**Java进阶**

**第一单元 枚举**

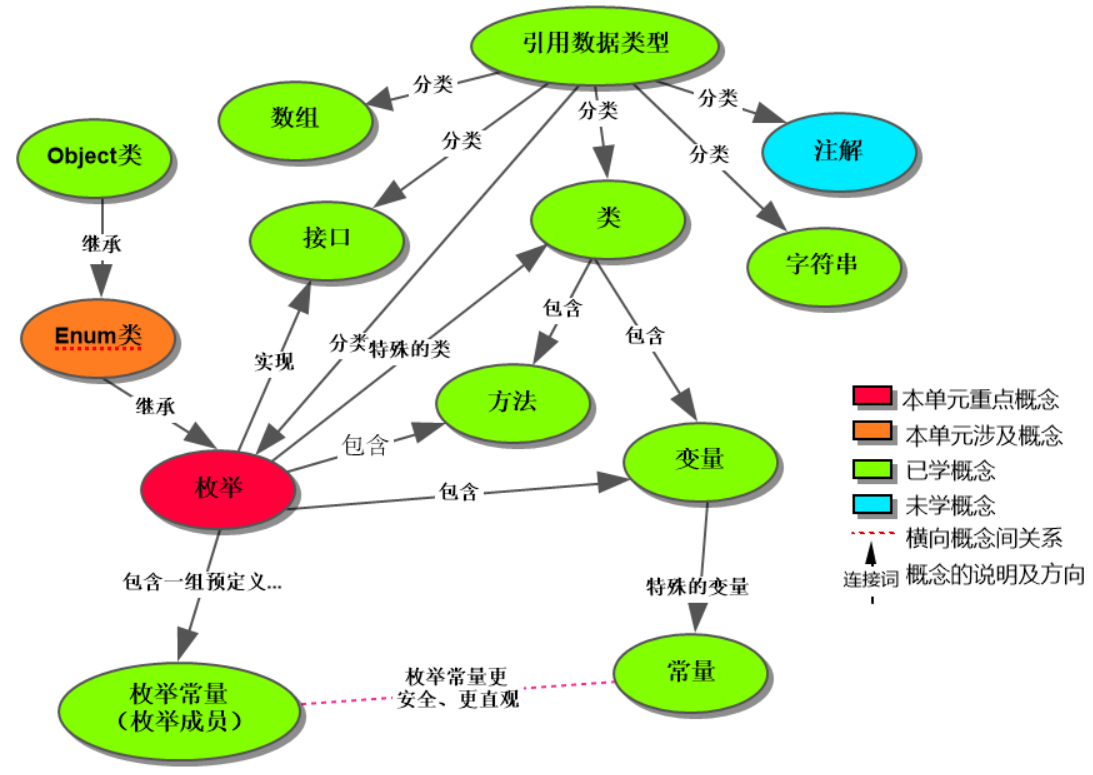
1. **本单元教学目标**
2. **知识目标**
3. 理解枚举的概念
4. 理解枚举的特点
5. 掌握枚举的语法实现
6. **能力目标**
7. 会使用枚举实现常量的定义
8. 会使用枚举实现接口
9. 会使用枚举实现抽象类
10. 能够在switch结构中使用枚举
11. **课时分配**
12. **第一课时：预备课**
13. **第二课时：复习Java基础（一）**
14. **第三课时：复习Java基础（二）**
15. **第四课时：枚举**
16. **知识目标**

