

黑马程序员

《Python 快速编程入门（第2版）》

教学设计

课程名称：Python 快速编程入门

授课年级：2020 年级

授课学期：2020 学年第二学期

教师姓名：某某老师

2020 年 09 月 09 日

课题名称	第 8 章 面向对象	计划学时	8 课时
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思维方式，将开发中遇到的事物皆看作对象。Python 支持面向对象编程，且 3.x 版的 Python 源码全部基于面向对象设计，因此了解面向对象编程思想对 Python 学习而言非常重要。本章将针对面向对象的相关知识进行详细地讲解。		
教学目标及基本要求	<ol style="list-style-type: none"> 1. 理解面向对象编程思想 2. 明确类和对象的关系，可熟练独立设计和使用类 3. 掌握类的属性和方法 4. 掌握构造方法和析构方法的使用 5. 理解面向对象的三大特性：封装、继承、多态，并能将其熟练地运用到程序开发中 6. 熟悉运算符的重载 		
教学重点	<ol style="list-style-type: none"> 1. 明确类和对象的关系，可熟练独立设计和使用类 2. 掌握类的属性和方法 3. 掌握构造方法和析构方法的使用 4. 理解面向对象的三大特性：封装、继承、多态，并能将其熟练地运用到程序开发中 		
教学难点	<ol style="list-style-type: none"> 1. 理解面向对象编程思想 2. 掌握类的属性和方法 3. 理解面向对象的三大特性：封装、继承、多态，并能将其熟练地运用到程序开发中 		
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</p> <p style="text-align: center;">（面向对象概述，类与对象的基础应用，类的成员）</p> <p>一、创设情境，导入面向对象的新概念</p> <p>（1）教师通过提问学生问题，引出面向过程和面向对象思想。</p> <p style="color: blue;">教师向学生提问问题，例如问题是：灯坏了，我们怎么解决呢？教师根据学</p>		

生的回答进行总结：我们可以通过两种方式完成，一种方式是自己买灯管、拆灯管、换灯管；另一种方式是找个维修师傅修理。第一种方式是面向过程的解决思路，关注解决问题的步骤，另一种方式是面向对象的解决思想，关注解决问题的对象。

（2）什么是面向对象？

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思维方式，将开发中遇到的事物皆看作对象。

（3）明确学习目标

- 要求学生理解面向对象编程的思想
- 要求学生掌握类的定义和使用
- 要求学生掌握属性的使用

二、进行重点知识的讲解

（1）教师根据课件，对比面向过程介绍面向对象的编程思想。

面向过程是早期开发语言中大量使用的编程思想，基于这种思想开发程序时一般会先分析解决问题的步骤，使用函数实现每个步骤的功能，之后按步骤依次调用函数。面向对象思想与面向过程不同，它关注的不是解决问题的过程，基于此种思想开发程序时会先分析问题，从中提炼出多个对象，将不同对象各自的特征和行为进行封装，之后通过控制对象的行为来解决问题。

（2）教师根据课件，分别介绍基于面向过程和面向对象开发五子棋游戏的思路，以区分这两种思想的区别。

（3）教师根据课件，介绍文件的打开与关闭操作，并使用代码进行演示。。

Python 可通过内置函数 `open()` 打开文件，可通过 `close()` 方法关闭文件，也可以使用 `with` 语句实现文件的自动关闭。

（4）什么是类和对象？

面向对象编程中有两个核心概念：类和对象，其中对象映射了现实生活中真实存在的事物，它可以看得见摸得着，例如，你现在手里的这本书就是一个对象；类是抽象的，它是对一群具有相同特征和行为的事物的统称，例如，“书是人类进步的阶梯”，这里的书并不具体指哪一本，而是一个类。简单地说，类是现实具有相同特征的一些事物的抽象，对象是类的实例。

（5）教师根据课件，介绍类定义的内容，并使用代码进行演示。

程序中类有一个名称，包含描述类特征的数据成员，以及描述类行为的成员函数，其中数据成员称为属性，成员函数称为方法。Python 使用关键字 `class` 来定义一个类。

(6) 教师根据课件，介绍对象创建与使用的内容，并使用代码进行演示。

程序若想要使用类，则需要根据类创建一个对象。

(7) 教师根据课件，介绍属性的内容，并使用代码进行演示。

属性按声明的方式可以分为两类：类属性和实例属性。类属性是声明在类内部、方法外部的属性，它可以通过类或对象进行访问，但只能通过类进行修改；实例属性是在方法内部声明的属性，Python 支持动态添加实例属性。

三、归纳总结，布置课后作业

(1) 回顾上课前的学习目标，对本节课知识点进行总结。

教师带领学生总结本节课需要掌握的知识点，包括面向对象思想、类的定义、对象的创建与使用、属性。

(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

(类的成员，特殊方法)

