静态类与非静态类一些注意事项：

1. 静态成员与非静态成员的访问方法：

**静态**

1. 静态成员能够出现在静态类或非静态类中，想在其他地方访问这些类中的静态成员我们需要通过 类名.成员名 来访问
2. 静态成员中只允许出现静态成员，如果要想在静态成员中访问非静态的成员，需要进行一次实例化，通过实例名来访问

**非静态**

1. 非静态成员只允许出现在非静态类中，想在其他地方访问这些类中的非静态成员，我们可以需要进行一次类的实例化，然后通过 实例名.成员名 来访问
2. 非静态成员中允许出现非静态成员和静态成员
3. 静态成员储存在内存的静态储存区域中
4. 静态的构造函数不能有访问修饰符（默认private，所以静态类不能创建对象）和方法后面的参数，

它只能在：

1. 创建包含静态构函数的类的对象时
2. 第一次访问包含静态构造造函数的类的静态成员时

才会被调用

1. 静态类不能被继承，不能创建对象，因为被某个类继承和创建自身对象是一个实例化的过程，而静态类里面都是静态成员
2. 静态类能够继承接口
3. 静态成员在调用前就已经初始化，默认值为0；非静态成员需要实例化对象时才初始化
4. 在一个类中，包含着静态成员与非静态成员，当我们实例化这个类时，优先初始化静态成员，且不管创建了多少个对象，静态成员至始至终只会初始化一次；而非静态成员则是我们创建了多少个对象就初始化多少次，并且在静态成员的初始化完成后才会初始化非静态成员
5. 由于静态成员至始至终只会初始化一次，所以静态成员也在本项目中是：**全局变量、资源共享**
6. 因为静态成员是资源共享的，所以在一个程序中，静态成员的定义应当尽量减少，因为静态成员只会在程序结束的时候才可以释放资源，因为它的资源是共享的