* 掌握枚举的概念

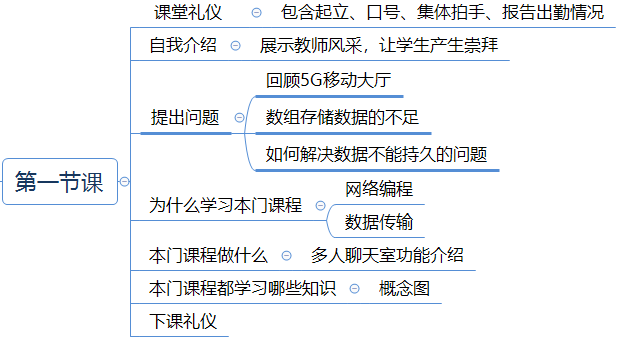
1. **能力目标**

* 掌握枚举定义和调用的方法
* 能够在switch结构中使用枚举

1. **本单元概念图**

****

1. **教学过程脚本设计—第一课时**
2. **脚本设计思路**



1. **脚本切片设计思路**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 内容 | 目标 | 设计亮点 | 时间 |
| 1 | 课堂礼仪 | 师生互相问好，调整状态，集中精力 | 学院口号，强化学生信念 |  |
| 2 | 自我介绍 | 展示讲师自身特点，让学生产生“崇拜” | 讲解工作经历、教学经历 |  |
| 3 | 提出问题 | 1. 回顾Java基础的知识体系结构 2. 通过回顾贯穿案例，讨论Java基础的应用局限 |  |  |
| 4 | 为什么学习本门课程 | 通过实际应用来引出学习本门课程的原因 |  |  |
| 5 | 贯穿案例 | 1、讲解本门课程贯穿案例 |  |  |
| 6 | 概念图 | 1、讲解本门课概念图 |  |  |

1. **脚本切片一：**

|  |  |  |
| --- | --- | --- |
| **课堂模块** | **教学内容** | **教学标准** |
| **课堂礼仪** | 上课铃响后，  班长喊起立—学生问好—班级口号—鼓掌—整理着装—欢迎教师上课 | 1、上课前，提前30秒面带自信笑容走进教室，观察、问候学生们(等待上课)  2、上课铃声响起！  3、老师检查学生工装，整理工装速度及姿势  IMG_256 IMG_256 |
| **讲师：**上课  **班长:** 起立  **班长:** 我们班级的口号是  **全班:** 。。。。。  **班长:** 开始  **全班:** 拍手操  **班长:** 整理工装  **讲师:** 大家请座 |  |

1. **脚本切片二：**

|  |  |  |
| --- | --- | --- |
| **课堂模块** | **教学内容/台词** | **教学标准** |
| 自我介绍 | **讲师自我介绍** | 1. 吸引学员注意力，让学员对讲师有基本的了解和信任 2. 建立师生情感上的联接 3. 明确讲师与课程的关系是专业的、权威的，从而让学生对课程有期待和兴奋 |
| **讲师：**  根据自身教育、工作经历完成自我介绍 | 1. 展示PPT个人基本信息：姓名、工作经历、课程关系 |

1. **脚本切片三：**

|  |  |  |
| --- | --- | --- |
| **课堂模块** | **教学内容/台词** | **教学标准** |
| **回顾概念图** | **1、提出问题引入**  讲师：同学们，大家回想一下，在上个月Java基础的课程中，大家都学习了哪些内容？  学生：【省略答案】  讲师：主要学习了程序控制结构、数组、面向对象、常用类等基础知识。  讲师：除了这四部分内容，还有一些其他的基础知识，通过概念图来看一下。  讲师：在第一单元中，学习了如何配置环境变量、如何安装JDK，准备开发环境。以及Java语言的特点及发展历程等基本知识。  讲师：从第二单元开始一直到第四单元，学习了Java基本语法，包括运算符、表达式、以及程序控制语句结构。  讲师：计算机是用来处理数据的，那么数据是如何存储的呢？  讲师：存储数据是通过容器来完成的，数组就是一种存储数据的容器。  讲师：那么数组的特点是什么？  学生：长度固定的、存储元素类型必须相同。  讲师：首先使用数组存储数据时必须是同类型数据，其次数组在声时就确定了长度，无法改变。  讲师：这是数组的特点。  讲师：说到数据类型，在Java中包含了基本数据类型和引用数据类型。  讲师：引用数据类型中，又包含了类、接口和枚举。  讲师：类是如何而来的呢？  学生：【省略答案】  讲师：通过对事物的抽象，形成了类。每一个类的实例，又形成对象。所以，类是对象的模板、对象是类的实例。  讲师：在第八和第九单元讲解了如何在类中使用变量和方法。  讲师：除了以上内容，在上个月还学习了一个重要知识就是面向对象的概念。  讲师：从第十单元开始，到第十四单元。分别介绍了面向对象的概念以及四大特征。抽象、封装、继承和多态。  讲师：面向对象是Java语言的核心特征，是上个月课程中的重中之重。  讲师：在第十七单元，讲解实现类型转换的两种方式，并介绍Java核心类库中常用的包以及工具类。  讲师：这些工具类包含了一些特定的方法，使用方便快捷。例如、字符串的操作，算术运算的操作等。  讲师：在第十八单元中则最后讲解了Java的异常处理机制。包括在程序运行过程中可能引发的异常类型，以及针对异常所能采用的处理方法。  讲师：以上就是在上个月Java基础课程中，学习的主要内容。 | 1. 设疑逐步引导 2. 学生小组讨论 |
| **回顾贯穿案例** | **2、贯穿案例**  讲师：通过运用Java基础所学的相关知识，在上个月完成了5G移动大厅的贯穿案例。  讲师：大家还记得5G移动大厅的功能有那些吗？  学生：【省略答案】  讲师：大家在完成这些功能模块时，是不是使用程序控制结构、数组、面向对象这些知识？  学生：是。  讲师：那么大家回想一下。  讲师：在贯穿案例中，使用什么来保存用户信息呢？  学生：数组。  讲师：数组的特点是长度固定，用户数量的增加怎么办？  学生：【省略答案】  讲师：重新定义数组并不是有效的解决方式。  讲师：在本门课程中将讲解另一种数据存储方式，使用集合保存数据,并且不会因为数据量增加的而产生影响。  讲师：大家看一下这段代码，当创建MobileCard对象时，对象的信息会保存在哪里？  学生：【省略答案】。  讲师：很好，数据都保存在内存中。  讲师：既然保存在内存中，一旦程序运行结束，内存中的数据还会存在吗？  讲师：不存在。  讲师：下次运行时是不是需要再次进行选择？  学生：【省略答案】  讲师：如何能让程序退出后，数据依然存在呢？  讲师：在上课时遇到重点知识时，老师都会要求大家做什么？  学生：做笔记。  讲师：俗话说“好记性不如烂笔头”，将知识写在本子上，不管是今天，还是明天，只要想看的时候就可以看到。  讲师：在计算机中，可以将数据保存到文件中，需要使用时直接从文件中读取使用，数据不会因为程序结束而丢失。  讲师：将数据保存到文件的过程，需要借助文件读写与IO流的操作，这部分内容也会在本门课程中给大家讲解。 | 1. 通过问题引入 2. 引导学员讨论 |