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) 回顾上个课时所学习的内容，继续介绍本课时的内容。

上节课主要为大家介绍了面向对象、类的定义与使用、属性，接下来，本节课将为大家介绍类的其它成员，包括方法、私有成员及特殊方法。

(3) 明确学习目标

- 要求学生掌握方法的使用
- 要求学生掌握私有成员的使用
- 要求学生掌握构造方法与析构方法

二、进行重点知识的讲解

(1) 教师根据课件，介绍方法的内容，并使用代码进行演示。

Python 中的方法按定义方式和用途可以分为三类：实例方法、类方法和静态方法。实例方法形似函数，但它定义在类内部、以 `self` 为第一个形参；类方法是定义在类内部、使用装饰器 `@classmethod` 修饰的方法；静态方法是定义在类内部、使用装饰器 `@staticmethod` 修饰的方法。

(2) 实例方法、类方法、静态方法有什么区别？

实例方法是定义在类内部、以 `self` 为第一个形参的方法，它只能通过对象调用；类方法是定义在类内部、使用装饰器 `@classmethod` 修饰的方法，它可以通过类和对象调用；静态方法是定义在类内部、使用装饰器 `@staticmethod` 修饰的方法，它可以通过类和对象调用。

(3) 教师根据课件，引出私有成员的概念。

类的成员默认是公有成员，可以在类的外部通过类或对象随意地访问，这样显然不够安全。为了保证类中数据的安全，Python 支持将公有成员改为私有成员，在一定程度上限制在类的外部对类成员的访问。

(4) 教师根据课件，介绍私有成员的内容，并使用代码进行演示。

Python 通过在类成员的名称前面添加双下划线(`__`)的方式来表示私有成员。类的私有成员只能在类的内部直接访问，可以在类的外部通过类的公有方法间接访问。

(5) 教师根据课件，介绍构造方法的内容，并使用代码进行演示。

构造方法（即 `__init__()` 方法）是类中定义的特殊方法，该方法负责在创建对象时对对象进行初始化。每个类默认都有一个 `__init__()` 方法，如果一个类中显式地定义了 `__init__()` 方法，那么创建对象时调用显式定义的 `__init__()` 方法；否则调用默认的 `__init__()` 方法。

(6) 教师根据课件，介绍析构方法的内容，并使用代码进行演示。

析构方法（即 `__del__()` 方法）是销毁对象时系统自动调用的方法。每个类默认都有一个 `__del__()` 方法，如果一个类中显式地定义了 `__del__()` 方法，那么销毁该类的对象时会调用显式定义的 `__del__()` 方法；如果一个类中没有定义 `__del__()` 方法，那么销毁该类的对象时会调用默认的 `__del__()` 方法。

三、归纳总结，布置作业

(1) 回顾上课前的学习目标，对本节课知识点进行总结。

教师带领学生总结本节课需要掌握的知识点，包括方法、私有成员、构造方法、析构方法。

(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

(实训案例，封装)

一、回顾上节课内容，继续讲解本节课的内容

(1) 教师对学生们的疑问进行统一答疑。

(2) 回顾上个课时所学习的内容，继续介绍本课时的内容。

上节课主要为大家介绍了方法、私有成员、构造方法和析构方法，接下来，本节课先结合前面的知识点演示两个精彩的实例，再为大家介绍面向对象一个重要特性——封装。

(3) 明确学习目标

- 要求学生理解封装的含义
- 要求学生能熟练地应用封装到程序开发中

二、进行重点知识的讲解

(1) 教师根据课件，介绍 8.5.1 的实例，并使用代码进行演示。

(2) 教师根据课件，介绍 8.5.2 的实例，并使用代码进行演示。

(3) 什么是封装？

封装是面向对象的重要特性之一，它的基本思想是对外隐藏类的细节，仅提供一个用于访问类成员的公开接口。如此，类的外部无需知道类的实现细节，只需要使用公开接口便可访问类的内容，这在一定程度上保证了类内数据的安全。

(4) 教师根据课件，介绍封装的内容，并使用代码进行演示。

三、归纳总结，布置作业

(1) 回顾上课前的学习目标，对本节课知识点进行总结。

教师带领学生总结本节课需要掌握的封装。

(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。

第四课时

(继承，多态)

一、回顾上节课内容，继续讲解本节课的内容

(1) 教师对学生们的疑问进行统一答疑。

(2) 回顾上个课时所学习的内容，继续介绍本课时的内容。

上节课主要为大家介绍了封装的内容，接下来，本节课将为大家继续介绍面向对象其它两个重要特性——继承和多态。

(3) 明确学习目标

- 要求学生理解什么是继承和多态
- 要求学生能熟练地应用继承、多态到程序开发中

二、进行重点知识的讲解

(1) 什么是继承？

继承是面向对象的重要特性之一，它主要用于描述类与类之间的关系，在不改变原有类的基础上扩展原有类的功能。若类与类之间具有继承关系，被继承的类称为父类或基类，继承其他类的类称为子类或派生类，子类会自动拥有父类的公有成员。

