

传智播客

《Python 程序开发案例教程》

教学设计

课程名称：Python 程序开发案例教程

授课年级：2019 年级

授课学期：2019 学年第一学期

教师姓名：某某老师

2019 年 09 月 09 日

课题名称	第7章 类与面向对象	计划学时	7学时
内容分析	面向对象是程序开发领域的重要思想，这种思想模拟了人类认识客观世界的逻辑，是当前计算机软件工程学的主流方法；类是面向对象的实现手段。Python在设计之处就已经是一门面向对象语言，了解面向对象编程思想对于学习Python开发至关重要。		
教学目标及基本要求	<ol style="list-style-type: none"> 1. 理解面向对象的概念，明确类和对象的含义 2. 掌握类的定义与使用方法 3. 熟练创建对象、访问对象成员 4. 掌握实现成员访问限制的意义，可熟练访问受限成员 5. 了解构造方法与析构方法的功能与定义方式 6. 熟悉类方法和静态方法的定义与使用 		
教学重点	<ol style="list-style-type: none"> 1. 类的定义与访问 2. 对象的创建与访问 3. 单继承 4. 多继承 5. 方法的重写 6. super()函数 		
教学难点	<ol style="list-style-type: none"> 1. 访问限制 2. 构造方法 3. 类方法 4. 静态方法 5. 方法的重写 6. 多态 		
教学方式	教学采用教师课堂讲授为主，使用教学PPT讲解		

教 学 过 程	<p style="text-align: center;">第一课时 (面向对象概述、面向对象的基本概念)</p> <p>一、创设情境，引出面向对象</p> <p>(1) 教师通过提出需求，引出什么是面向对象。</p> <p>(2) 明确学习目标</p> <ul style="list-style-type: none">● 要求学生了解面向对象概述● 要求学生了解面向对象的基本概念 <p>二、进行重点知识的讲解</p> <p>(1) 教师根据课件，讲述面向对象的概述。</p> <p>教师根据课件由面向过程引出面向对象概述，面向对象编程的着眼之处在于角色以及角色之间的联系。使用面向对象编程思想解决问题时，开发人员首先会从问题之中提炼出问题涉及的角色，将不同角色各自的特征和关系进行封装，以角色为主体，通过描绘角色的行为去描述解决问题的过程。</p> <p>(2) 教师根据课件，讲述面向对象的基本概念。</p> <p>教师根据分别讲解对象、类、抽象、封装、继承、多态等概念为学生理解面向对象的概念。</p> <p>对象：一般意义上讲，对象是现实世界中可描述的事物，它可以是有形的也可以是无形的，一本书、一家图书馆，都可以称为对象。</p> <p>类：从具体的事物中把事物中的共同的特征抽取出来，找出事物间的共性，抽象出一个概念模型，就是一个定义的一类。</p> <p>抽象：抽象是抽取特定实例的共同特征。</p> <p>封装：封装是将数据和数据处理过程封装成一个整体，以实现独立性很强的模块。</p> <p>继承：继承描述的是类与类之间的关系，通过继承，新生类可以在无需赘写原有类的情况下，对原有类的功能进行扩展。</p> <p>三、归纳总结，布置作业/随堂练习</p> <p>(1) 回顾上课前的学习目标，并对本节课的内容进行总结。</p> <p>教师总结本节课需要掌握的知识点，包括面向对象的概述、面向对象的基本概念。</p>
------------------	---

(2) 布置随堂练习，检查学生掌握情况。

根据博学谷和随堂练习资源，给学生布置随堂练习，检测学生的掌握程度，并对学生出现的问题进行解决。

(3) 使用博学谷系统下发课后作业。

第二课时

(类和对象的关系、类的定义与访问、对象的创建与使用、访问限制)