1. **脚本切片四：**

|  |  |  |
| --- | --- | --- |
| **课堂模块** | **教学内容/台词** | **教学标准** |
| 为什么学习本门课 | 1、提出问题  讲师：大家都使用过QQ或者微信，对不对？  学生：【省略答案】  讲师：不论是QQ还是微信，都可以实现在任意的时间和地点与他人进行信息交互。  讲师：但是这些交互的前提条件是什么？  学生：必须有网络。  讲师：是的，有了网络我们才能和别人发起交互，这种交互的实现就是通过网络编程来完成的。  讲师：当建立了网络连接之后，不同设备之间就可以实现数据的传输。 那么大家再想一下，使用QQ给好友传文件时，一次能传几个？  学生：【省略答案】。  讲师：同一个时间，一次传递多个文件，在计算机当中称为“并发”。  讲师：那么计算机为什么可以在同一时间，同时发起多个相同操作呢？  讲师：因为使用线程技术，在生活中很多情况下都应用了线程技术。例如多文件的传输，下载等都是基于线程技术实现的。  讲师：综上所述，大家可以发现，通过Java进阶课程的学习，增加了很多新的内容。  讲师：可以解决Java基础中无法解决的问题，可以解决更复杂的应用问题。  讲师：这也是为什么要学习Java进阶这门课的原因。 |  |

