

```
# 继续优化, 根据 NPC 类型来设定 blood 和 weapon
# 将代码尽量集中, 是有好处的

# NPC 类
class NPC(object):

    # 初始化 NPC 的属性
    def __init__(self, name):
        self.name = name

        self.npc_type = type(self).__name__

        print('')
        print(self.npc_type, 'NPC created!')

        if self.npc_type == 'Soldier':
            self.weapon = 'sword'
            self.blood = 1000

        if self.npc_type == 'Wizard':
            self.weapon = 'staff'
            self.blood = 2000

    # 定义方法 - 显示 NPC 属性
    def show_properties(self):
        print('name:', self.name)
        print('weapon:', self.weapon)
        print('blood:', self.blood)

    # 定义方法 - 通用攻击
    def fight_common(self):
        print('Fight Common')

# 战士 Soldier 类
class Soldier(NPC):

    # 建立 soldier 的初始化
    def __init__(self, name):
        # 调用 父类 NPC 的初始化方法
        NPC.__init__(self, name)

        # soldier 自己的初始化
        self.soldier_level = 1

    # 定义方法 - 显示 NPC 属性
    def show_properties(self):
        # 通过 super 来调用父类方法
        super(Soldier, self).show_properties()
        print('soldier_level', self.soldier_level)
```

```
# 巫师 Wizard 类
class Wizard(NPC):

    # 建立 Wizard 的初始化
    def __init__(self, name):
        # 调用 父类 NPC 的初始化方法
        NPC.__init__(self, name)

        # soldier 自己的初始化
        self.wizard_level = 1

    # 定义方法 - 显示 NPC 属性
    def show_properties(self):
        # 通过 super 来调用父类方法
        super(Wizard, self).show_properties()
        print('wizard_level', self.wizard_level)

    # 定义一个巫师专用的战斗方法
    def fight_magic(self):
        print('Wizard Magic!')

s = []

for i in range(2):
    n = Soldier('AA' + str(i))
    n.show_properties()
    s.append(n)

for i in range(2):
    n = Wizard('CC' + str(i))
    n.show_properties()
    s.append(n)
```