

Assignment 6

191220022 丁一凡

一、概念简答题

1. 在C++中，protected类成员访问控制的作用是什么？

缓解继承与数据封装的矛盾：在基类中声明为 `protected` 的成员可以被派生类使用，但不能被基类的实例用户使用

2. 请简述派生类对象的初始化和析构顺序，并简述理由，为什么需要按照这个顺序？

在创建派生类的对象时，派生类的构造函数在进入其函数体前会先调用基类的构造函数，然后再执行自己的函数体。

在对象消亡时，先调用自己的析构函数，最后调用基类的析构函数。

派生类是继承自基类的，保有基类的所有特征，所以应该先调用基类的构造函数，再在此基础上添加派生类新的特征，析构函数同理。

二、代码编程题

1.

`Square` 类用 `public` 方式继承了 `Rectangle` 类，这样 `set_width()` 以及 `set_height()` 接口对于 `Square` 的实例用户也是 `public` 类型的，这样用户就可以通过这两个接口修改数据，使得图形不符合正方形的定义。

应该考虑用 `protected` 或者 `private` 方式继承

2.

```
enum TimeZone { w12=-12,w11,w10,w9,w8,w7,w6,w5,w4,w3,w2,w1,
GMT,E1,E2,E3,E4,E5,E6,E7,E8,E9,E10,E11,E12};

class ExtTime: protected Time
{
    TimeZone timezone;
public:
    ExtTime():Time(0,0,0)
    {
        timezone = GMT;
    }
}
```

```

}
ExtTime(int h, int m, int s, TimeZone t):Time(h,m,s)
{
    timezone = t;
}
void set(int h, int m, int s, TimeZone t)
{
    Time::set(h, m, s);
    timezone = t;
}
void display()
{
    string t;
    switch(timezone)
    {
        case w12:t = "W12"; break;
        case w11:t="W11";break;
        case w10:t="W10";break;
        case w9:t="W9";break;
        case w8:t="W8";break;
        case w7:t="W7";break;
        case w6:t="W6";break;
        case w5:t="W5";break;
        case w4:t="W4";break;
        case w3:t="W3";break;
        case w2:t="W2";break;
        case w1:t="W1";break;
        case GMT:t="GMT";break;
        case E12:t = "E12"; break;
        case E11:t="E11";break;
        case E10:t="E10";break;
        case E9:t="E9";break;
        case E8:t="E8";break;
        case E7:t="E7";break;
        case E6:t="E6";break;
        case E5:t="E5";break;
        case E4:t="E4";break;
        case E3:t="E3";break;
        case E2:t="E2";break;
        case E1:t="E1";break;
    }
    cout << t << " ";
    Time::display();
}
bool equal(const ExtTime &other_time)
{
    return timezone == other_time.timezone && hour == other_time.hour && min
== other_time.min && sec == other_time.sec;
}
bool less_than(const ExtTime & other_time)
{
    Time t(other_time.hour, other_time.min, other_time.sec);
    return Time::less_than(t);
}
};

```

此题适合用 `public` 继承，当基类和派生类对于基类中数据成员的定义和操作不变时，可以采用 `public` 继承，否则考虑采用 `protected` 或 `private` 继承。