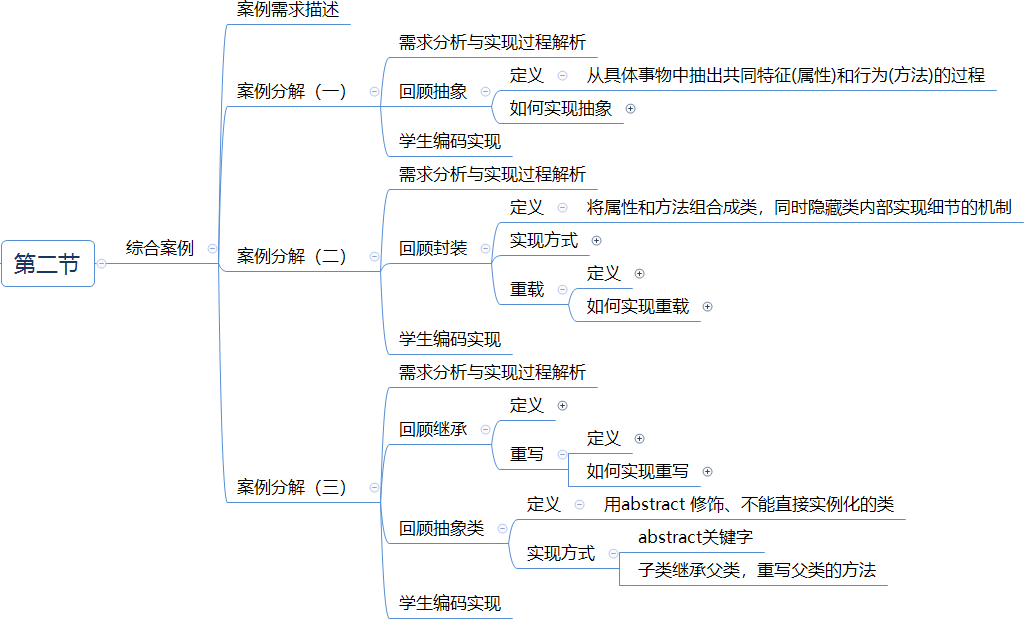
1. **脚本切片五：**

|  |  |  |
| --- | --- | --- |
| **课堂模块** | **教学内容/台词** | **教学标准** |
| **本门课贯穿案例** | 1. **本课程贯穿案例**   讲师：现在我们来看一下，在本门课程中将要完成的贯穿案例。  讲师：通过运用所学的相关知识，在本门课程中，将要完成开心聊天室的功能开发。  讲师：聊天室的主要功能分为用户登录、用户注册、聊天、聊天记录查看、退出系统几个模块。  讲师：其中聊天模块中，用户可以实现群聊、单聊、以及私聊三种方式。  讲师：在聊天记录模块中，用户可以查看聊天记录，也可以删除聊天记录。  讲师：以上是开心聊天室的功能划分。 |  |

1. **脚本切片六：**

|  |  |  |
| --- | --- | --- |
| 本门课概念图 | **1、讲解本门课概念图**  讲师：下面通过概念图给大家展示本门课程中的知识体系结构。  讲师：首先我们已经知道在Java环境中，Java程序必须要经过编译才能在虚拟机中运行。  讲师：虚拟机会将编译后的Java类文件加载到内存中，类的加载分为静态加载和动态加载。  讲师：静态加载就是在编译时就已经需要的类都加载完毕。而动态加载则是在运行时才进行类的加载。  讲师：动态加载是本门课程的重点，典型应用就是反射技术，通过反射技术来动态获取类的信息。  讲师：此外，应用程序在运行时，都会产生一个进程。一个进程可能包含一个或多个线程。  讲师：掌握了线程编程，就可以开发多任务处理的应用程序。  讲师：在Java基础阶段，使用数组来存储数据。但是由于数组存储的数据有限，所以本门课将学习使用集合和Map容器存储数据。  讲师：解决数据存储问题，再来看数据的传输。  讲师：在计算机中，数据可以通过IO流实现传输，IO流按照数据单元分为字节流和字符流。按角色分为处理流和节点流。  讲师：同时 IO操作根据是否阻塞和同步分为BIO、NIO和AIO。  讲师：另外，在本门课程中将介绍java.net核心包，该包提供了用于支持网络编程的相关类库，如ServerSocket套接字和Socket套接字。  讲师：实现网络间的通信，需要借助通信协议来完成，包括TCP协议、UDP协议。  讲师：最后，在开发过程中，不同类型之间数据的转换容易引发异常，导致程序运行错误。  讲师：在程序中引入泛型，通过泛型对数据类型进行约束，则可以解决类型转换异常的问题。  讲师：以上就是本门课的主要内容。 |  |

1. **教学过程脚本设计——第二课时**
2. **脚本设计思路**



1. **脚本切片设计思路**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **内容** | **目标** | **设计亮点** | **时间** |
| 1 | 引入 | 1. 明确Java基础回顾的核心内容 2. 理解并掌握使用面向对象的开发方法 |  |  |
| 2 | 案例分析（一） | 1. 回顾面向对象的概念 2. 复习对象的概念 3. 面向对象的四大特征 |  |  |
| 3 | 案例分析（二） | 1. 理解抽象的概念 2. 理解类的概念 3. 理解类和对象的关系 |  |  |
| 4 | 案例分析（三） | 1. 理解封装的概念 2. 掌握访问修饰符权限的划分 |  |  |

