单例模式

核心作用是确保一个类只有一个实例，并且提供一个该实例的全局访问点

【测试】单例模式

```
# 测试单例模式
class MySingleton:
    __flag = None
    __status = True
    def __new__(cls, *args, **kwargs):
        if cls.__flag is None:
            cls.__flag = object.__new__(cls)
        else:
            cls.__status = False
        return cls.__flag
    def __init__(self, name):
        if self.__status:
            print("init")
            self.name = name

a = MySingleton("none")
b = MySingleton("123")
print(a)
print(b.name)
print(a is b)
```

【练习】工厂模式单例模式连用

```
# 测试工厂模式和单例模式何合并使用

class Factory:
    def create(self, brand):
        self.brand = brand
        if self.brand == 1:
            return Class1()
class Class1:
    __flag = None
    __status = True
    def __new__(cls, *args, **kwargs):
        if cls.__flag is None:
            cls.__flag = object.__new__(cls)
        else:
            __status = False
        return cls.__flag
    def __init__(self):
```

```
        if self.__status:
            self.value = 1
            print(self.value)

f = Factory().create(1)
```