

--初始化一个类

```
Person={name, age, gender, address="China"}
```

--模拟构造方法

```
function Person:New()
    obj={}
    setmetatable(obj, Person)
    Person.__index=Person
    return obj
end
```

--模拟一个方法

```
function Person>Show()
    print(self.name, self.age, self.gender, self.address)
end
```

--实例化对象

```
l1k=Person:New()
l1k.name="Jack"
l1k.age=10
l1k.gender="男"
l1k>Show()
```

```
mk=Person:New()
mk.name="Monkey"
mk.age=1
mk.gender="ss"
mk>Show()
```

---

## 类与对象如何实现

```
Hero = {name, hp, mp, attack}
function Hero:New()
    obj={}
    setmetatable(obj, Hero)
    Hero.__index=Hero
    return obj
```

```
end

function Hero:Attack1()
    print(self.name.. "Attack1")
end

function Hero:Attack2()
    print(self.name.. "Attack2")
end

function Hero:ToString()
    print(string.format("name:%s, hp:%s, mp:%s", self.name, self.hp, self.mp))
end

mk=Hero:New()
mk.name="SS"
mk.hp=100
mk.mp=20
mk:Attack1()
mk:Attack2()
mk:ToString()
```

---

## 继承关系

--创建一个动物类

```
Animal={name}
```

--构造方法

```
function Animal:New(name)
```

```
    local obj={}
```

```
    setmetatable(obj, self)
```

```
    self.__index=self
    self.name=name
    return obj
end
```

--普通方法

```
function Animal:ToString()
    print(self.name.."Animal类中的方法")
```

```
end
```

--子类继承父类

```
Cat=Animal:New(nil)
```

--子类的构造方法

```
function Cat:New(name)
    local obj=Animal:New(name)
    setmetatable(obj, self)
    self.__index=self
    self.name=name
    return obj
```

```
end
```

--子类中的普通方法

```
function Cat:Eat(food)
    print(self.name.."吃:".. food)
```

end

jfm=Cat:New("加菲猫")

jfm:ToString()

jfm:Eat("鱼")

print(jfm.name)