一、回顾上节课的内容，继续讲解本课时的知识

(1) 教师对学生们的疑问进行统一答疑。

(2) 回顾总结上节课内容，继续介绍本课时的内容。

上节课我们学习了面向对象概述与面向对象基本概念，本节课将带领大家学习类和对象的关系、类的定义与访问、对象的创建与使用、访问限制。

(3) 明确学习目标

- 要求学生了解类和对象的关系
- 要求学生掌握类的定义
- 要求学生掌握对象的创建与使用
- 要求学生掌握访问限制

二、进行重点知识的讲解

(1) 教师根据课件，讲解类和对象的关系。

面向对象编程思想力求在程序中对事物的描述与该事物在现实中的形态保持一致。为此，面向对象的思想中提出了两个概念：类和对象。类是对多个对象共同特征的抽象描述，是对象的模板；对象用于描述现实中的个体，它是类的实例。

(2) 教师根据课件，讲解类的定义方式。

在 Python 中使用关键字 `class` 定义类，其格式为：

`class` 类名:

 属性名 = 属性值

 def 方法名(self):

 方法体

上述格式中的 `class` 是定义类的关键字，其后的类名是类的标识符，类名首字母一般为大写。类名后的冒号(:)必不可少，之后的属性和方法都是类的成员，

其中属性类似 Python 中的变量，方法类似函数，但需要注意，方法中有一个指向对象的默认参数 `self`。

(3) 教师根据课件，讲解对象的创建与使用。

在 Python 中类定义完成之后，并不能直接使用，需要通过实例化对象才能使用，创建对象的格式为：`对象名 = 类名()`。在访问对象成员时可通过对象名.属性或对象名.方法()来访问属性或方法。

(4) 教师根据课件，讲解什么是访问限制。

类中定义的属性和方法默认为公有属性和方法，该类的对象可以任意访问类的公有成员，但考虑到封装思想，类中的代码不应该被外部轻易访问。为了契合封装原则，Python 支持将类中的成员设置为私有成员，在一定程度上限制对象对类成员的访问。Python 通过在类成员名之前添加双下划线(`__`)来限制成员的访问权限，其语法是为 `__属性名`、`__方法名`。

三、归纳总结，布置作业

(1) 回顾学习目标，总结本节课需要了解类和对象的关系、掌握类的定义与访问、掌握对象的创建与使用、掌握访问限制。

(2) 布置随堂练习，检查学生掌握情况。

根据博学谷和随堂练习资源，给学生布置随堂练习，检测学生的掌握程度，并对学生出现的问题进行解决。

(3) 使用博学谷系统下发课后作业。

第三课时

(构造方法、析构方法、类方法、静态方法)

一、回顾上节课内容，继续介绍本课时的内容

(1) 教师对学生们的疑问进行统一答疑。

(2) 教师通过提问学生问题，由上一课时引出本课时要讲解的内容。

(3) 明确学习目标

- 要求学生掌握构造方法
- 要求学生了解析构方法
- 要求学生掌握类方法
- 要求学生掌握静态方法

二、进行重点知识的讲解

(1) 教师根据课件，讲解什么是构造方法。

Python 中 `__init__()` 表示构造方法，在每个类中都有一个默认的 `__init__()` 方法，如果在定义类时显式地定义了 `__init__()` 方法，则创建对象时 Python 解释器会调用显式定义的 `__init__()` 方法；如果定义类时没有显式定义 `__init__()` 方法，那么 Python 解释器会调用默认的 `__init__()` 方法。

(2) 教师根据课件，讲解什么是类方法。

Python 中类方法使用 `@classmethod` 修饰，类方法的第一个参数为 `cls` 而非 `self`，类方法即可有对象调用，亦可由类调用，类方法可以修改类属性。

(3) 教师根据课件，讲解什么是静态方法。

Python 中静态方法使用 `@staticmethod` 修饰，静态方法中没有 `self` 参数，静态方法使用“类名.方法/属性名”形式访问类的成员；类方法即可由对象调用，也可以使用类直接调用。