1. **脚本切片一：[视频切片](file:///C:\\Users\\李巍\\AppData\\Roaming\\Microsoft\\Word\\2、回顾和作业点评3’.mp4)1**

|  |  |  |
| --- | --- | --- |
| **课堂模块** | **教学内容/台词** | **教学标准/素材** |
| **引入** | 1、引入回顾内容  讲师：下面我们队Java基础的内容做个复习。  讲师：在Java基础课程中，除了学习Java的基本语法，另一个重点就是学习面向对象的开发方法。  讲师：面向对象的核心内容体现为四大特征，是什么？  学生：【省略答案】  讲师：四大特征包括了抽象、封装、继承和多态。  讲师：为了能够牢固掌握面向对象的开发方法，我们通过一个综合的案例来进行回顾。 |  |
|  | 2、案例介绍  讲师：案例的需求是实现一个员工信息管理系统，完成以下功能。  讲师：包括基本信息的管理、财务工资计算和汇总两大功能。  讲师：在基本信息管理模块中，需要能够获取和修改员工的基本信息，并且能够打印员工信息。  讲师：信息包括姓名、年龄、性别等等。  讲师：在财务工资计算和汇总模块中，能够根据员工的岗位计算员工的当月工资以及工资总额。  讲师：以上即使综合案例的整体需求。 |  |

1. **脚本切片二：[视频切片](file:///C:\\Users\\李巍\\AppData\\Roaming\\Microsoft\\Word\\2、回顾和作业点评3’.mp4)2**

|  |  |  |
| --- | --- | --- |
| **课堂模块** | **教学内容/台词** | **教学标准/素材** |
| **案例分解（一）** | **1、案例分解（一）**  讲师：将综合案例进行分解，首先完成员工信息系统中的基本信息管理功能。  讲师：员工信息包括，姓名、年龄、性别等。  讲师：功能包括打印基本信息。  讲师：­­­­我们分析一下，如果要体现员工的这些信息，需要首先创建什么？  学生：类。  讲师：创建了，那这个类是如何得来的呢？  学生：抽象。  讲师：很好，通过抽象得到类，下面那我们就看一下这个类的具体实现过程。 |  |
|  | **2、创建员工类**  讲师：通过抽象得到的类，具体的创建过程看一下代码。  讲师：首先创建了一个类，类的名称是Employee类。  讲师：大家注意类里面的成员变量都是什么？  学生：【省略答案】  讲师：也就是将员工的属性都作为类的成员变量。  讲师：大家再继续看这里是不是输出功能？  学生：嗯  讲师：在类里输出功能对应成方法。  讲师：经过抽象得到类，将对象的属性抽象成变量，而功能抽象为方法。  讲师：有了这个类，还需要一个测试类。  讲师：在测试类中，创建了员工类的对象实例，通过实例访问成员变量并进行赋值。  讲师：最后通过调用成员方法输出员工信息。  讲师：结果如图显示。 |  |
|  | **3、相关知识概念图**  讲师：大家来看代码的过程中，在实现时是不是用到了对象、类、变量和方法，在得到类的时候还使用了什么？  学生：抽象。  讲师：我们知道面向对象是以对象为中心的开发方法。  讲师：对象是如何而来的？  讲师：对象是客观世界对象进行映射，得到了计算机中的对象。  讲师：客观世界对象的行为和特征分别映射成属性和方法。  讲师：对象的属性和方法经过抽象就构成了类的成员。  讲师：此外抽象作为面向对象四大特征之一，分为行为抽象和特征抽象。  讲师：那什么是抽象，如何给抽象下定义呢？ |  |
|  | **3、抽象的定义**  讲师：首先我们需要明确的是，类是通过抽象得到的结果。  讲师：因此抽象是一个过程，抽取的目标是共同特征和共同行为。  讲师：所以，抽象的定义就是从具体事物中抽出共同特征(属性)和行为(方法)的过程。  讲师：那么这个就是抽象的定义。 |  |
|  | **4、课堂编程**  讲师：好，大家通过编码来亲自实现一下，时间5分钟。 |  |

