

# 一. 面向对象三大特性

---

- 封装
  - 将属性和方法书写到类的里面的操作即为封装
  - 封装可以为属性和方法添加私有权限
- 继承
  - 子类默认继承父类的所有属性和方法
  - 子类可以重写父类属性和方法
- 多态
  - 传入不同的对象，产生不同的结果

## 二. 多态

---

### 2.1 了解多态

---

多态指的是一类事物有多种形态，（一个抽象类有多个子类，因而多态的概念依赖于继承）。

- 定义：多态是一种使用对象的方式，子类重写父类方法，调用不同子类对象的相同父类方法，可以产生不同的执行结果
- 好处：调用灵活，有了多态，更容易编写出通用的代码，做出通用的编程，以适应需求的不断变化！
- 实现步骤：
  - 定义父类，并提供公共方法
  - 定义子类，并重写父类方法
  - 传递子类对象给调用者，可以看到不同子类执行效果不同

### 2.2 体验多态

---

```
class Dog(object):
    def work(self): # 父类提供统一的方法，哪怕是空方法
        print('指哪打哪...')

class ArmyDog(Dog): # 继承Dog类
    def work(self): # 子类重写父类同名方法
        print('追击敌人...')

class DrugDog(Dog):
    def work(self):
        print('追查毒品...')

class Person(object):
    def work_with_dog(self, dog): # 传入不同的对象，执行不同的代码，即不同的work函数
        dog.work()
```

```
ad = ArmyDog()
dd = DrugDog()

wanzi = Person()
wanzi.work_with_dog(ad)
wanzi.work_with_dog(dd)
```

## 三、学生信息管理系统面向对象封装

### 3、1 目标

- 了解面向对象开发过程中类内部功能的分析方法
- 了解常用系统功能
  - 添加
  - 删除
  - 修改
  - 查询

### 3、2 系统需求

使用面向对象编程思想完成学员管理系统的开发，具体如下：

- 系统要求：学员数据存储在文件中
- 系统功能：添加学员、删除学员、修改学员信息、查询学员信息、显示所有学员信息、保存学员信息及退出系统等功能。

### 3、3 具体实现

#### 3、3、1 学生类

需求：

- 学员信息包含：姓名、语文成绩、数学成绩、英语成绩、总分

代码如下：

```
class Student:
    """学生信息类"""
    def __init__(self, name, chinese, math, english):
        """
        :param name: 姓名
        :param chinese: 语文成绩
        :param math: 数学成绩
        :param english: 英语成绩
        """
        self.name = name
        self.chinese = chinese
        self.math = math
        self.english = english
        self.total = chinese + math + english # 总分
```

## 3、3、2 学生管理系统类

需求：

- 存储数据的位置：文件(students.json)
  - 加载文件数据
  - 修改数据后保存到文件
- 存储数据的形式：列表存储学员对象
- 系统功能
  - 加载学员信息
  - 新建学生信息
  - 显示全部信息
  - 查询学生信息
  - 删除学生信息
  - 修改学生信息
  - 保存学员信息
  - 退出系统

### 3、3、2、1 系统框架

需求：系统功能循环使用，用户输入不同的功能序号执行不同的功能。

- 步骤
  - 定义程序入口函数
    - 加载数据
    - 显示功能菜单
    - 用户输入功能序号
    - 根据用户输入的功能序号执行不同的功能
  - 定义系统功能函数，添加、删除学员等

```
class StudentManager:
    """学生信息管理系统类"""
    def __init__(self):
        # 存储学员数据 -- 列表
        self.student_list = []

    # 1.程序入口函数
    def run(self):
        # 1. 加载文件里面的学员数据
        self.load_student()

        while True:
            # 2. 显示功能菜单
            self.show_menu()
            # 3. 用户输入目标功能序号
            action = input('请选择您想要进行的操作:')

            # 4. 根据用户输入的序号执行不同的功能 -- 如果用户输入1, 执行新建
            if action == '1':
                # 1. 新建学生信息
```

```

        print('1. 新建学生信息')
        self.add_student()

    elif action == '2':
        # 2. 显示全部信息
        print('2. 显示全部信息')
        self.show_student()

    elif action == '3':
        # 3. 查询学生信息
        print('3. 查询学生信息')
        self.search_student()

    elif action == '4':
        # 4. 删除学生信息
        print('4. 删除学生信息')
        self.del_student()

    elif action == '5':
        # 5. 修改学生信息
        print('5. 修改学生信息')
        self.modify_student()

    elif action == '0':
        print('0. 退出系统')
        self.save_student()
        # 退出系统 -- 退出循环
        break

# 2.加载学员信息
def load_student(self):
    pass

# 二. 系统功能函数
# 2.1 显示功能菜单
@staticmethod
def show_menu():
    str_info = """*****
欢迎使用【学生信息管理系统】V1.0
请选择你想要进行的操作
1. 新建学生信息
2. 显示全部信息
3. 查询学生信息
4. 删除学生信息
5. 修改学生信息

0. 退出系统
*****"""

    # 新建学生信息
    def add_student(self):
        pass

    # 显示全部信息
    def show_student(self):

