设计原则评估

1. 模块化：我们设置的道具模块与技能模块在耦合度上属于松散的耦合，同时每个模块的内聚度较高，可以实现总体耦合度较低。
2. 接口：

道具模块：

目标：实现道具功能

前置条件：基础操作正常运行

协议：按设计的道具功能运行

后置条件：道具模块内容的改变以及基础操作的进行

质量属性：保证项目的正常运作

技能模块：

目标：实现技能功能

前置条件：基础操作正常运行

协议：按设计的技能功能运行

后置条件：技能模块内容的改变以及基础操作的进行

质量属性：保证项目的正常运作

1. 信息隐藏：每一个模块依据其耦合度来实现信息隐藏的程度。其中两个模块部分间的元素彼此关联较少，有较好的信息隐藏效果。
2. 增量式开发：我们在开发时利用了使用图来进行了增量式的开发设计。
3. 抽象:每一个模块的功能实现基于抽象的设计，在关注一个模块中的技术细节时，基于框架忽略另一个模块中的技术细节。
4. 通用性：因为进阶操作中的两个模块相对独立所以不具有很高的通用性。

面向对象评估设计：

1. 术语：根据对象设计了对象的类和接口，即技能类和道具类
2. 继承与对象组合：我们使用组合的方式来实现，以在运行时方便的实现对构件对象的替换。
3. 可替换性：子类支持父类的所有方法，并且他们的签名是兼容的，子类方法的参数和返回与父类相同。子类方法满足父类方法的规格说明。子类中保留了父类中声明的所有性质。我们进阶操作是基于基础操作展开的，所以进阶操作支持基础操作的所有方法。