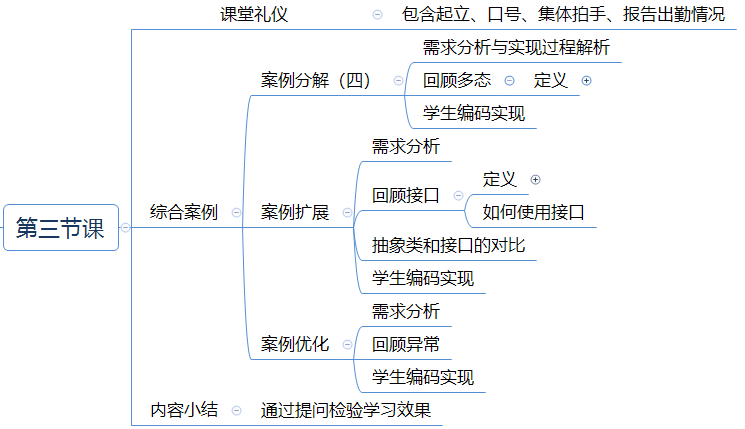
1. **脚本切片三：[视频切片](file:///C:\\Users\\李巍\\AppData\\Roaming\\Microsoft\\Word\\2、回顾和作业点评3’.mp4)3**

|  |  |  |
| --- | --- | --- |
| **课堂模块** | **教学内容/台词** | **教学标准/素材** |
| **案例分解（二）** | **1、案例分解（二）**  讲师：在生活中，每个人都有属于个人的私人物品。这些物品可以随便被他人使用吗？  学生：不能  讲师：不能，只有在征得主人同意的情况下，别人才可以使用。  讲师：那么在计算机当中也是如此。在之前创建员工类时，我们没有添加属性的访问修饰符。  讲师：这样会造成什么后果呢？  学生：【省略答案】  讲师：成员变量会被随意进行方法，造成数据不安全，可以随便进行修改。讲师：所以，我们就需要使用Java的封装机制，对成员进行封装，对代码进行保护。 |  |
| **2、封装员工类**  讲师：对员工类进行封装，我们看一下是如何实现的。  讲师：使用private关键字对成员变量修饰。  讲师：private关键字表示的是什么的意思呢？  学生：私有的。  讲师：没错，私有的成员，是无法从外部直接访问的。  讲师：那么用什么方法可以访问到私有成员呢？  学生：【省略答案】  讲师：一种方式是创建带参数的构造方法，通过构造方法访问私有成员。  讲师：另一种方法是设置公有的set方法和get方法。  讲师： set方法访问私有成员进行赋值。get方法就可以获取成员变量。。  讲师：这样，就实现对类的封装。  讲师：在访问时创建测试类，在测试类中，通过构造方法对成员变量进行初始化，然后调用方法输出结果。 |  |
| **3、封装概念图**  讲师：封装是面向对象的四大特征之一。  讲师：在实现封装时，可以对属性进行封装，也可以对方法进行封装，当对属性和方法都实现了封装后，就相当于是对类进行了封装。  讲师：封装需要通过访问修饰符来限制访问权限。  讲师：除了private以外，还有public、protected以及缺省三种。每一种都具有不同的访问权限。  讲师：属性的封装通过private修饰，并提供setter和getter接口来，从而实现对成员变量的封装。  讲师：此外当一个方法也被使用private来修饰时，这个方法就被封装了。  讲师：那什么是封装呢？我们来看一下封装的定义 |  |
|  | **4、封装**  讲师：封装是将属性和方法组合成类，同时隐藏类内部实现细节的机制。  讲师：那么封装分类包括属性的封装。  讲师：属性的封装就是修改成员变量的可见性，设置成私有的，对外提供setter和getter接口。  讲师：此外对方法的封装，提取出公共代码，让这个方法执行一件事。  讲师：最后是类的封装，类的封装主要是从规范角度出发，要体现出类的内聚性、一致性、封装性、清晰性和完整性。  **讲师：**内聚性就是类作为一个最基本的单元。  **讲师：**一致性就是类、成员变量、方法名要有序**。**  **讲师：**封装性对外提供访问接口，内部隐藏细节。  **讲师：**清晰性是指封装结构应该清晰、易于理解。  **讲师：**完整性是充分考虑多种应用场合，满足不同的应用需求 |  |
|  | **5、访问修饰符的级别**  讲师：Java中提供了四种访问修饰符，每个修饰符的权限又各不相同。  讲师：从列表中可以看到，每一个访问修饰符的权限。  讲师：其中private修饰符的权限最小，只能在本类中可见。  讲师：缺省也就是没有修饰符的权限，只能在本类和本包中可见。注意，缺省是没有对应关键字，有的同学会认为是default，这一点需要注意。  讲师：第三个是protected修饰符，它不仅可以是在本类和本包中访问，还能在子类中访问，权限进一步扩大。  讲师：最后一个就是public修饰符，表示公开的，公共的。权限最大，可以随时进行访问。  讲师：这四种修饰符在使用时，需要结合业务需求、代码实现方式有选择的使用。 |  |
|  | **6、课堂编程**  讲师：下面的时间给大家5分钟，将员工类进行封装，并完成员工信息的输出显示。 |  |

