

面向对象作业题 (03)

注意：做题之前请务必讲课上的10个练习题做完。

1. 简述面向对象三大特性；
2. 列举面向对象中所有成员；
3. 用面向对象的嵌套表示：学生、班级、老师的关系，并创建相关对象进行嵌套【根据自己的理解编写】；
4. 看代码写结果；

```
class F3(object):

    def f1(self):
        ret = super().f1()
        print(ret)
        return 123

class F2(object):

    def f1(self):
        print('123')

class F1(F3, F2):
    pass

obj = F1()
obj.f1()
```

5. 看代码写结果；

```
class F1(object):

    def __init__(self, a1):
        self.a1 = a1

    def f2(self, arg):
        print(self.a1, arg)

class F2(F1):

    def f2(self, arg):
        print('666')

obj_list = [F1(1), F2(2), F2(3)]
for obj in obj_list:
    item.f2()
```

6. 看代码写结果;

```
class F1(object):

    def __init__(self,a1):
        self.a1 = a1

    def f2(self,arg):
        print(self.a1,arg)

class F2(F1):

    def f2(self,arg):
        print('666')

obj_list = [F1(1),F2(2),F2(3)]
for obj in obj_list:
    print(item.f2())
```

7. 看代码写结果;

```
class F1(object):

    def __init__(self,num):
        self.num = num

    def func(self,request):
        print(self.num,request)

    def run(self):
        self.func(999)

class F2(StarkConfig):

    def func(self,request):
        print(666,self.num)

objs = [F1(1),F2(2),F2(3)]
objs[1].run()
objs[2].run()
```

8. 看代码写结果;

```
class UserInfo(object):
    pass

class Department(object):
    pass

class StarkConfig(object):
```

```

def __init__(self, num):
    self.num = num

def changelist(self, request):
    print(self.num, request)

def run(self):
    self.changelist(999)

class RoleConfig(StarkConfig):

    def changelist(self, request):
        print(666, self.num)

class AdminSite(object):

    def __init__(self):
        self._registry = {}

    def register(self, k, v):
        self._registry[k] = v(k)

site = AdminSite()
site.register(UserInfo, StarkConfig)
site.register(Department, RoleConfig)
for k, row in site._registry.items():
    row.run()

```

9. 看代码写结果;

```

class UserInfo(object):
    pass

class Department(object):
    pass

class StarkConfig(object):

    def __init__(self, num):
        self.num = num

    def get_vals(self):
        v = [11, 22, 33]

        extra = self.extra_vals()
        if extra:
            v.extend(extra)
        return v

    def extra_vals(self):
        pass

```

```

    def run(self):
        return self.get_vals()

class RoleConfig(StarkConfig):

    def extra_vals(self):
        return [99, 88]

class AdminSite(object):
    def __init__(self):
        self._registry = {}

    def register(self, k, v):
        self._registry[k] = v(k)

site = AdminSite()
site.register(UserInfo, StarkConfig)
site.register(Department, RoleConfig)
for k, row in site._registry.items():
    print(row.run())

```

10. 背写你了解的所有特殊方法并附示例;

11. 看代码写结果;

```

class StarkConfig(object):

    def __init__(self, num):
        self.num = num

    def __call__(self, *args, **kwargs):
        print(self.num)

class RoleConfig(StarkConfig):
    def __call__(self, *args, **kwargs):
        print(self.num)

v1 = StarkConfig(1)
v2 = RoleConfig(11)

v1()
v2()

```

12. 看代码写结果;

```

class StarkConfig(object):

    def __init__(self, num):
        self.num = num

```

```

def run(self):
    self()

def __call__(self, *args, **kwargs):
    print(self.num)

class RoleConfig(StarkConfig):

    def __call__(self, *args, **kwargs):
        print(345)

v1 = StarkConfig(1)
v2 = RoleConfig(11)

print(v1.run())
print(v2.run())

```

13. 看代码写结果。

```

class StarkConfig(object):

    def __init__(self, num):
        self.num = num

    def run(self):
        self()

    def __call__(self, *args, **kwargs):
        print(self.num)

class RoleConfig(StarkConfig):

    def __call__(self, *args, **kwargs):
        print(345)

    def __getitem__(self, item):
        return self.num[item]

v1 = RoleConfig('alex')
v2 = StarkConfig("wupeiqi")
print(v1[1])
print(v2[2])

```