```

```

        pass

# 查询学生信息
def search_student(self):
    pass

# 删除学生信息
def del_student(self):
    pass

# 修改学生信息
def modify_student(self):
    pass

# 保存学员信息
def save_student(self):
    pass

```

### 3、3、2、2 加载学员信息

```

def load_student(self):
    # 1. 打开文件:
    f = open('students.json', mode='r', encoding='utf-8')
    # 2. 读取数据: 文件读取出的数据是字符串还原列表类型; [{}] 转换 [学员对象]
    data = f.read()
    new_list = eval(data)
    self.student_list = [Student(i['name'], i['chinese'], i['math'],
i['english']) for i in new_list]
    # 3. 关闭文件
    f.close()

```

### 3、3、2、3 新建学生信息

```

def add_student(self):
    # 1. 用户输入姓名、性别、手机号
    name = input('请输入您的姓名: ')
    chinese = int(input('请输入您的语文成绩: '))
    math = int(input('请输入您的数学成绩: '))
    english = int(input('请输入您的英语成绩: '))

    # 2. 创建学员对象 -- 类 ? 在student类里面 根据Student类, 创建学生对象
    student = Student(name, chinese, math, english)

    # 3. 将该对象添加到学员列表
    self.student_list.append(student)

```

### 3、3、2、4 显示全部信息

```

def show_student(self):
    # 1. 打印表头
    print('姓名\t\t语文\t\t数学\t\t英语\t\t总分')

    # 2. 打印学员数据
    for stu in self.student_list:

        print(f'{stu.name}\t\t{stu.chinese}\t\t{stu.math}\t\t{stu.english}\t\t{stu.total}')

```

### 3、3、2、5 查询学生信息

```

def search_student(self):
    # 1. 用户输入目标学员姓名
    search_name = input('请输入您要查询的学员姓名: ')

    # 2. 遍历列表。如果学员存在打印学员信息，否则提示学员不存在
    for stu in self.student_list:
        if search_name == stu.name:
            print('姓名\t\t语文\t\t数学\t\t英语\t\t总分')

            print(f'{stu.name}\t\t{stu.chinese}\t\t{stu.math}\t\t{stu.english}\t\t{stu.total}')
            break
        else:
            print('该学生不存在或者姓名输入错误! ')

```

### 3、3、2、6 查询学生信息

```

def del_student(self):
    # 1. 用户输入目标学员姓名
    del_name = input('请输入要删除的学员姓名: ')

    # 2. 遍历学员列表，如果用户输入的学员存在则删除学员对象，否则提示学员不存在
    for i in self.student_list:
        if del_name == i.name:
            # 删除该学员对象
            self.student_list.remove(i)
            break
        else:
            # 循环正常价结束执行的代码：循环结束都没有删除任何一个对象，所以说明用户输入的目标学员不存在
            print('该学生不存在或者姓名输入错误! ')

```

### 3、3、2、7 修改学生信息

```

def modify_student(self):
    # 1. 用户输入目标学员姓名
    modify_name = input('请输入要修改的学员姓名: ')

    # 2. 遍历列表数据，如果学员存在修改姓名性别手机号，否则提示学员不存在
    for stu in self.student_list:
        if modify_name == stu.name:

```

```

# 重新输入成绩
name = input('请重新输入学生的姓名:') # --> str
chinese = input('请重新输入学生语文成绩:')
math = input('请重新输入学生数学成绩:')
english = input('请重新输入学生英语成绩:')

if name:
    stu.name = name
if chinese:
    stu.chinese = int(chinese)
if math:
    stu.math = int(math)
if english:
    stu.english = int(english)
# stu.total = stu.chinese + stu.math + stu.english
print(f'{modify_name}修改学员信息成功')
break
else:
    print('该学生不存在或者姓名输入错误! ')

```

### 3、3、2、8 保存学员信息

```

def save_student(self):
    # 1. 打开文件
    f = open('students.json', mode='w', encoding='utf-8')

    # [对象] 转换成 [列表]
    final_stu_list = [{"name": i.name, "chinese": i.name, "math": i.name,
"english": i.name, "total": i.name} for i in self.student_list]

    # 2. 文件写入数据
    # 2.1 [字典] 转换成 [字符串]
    students_str = json.dumps(final_stu_list, ensure_ascii=False)
    # 2.2 文件写入 字符串数据
    f.write(students_str)
    # 3. 关闭文件
    f.close()

```