

## 实验 1 类和对象

班级 信安 20-2    学号 20101110201    姓名 李天昊

### [实验目的]

- 1、掌握声明类的方法，类和类的成员的概念以及定义对象的方法。
- 2、初步掌握用类和对象编制基于对象的程序。
- 3、学习检查和调试基于对象的程序。

### [实验要求]

给出以下各实验内容的源程序代码，并把编译、运行过程中出现的问题以及解决方法填入实验报告中，按时上交。

[实验学时] 2 学时。

### [实验内容]

#### 1、有以下程序：

```
#include <iostream>
using namespace std;
class Time                      //定义 Time 类
{ public:                      //数据成员为公用的
    int hour;
    int minute;
    int sec;
};

int main()
{ Time t1;                    //定义 t1 为 Time 类对象
  cin>> t1.hour;              //输入设定的时间
  cin>> t1.minute;
  cin>> t1.sec;
  cout<<t1.hour<<“:”<<t1.minute<<“:”<<t1.sec<<endl;
  return 0;
}
```

改写程序，要求：

- ①将数据成员改为私有的。
- ②将输入和输出的功能改为由成员函数实现，并在类体内定义成员函数。

- expl-1.cpp

```
#include <iostream>

using namespace std;

class CTime {
private:
    int m_hour;
    int m_minute;
    int m_second;
public:
    CTime(int h = 0, int m = 0, int s = 0);
    CTime(const CTime &t);
    ~CTime();

    void set_time(int h, int m, int s) {
        m_hour = h;
        m_minute = m;
        m_second = s;
    }
    inline void show_time() {
        cout << m_hour << ":" << m_minute << ":" << m_second << endl;
    }
};

CTime::CTime(int h, int m, int s): m_hour(h), m_minute(m), m_second(s) { }
CTime::CTime(const CTime &t): m_hour(t.m_hour), m_minute(t.m_minute), m_second(t.m_second) { }
CTime::~CTime() { }

int main() {

    CTime t1;
    CTime t2(16, 43, 21);
```

```
t1.show_time();
t2.show_time();
t1.set_time(17, 44, 22);
t1.show_time();
t2.show_time();

return 0;
}
```

- 编译成功，测试通过：

```
david@ThinkpadT480s:/mnt/d/#Programming/CPP/NCUT/EXP$ g++ expl-1.cpp -o main
david@ThinkpadT480s:/mnt/d/#Programming/CPP/NCUT/EXP$ ./main
0:0:0
16:43:21
17:44:22
16:43:21
david@ThinkpadT480s:/mnt/d/#Programming/CPP/NCUT/EXP$ |
```

2、编写一个基于对象的程序：输入半径，计算圆的周长和面积并输出。

实现提示：

- ①数据成员包括圆的半径 radius;
- ②成员函数包括计算周长: length(), 计算面积: area()。
- ③main()函数中以半径值 10.0 创建一个对象，分别输出该对象的周长和面积。

- expl-2.cpp

```
#include <iostream>

using namespace std;

class CCircle {
private:
    double m_radius;
    const double PI = 3.14;
public:
    CCircle(double r);
    CCircle(const CCircle &c);
    ~CCircle();
    double length();
    double area();
};
```

```
};

CCircle::CCircle(double r) {
    if (r < 0) throw "Invalid Radius Value. ";
    m_radius = r;
}
CCircle::CCircle(const CCircle &c): m_radius(c.m_radius) { }
CCircle::~~CCircle() { }
inline double CCircle::length() {
    return 2 * PI * m_radius;
}
inline double CCircle::area() {
    return PI * m_radius * m_radius;
}

int main() {
    CCircle c(10.0);
    cout << "circumference: " << c.length() << endl;
    cout << "area: " << c.area() << endl;
    return 0;
}
```

- 编译成功，测试通过：

```
david@ThinkpadT480s:/mnt/d/#Programming/CPP/NCUT/EXP$ g++ exp1-2.cpp -o main
david@ThinkpadT480s:/mnt/d/#Programming/CPP/NCUT/EXP$ ./main
circumference: 62.8
area: 314
david@ThinkpadT480s:/mnt/d/#Programming/CPP/NCUT/EXP$ |
```