1. **脚本切片四：[视频切片](file:///C:\\Users\\李巍\\AppData\\Roaming\\Microsoft\\Word\\2、回顾和作业点评3’.mp4)4**

|  |  |  |
| --- | --- | --- |
| **课堂模块** | **教学内容/台词** | **教学标准/素材** |
| **案例分解（三）** | **1、案例分解（三）**  讲师：下面我们继续实现员工管理系统中的员工工资核算功能  讲师：员工根据岗位不同，工资收入也各有差异，在计算工资时，计算的公式也不同。  讲师：普通员工工资按照单日工资\*出勤天数进行统计。  讲师：司机的工资按照单日工资\*出勤天数+津贴统计。  讲师：经理的工资时按照年薪/12进行统计。  讲师：大家思考一下，该如何实现呢？  学生：【省略答案】。  讲师：很好，通过抽象类来完成。  讲师：为什么使用抽象类来完成呢？  学生：【省略回答】。  讲师：使用抽象类，将重复的代码提取出来，这样可以减少重复代码的编写工作量。  讲师：具体的实现思路是，抽象出员工类，并声明工资计算的方法为抽象方法。  讲师：用普通员工类、司机类和经理类继承抽象员工类，并重写工资计算方法。  讲师：编写测试类，测试不同岗位员工工资的计算方法。 |  |
| **2、代码分析**  讲师：具体的实现代码如下。  讲师：使用abstract关键字声明了一个抽象的员工类，AbstractEmployee。  讲师：虽然员工的岗位不同，但是具有的基本属性是一样的。所以可以将相同的属性都提取到AbstractEmployee类中作为公共属性。  讲师：然后，通过有参构造方法，对成员变量进行初始化赋值。  讲师：我们需要计算的是工资，因此还需要声明了一个抽象方法，用来获取工资计算的结果。  讲师：大家注意，在抽象类中既可以包含抽象方法，还可以有非抽象方法。  讲师：为了方便员工信息的输出，我们重写了toString()方法，这样当输出对象时就会自动调用toString（）。  讲师：这样一个抽象的员工类就完成。  讲师：有了抽象员工类，下面就需要让不同岗位的员工来进行继承。  讲师：首先我们让普通员工类继承抽象员工类类，由于日工资和出勤天不是所有岗位都有，因此这两个属性依然放在普通员工类中。  讲师：在初始化赋值时，首先是通过super关键字调用了父类的构造方法。  讲师：给共同的属性进行赋值，。  讲师：然后使用this关键字给子类中独有的属性进行赋值。  讲师：最后就是实现getSalary（）方法，按照普通员工的计算公式返回工资结果。  讲师：通过让普通员工类继承抽象员工类，减少重复代码的编写、程序结构更加清晰。  讲师：同理，下面需要让司机类也继承抽象员工类。  讲师：大家仔细观察一下司机类的实现代码。  讲师：有的同学发现了，司机类并没有直接继承抽象员工类，而是继承普通员工类，这是为什么？  学生：【省略答案】  讲师：原因在于，司机作为员工，他与普通员工类都具有日工资和出勤天数这两个属性，差别是多了一个津贴属性。  讲师：因此为了减少重复编码，实现代码复用，让司机类直接继承普通员工类。并再次重写了getSalary()方法。  讲师：最后是经理类，实现过程比较简单，直接继承抽象员工类，并且重写了getSalary()方法。  讲师：至此，就已经完成了对三个岗位员工计算工资的编码。下面通过测试类来查看一下运行效果。  讲师：在测试类中，分别创建了普通员工类、司机类和经理类的对象实例，并进行了成员初始化。  讲师：然后通过调用getName（）方法和getSalary（）方法，输出员工工资的计算结果。 |  |
| **3、继承概念图**  讲师：在实现计算工资的代码中，我们使用到了继承。  讲师：继承是面向对象特征之一，通过继承实现了最大程度的代码复用。继承的定义就是子类复用父类的属性和方法的机制  讲师：实现继承，需要使用extends关键字。包括单继承、多继承、多级继承、层次继承、混合继承。  讲师：继承中的父类和子类又是如何划分的呢？  学生【省略答案】。  讲师：被继承的类称为父类，也可以称为基类或者超类  讲师：子类则是由父类派生得来的，因此又称为是派生类。  讲师：既然是继承关系，那么子类可以调用父类中允许访问的成员，包括成员变量和成员方法。  讲师：使用super关键字可以实现对父类成员的调用，而使用this关键字则可以用调用本类的成员。  讲师：当子类继承父类后，可以对父类的方法进行重写，以满足子类的需要。  讲师：以上是继承相关知识的概念回顾。 |  |
|  | **4、重载与重写的比较**  讲师：当子类继承父类后，可以对父类中的方法进行重写，以满足子类的需求。  讲师：在封装的学习时，我们学习过方法的重载，典型应用就是构造方法重载。  讲师：那么大家回想一下，重载和重写有什么区别呢？  讲师：重写是发生在子类内，对父类的方法进行重新实现。重载是在一个类内部发生的，方法的不同实现。  讲师：不论是重写还是重载，执行的方法名称都必须一致。  讲师：对于方法参数而言，重写时，参数必须相同。而重载时参数不能相同。  讲师：重写时，返回值需要相同或者是其子类，重载时与返回值无关。  讲师：最后一点，重写时，重写方法的访问权限不能比父类更严格。重载时与访问修饰符无关。 |  |
|  | **4、课堂编程**  讲师：下面给大家15分钟的时间，使用继承的方式实现员工工资计算。 |  |