三、归纳总结，布置作业

(1) 回顾学习目标，总结本节课需要掌握的构造方法、析构方法、类方法、静态方法。

(2) 使用博学谷系统下发课后作业。

第四课时

(实例 1：银行管理系统、单继承、多继承)

一、回顾上节课内容，继续介绍本课时的内容

(1) 教师对学生们的疑问进行统一答疑。

(2) 教师通过提问学生问题，由上一课时引出本课时要讲解的内容。

(3) 明确学习目标

- 要求学生了解实例 1：银行管理系统的实现过程
- 要求学生掌握单继承
- 要求学生掌握多继承

二、进行重点知识的讲解

(1) 教师根据课件，讲解实例 1：银行管理系统。

教师根据教学资源实现银行管理系统，并为学生讲解实现过程。

(2) 教师根据课件，讲解什么继承。

在 Python 中，类与类之间也具有继承关系，其中被继承的类称为父类或基类，继承的类称为子类或派生类。子类继承父类时，会自动拥有父类中的方法和属性。

(3) 教师根据课件，讲解什么是单继承。

单继承指的是只继承一个父类，其格式为 `class 子类(父类):pass`

(4) 教师根据课件，讲解什么是多继承。

多继承指的是一个子类继承多个父类，其格式为 `class 子类(父类 A,父类 B):pass`。

三、归纳总结，布置作业

- (1) 回顾学习目标，总结本节课需要了解实例：银行管理系统实现过程，掌握单继承、多继承。
- (2) 使用博学谷系统下发课后作业。

第五课时

(方法的重写、super()函数、实例 2：井字棋、多态)

一、回顾上节课内容，继续介绍本课时的内容

- (1) 教师对学生们的疑问进行统一答疑。
- (2) 教师通过提问学生问题，由上一课时引出本课时要讲解的内容。
- (3) 明确学习目标

- 要求学生掌握方法的重写
- 要求学生掌握 super()函数
- 要求学生掌了解实例 2：井字棋实现过程
- 要求学生了解多态

二、进行重点知识的讲解

(1) 教师根据课件，讲解方法的重写。

子类可以继承父类的属性和方法，若父类的方法不能满足子类的要求，子类可以重写父类的方法，以实现理想的功能。

(2) 教师根据课件，讲解 super()函数的使用。

如果子类重写了父类的方法，但仍希望调用父类的方法，可以使用 Python

中 `super()` 函数，其使用格式为 `super().方法名()`。

(3) 教师根据课件，讲解实例 2：井字棋。

教师根据教材资源实现实例 2：井字棋，并向学生讲解其实现过程。

(4) 教师根据课件，讲解什么是多态。

Python 中并不需要显式指定对象的类型，只要对象具有预期的方法和表达式操作符，就可以使用对象。也可以说，只要对象支持所预期的“接口”，就可以使用，从而实现多态。

三、归纳总结，布置作业

(1) 回顾学习目标，总结本节课需要掌握方法的重写、`super()` 函数，了解实例 2：井字棋实现过程，了解什么是多态。

(2) 使用博学谷系统下发课后作业。

第六课时 (上机练习)

上机练习主要针对本章中需要重点掌握的知识点，以及在程序中容易出错的内容进行练习，通过上机练习可以考察同学对知识点的掌握情况，对代码的熟练程度。

上机一：（练习教材 7.1-7.5 示例代码）

形式：独立完成

要求：

(1) 要求学生能够熟练掌握教材中示例代码。

(2) 要求学生能够自己实现实例 1 程序。

第七课时 (上机练习)

上机练习主要针对本章中需要重点掌握的知识点，以及在程序中容易出错的内容进行练习，通过上机练习可以考察同学对知识点的掌握情况，对代码的熟练程度。

上机一：（练习教材 7.6-7.8 示例代码以及实例 2）

形式：独立完成

要求：

	<p>(1) 要求学生能够熟练掌握教材中示例代码。</p> <p>(2) 要求学生能够自己实现实例 2 程序。</p>
思考题 和习题	见教材第 7 章配套的习题
教 学 后 记	