3、需要求 3 个长方柱的体积，请编写一个基于对象的程序。数据成员包括 length(长)、width(宽)、height(高)。要求用成员函数实现以下功能：

- ①由键盘分别输入 3 个长方柱的长、宽、高；
- ②计算长方柱的体积；
- ③输出 3 个长方柱的体积。

- exp1-3.cpp

```
#include <iostream>
#include <cmath>

using namespace std;
```

```
// =====  
===== //  
//  
// CLASS START  
//  
// =====  
===== //  
  
class CCube {  
private:  
    double m_length;  
    double m_width;  
    double m_height;  
    double m_volume;  
    static size_t sm_nTotalVolume;  
    static size_t sm_nTotalNumber;  
public:  
    CCube();  
    CCube(double l, double w, double h);  
    CCube(const CCube &r);  
    ~CCube();  
  
    void set_value(istream &is = cin);  
    inline double get_volume();  
  
    inline static void PrintTotal();  
};  
  
size_t CCube::sm_nTotalVolume = 0;  
size_t CCube::sm_nTotalNumber = 0;  
  
CCube::CCube() {  
    m_volume = -1;  
}  
  
CCube::CCube(double l, double w, double h) {  
    if (l < 0) throw "Invalid Length Value";  
    if (w < 0) throw "Invalid Width Value";  
    if (h < 0) throw "Invalid Height Value";  
}
```

```
m_length = 1;
m_width = w;
m_height = h;
m_volume = m_length * m_width * m_height;
++sm_nTotalNumber;
sm_nTotalVolume += m_volume;
}

CCube::CCube(const CCube &r): m_length(r.m_length), m_width(r.m_width), m_height(r.m_height) { }

CCube::~~CCube() {
    --sm_nTotalNumber;
    sm_nTotalVolume -= m_volume;
}

void CCube::set_value(istream &is) {
    // 初次设置长宽高
    if (m_volume == -1) {
        ++sm_nTotalNumber;
        m_volume = 0;
    }
    // 从总体积中减去旧体积值
    sm_nTotalVolume -= m_volume;
    cout << "> set length, width and height: ";
    is >> m_length;
    is >> m_width;
    is >> m_height;
    // 计算并累加新体积值
    m_volume = m_length * m_width * m_height;
    sm_nTotalVolume += m_volume;
}

inline double CCube::get_volume() {
    return m_volume;
}

inline void CCube::PrintTotal() {
```

```
        cout << "The volume of " << sm_nTotalNumber << " cubes is " <<
sm_nTotalVolume << "." << endl;
    }

// =====
//                                     CLASS END
//                                     //
// =====
//                                     //

int main() {

    // 定义 CCube 类数组并打印初始数量和体积
    CCube c[3];
    CCube::PrintTotal();
    cout << endl;

    for (size_t i = 0; i != 3; ++i) {
        // 输入长宽高并打印体积
        c[i].set_value();
        cout << "> " << c[i].get_volume() << endl;
        // 打印累计数量和体积
        CCube::PrintTotal();
        cout << endl;
    }
    return 0;
}
```

- 编译成功，测试通过：

```
david@ThinkpadT480s:/mnt/d/#Programming/CPP/NCUT/EXP$ g++ expl-3.cpp -o main
david@ThinkpadT480s:/mnt/d/#Programming/CPP/NCUT/EXP$ ./main
The volume of 0 cubes is 0.

> set length, width and height: 1 1 1
> 1
The volume of 1 cubes is 1.

> set length, width and height: 2 2 2
> 8
The volume of 2 cubes is 9.

> set length, width and height: 3 3 3
> 27
The volume of 3 cubes is 36.
```

### [实验总结]

给出对本次实验的总结。

- 1、学习并掌握了声明 C++类的方法，理解了类和类的成员的概念和定义对象的方法；
- 2、尝试了用类和对象编制基于对象的 C++程序；
- 3、学习调试和检查基于对象的 C++程序。