(2) 教师根据课件，介绍单继承的内容，并使用代码进行演示。

单继承即子类只继承一个父类。若在定义类时不指明该类的父类，那么该类默认继承基类 `object`。

(3) 教师根据课件，介绍多继承的内容，并使用代码进行演示。

程序中的一个类也可以继承多个类，如此子类具有多个父类，也自动拥有所有父类的公有成员。

(4) 教师根据课件，引出重写这一概念。

程序中，子类会原封不动地继承父类的方法，但子类有时需要按照自己的需求对继承来的方法进行调整，也就是在子类中重写从父类继承来的方法。

(5) 教师根据课件，介绍重写的内容，并使用代码进行演示。

Python 中实现方法重写非常简单，只要在子类中定义与父类方法同名的方法，在方法中按照子类需求重新编写功能代码即可。

(6) 什么是多态？

多态是面向对象的重要特性之一，它的直接表现即让不同类的同一功能可以通过同一个接口调用，表现出不同的行为。

(7) 教师根据课件，介绍多态的内容，并使用代码进行演示。

三、归纳总结，布置作业

(1) 回顾上课前的学习目标，对本节课知识点进行总结。

教师带领学生总结本节课需要掌握的知识点，包括单继承、多继承、重写和多态。

(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。

第五课时

(运算符重载，实训案例)

一、回顾上节课内容，继续讲解本节课的内容

(1) 教师对学生们的疑问进行统一答疑。

(2) 回顾上个课时所学习的内容，继续介绍本课时的内容。

上节课主要为大家介绍了继承和多态的内容，接下来，本节课先为大家运算符重载的内容，并结合前面的知识演示两个精彩的实例。

(3) 明确学习目标

- 要求学生熟悉运算符的重载

二、进行重点知识的讲解

(1) 什么是运算符重载？

运算符重载是指赋予内置运算符新的功能，使内置运算符能适应更多的数据类型。当定义一个类时，如果这个类中重写了 Python 基类 object 内置的有关运算符的特殊方法，那么该特殊方法对应的运算符将支持对该类的实例进行运算。

(2) 教师根据课件，介绍特殊方法及其对应的运算符，并使用代码进行演示。

(3) 教师根据课件，介绍 8.10.1 的实例，并使用代码进行演示。

(4) 教师根据课件，介绍 8.10.2 的实例，并使用代码进行演示。

三、归纳总结，布置作业

(1) 回顾上课前的学习目标，对本节课知识点进行总结。

教师带领学生总结本节课需要熟悉的运算符重载。

(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。

第六课时

(阶段案例——银行管理系统)

一、回顾上节课内容，继续讲解本节课的内容

(1) 教师对学生们的疑问进行统一答疑。

(2) 回顾上个课时所学习的内容，继续介绍本课时的内容。

上节课主要为大家介绍了运算符重载的内容，并演示了两个实例，接下来，本节课将结合本书已经讲过的知识开发一个阶段案例——银行管理系统。

(3) 明确学习目标

- 要求学生掌握面向对象编程

二、进行重点知识的讲解

(1) 教师根据课件的需求，引出银行管理系统。

银行管理系统是一个集开户、查询、取款、存款、转账、锁定、解锁、退出等一系列业务的管理系统，随着计算机技术在金融行业的广泛应用，银行企业采用管理系统替代了传统手工记账的方式，这极大地缩短了用户办理基础储蓄业务的时间，提升了银行企业的形象。

(2) 教师根据课件的需求，介绍阶段案例的目标。

本案例要求编写程序，实现一个基于面向思想的、具有开户、查询、取款、存款、转账、锁定、解锁、退出功能的银行管理系统。

- (3) 教师根据课件，分析阶段案例的实现思路。
- (4) 教师根据课件，介绍阶段案例的实现过程，并使用代码进行演示。
- (5) 教师根据课件，介绍阶段案例的测试，并使用代码进行演示。

三、归纳总结，布置作业

- (1) 回顾上课前的学习目标，对本节课知识点进行总结。

教师带领学生总结本节课开发的银行管理系统。

- (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：（考察知识点为类的定义、属性、方法、构造方法）

形式：单独完成

题目：

请按照 8.5.1 的实例要求，开发一个具有添加、删除、修改、查找联系人信息及查看所有联系人信息的通讯录。

上机二：（考察知识点为类的定义、属性、方法、构造方法、继承）

形式：单独完成

题目：

请按照 8.10.1 的实例要求，实现一个基于面向对象思想的人机猜拳游戏。

第八课时

(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

	<p>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。</p> <p>上机一：（考察知识点为类的定义、属性、方法）</p> <p>形式：独立完成</p> <p>题目：</p> <p>请按照阶段案例的要求，实现一个具有开户、查询、取款、存款、转账、锁定、解锁、退出功能和基于面向对象开发的银行管理系统。</p>
思考题 和习题	见教材第 8 章配套的习题
教 学 后 记	