别:男 出生日期:1985年12月15日  
姓名:李小丫 性别:女 出生日期:1986年3月9日

## 程序7: exp\_307.cpp

- 补全后的代买:

```
#include <iostream>
#include <string.h>
using namespace std;
class boy; // 类boy的声明;
class girl
{
private:
    char *name;
    int age;

public:
    girl(char *na, int n)
    {
        name = new char[strlen(na) + 1];
        strcpy(name, na);
        age = n;
    }
    ~girl(void)
    {
        delete name;
    } // 释放new分配的内存
    void disp(boy &b);
};
class boy
{
private:
    char *name;
    int age;
    friend class girl; // 声明girl是boy的友元类
public:
```

```

boy(char *na, int n)
{
    name = new char[strlen(na) + 1];
    strcpy(name, na);
    age = n;
}
~boy(void)
{
    delete name;
} // 释放new分配的内存
};
void girl::disp(boy &b)
{
    cout << "女孩姓名:" << name;
    cout << "  年龄:" << age << endl;
    cout << "男孩姓名:" << b.name; // b为boy类的一个成员
    cout << "  年龄:" << b.age << endl;
}
int main(void)
{
    girl g1("李小丫", 12);
    boy b1("张海兵", 15);
    g1.disp(b1);

    return 0;
}

```

## 程序设计实验:

- exp\_308.cpp

```

#include <bits/stdc++.h>

class point
{
public:
    double x, y;
    point(double x, double y) : x(x), y(y) {}
    point() {}
    friend class distance;
};

class distance
{
public:
    double dis(point a, point b)
    {
        return sqrt((a.x - b.x) * (a.x - b.x) + (a.y - b.y) * (a.y - b.y));
    }
};

int main()

```

```
{  
    point a(1, 2), b(3, 4);  
    distance d;  
    std::cout << d.dis(a, b) << std::endl;  
  
    return 0;  
}
```