

## 19-基础总结 练习题答案

---

关于python的基础知识部分，已经学完了，下面让我们用这些知识来做一个小项目~

程序开始根据输入内容判断打印老师信息还是学生信息

定义一个年级类，用来返回年级，定义一个班级类返回班级，定义一个老师类继承年级和班级类，

在老师类中定义一个run方法，在调用时可以打印出老师所在的年级，班级，学科，姓名信息，定义一个学生类继承年级和班级类，在该类中定义一个run方法用来打印学生的姓名，年龄，年级，班级信息。

参考答案：

# 创建年级类

```
class Grade():
    #定义初始化方法
    def __init__(self, grades):
        #创建实例化属性self.grades, self.grades_name
        self.grades = str(grades)
        self.grades_name = '年级'
    #定义str魔法方法
    def __str__(self):
        #str方法的返回值为实例属性self.grades
        return self.grades
```

# 创建班级类

```
class Banji():
    #定义初始化方法
    def __init__(self, banji):
        #创建实例化属性self.grades, self.banji, self.banji_name
        self.grades = str(grades)
        self.banji = str(banji)
        self.banji_name = '班级'
    #定义str魔法方法
    def __str__(self):
        #str方法的返回值为实例属性self.grades
        return self.banji
```

# 创建老师类，继承Grade类和Banji类

```
class Teacher(Grade, Banji):
    #定义初始化方法
    def __init__(self, name, subject, grades_name, banji_name):
        #使用类名调用父类Grade的init方法
```

```

    Grade.__init__(self, grades_name)
    #使用类名调用父类Banji的init方法
    Banji.__init__(self, banji_name)
    #创建实例属性name和subject
    self.name = name
    self.subject = subject
#定义实例方法run
    def run(self):
        #进行格式化输出, 依次进行接收
        print('我是老师%s,%s年级, %d班的%s老师' % (self.name, self.grades,
int(self.banji), self.subject))

# 创建学生类, 继承父类Grade和 Banj
class Student(Grade, Banji):
    #定义初始化方法
    def __init__(self, name, age, grades_name, banji_name):
        #使用类名调用父类Grade的init方法
        Grade.__init__(self, grades_name)
        #使用类名调用父类Banji的init方法
        Banji.__init__(self, banji_name)
        #创建实例属性name和subject
        self.name = name
        self.age = age
    #定义实例方法run
    def run(self):
        #格式化输出
        print('我是%s, %s年级, %s班的学生,今年%s岁' % (self.name, self.grades,
self.banji, self.age))

#创建实例对象grade
grade = Grade('2') # 得到年级
#进行print输出时, 会调用str方法
print('年级', grade)
#创建实例对象banji
banji = Banji('3')
#进行print输出时, 会调用str方法
print('班级', banji)
#创建实例对象tea
tea = Teacher('name', 'sub', grade, banji)
#调用实例方法
tea.run()
#创建实例对象stu
stu = Student('小张', 19, grade, banji)
#调用实例方法
stu.run()

```

解题思路:

python面向对象种

`__init__`: 改造初始化方法, 可以让创建对象更加灵活

`__str__`: `print`输出对象的时候, 只要自己定义了 `__str__`方法, 那么就会打印从在这个方法中`return`的数据

首先定义年级和班级类, 分别使用`__init__`方法进行初始化, 然后使用 `__str__`方法进行返回年级和班级

定义老师类继承年级和班级类, 定义`run`方法, 进行格式化输出

定义学生类继承年级和班级类, 定义`run`方法, 进行格式化输出