单例模式

单例模式是一种常用的软件设计模式。在它的核心结构中只包含一个被称为单例的特殊类。通过单例模式可以保证系统中一个类只有一个实例而且该实例易于外界访问，从而方便对实例个数的控制并节约系统资源。如果希望在系统中某个类的对象只能存在一个，单例模式是最好的解决方案。

联想到静态变量

单例模式要做的就是仅仅创建一个类对象，且只能有这一个类对象。

如何做到？

将默认构造函数和拷贝构造函数设置为private,因为不能随意调用构造函数创建类对象。再设置private里的static成员变量为类指针指向类对象，在类外进行定义这个static成员变量，即定义一个独立存在的类对象。由于这个类对象是由static成员变量定义的，因此它在编译阶段就已经形成了。

同时，也要为这个类对象设置访问私有数据的接口，注意，通过static成员函数getInstance来获取static成员属性，因为如果用普通成员函数，则必须用类对象来调用，很明显，这个类无法构造普通的类对象，都是通过类名直接访问的。

其实总的来看就是，要创建单例模式的类，就要将它的构造函数都禁止掉，不能用普通创建类对象的方式创建多余的类对象，同时设置static成员变量指向唯一的类对象，设置static成员函数接口getInstance，只对类开放，这样就没有创建多余类对象的风险，private成员变量也不会被修改。

创建一个打印机单例模式代码如下：