1. **教学过程脚本设计——第三课时**
2. **脚本设计思路**



1. **脚本切片设计思路**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **内容** | **目标** | **设计亮点** | **时间** |
| 1 | 课堂礼仪 |  |  |  |
| 2 | 案例分解（四） | 1. 通过案例分析回顾多态 | 1. 通过检查培养学生预习的习惯。 |  |
| 3 | 案例扩展 | 1. 通过案例扩展回顾接口 | 1. 通过案例对比演示。 2. 强调静态常量存在的不足。 |  |
| 4 | 案例优化 | 1. 通过案例优化回顾异常 | 1. 通过属+种差的方式得出定义。 |  |
| 5 | 小结 | 1. 掌握枚举的语法。 2. 能够正确的定义枚举类型。 3. 正确调用枚举类型。 | 1. 保证学生完成第一个案例的运行，提升学生学习的信心。 2. 演示的方式给学生完整感受。 |  |

1. **脚本切片一：**[**[视频切片1](file:///C:\\Users\\lenovo\\AppData\\Roaming\\Microsoft\\Word\\1、课堂礼仪1‘.mp4)**](file:///C:\Users\李巍\AppData\Roaming\Microsoft\Word\2、回顾和作业点评3’.mp4)

|  |  |  |
| --- | --- | --- |
| **课堂模块** | **教学内容/台词** | **教学标准/素材** |
| **课堂礼仪**  **60s** | 讲师：上课！  班长：起立!  讲师：同学们好。  学生：老师好。  讲师：我们班的口号是？  学生：…【集体拍手】。  讲师：检查工装。  班长：1 2 3 4。  讲师：报告出勤情况。  班长：应到\*人，实到\*人。  讲师：请坐。 | 1. 讲师进入教室前整理着装，提前30秒面带自信的笑容走进教室，给学生打招呼。当上课铃声响起，喊出“上课”，声音高于授课声音。 2. 学生起立、喊口号和拍手时，讲师需严格检查学生整齐度和速度，如凌乱需要重新做。 3. 口号结束后，讲师需跟学生一起有节奏的鼓掌（必须站在讲台前）。 4. 学生报告出勤情况时，讲师要面带微笑，并以肯定的眼神看着班长（报告出勤的同学）。 |

1. **脚本切片四：[视频切片](file:///C:\\Users\\李巍\\AppData\\Roaming\\Microsoft\\Word\\2、回顾和作业点评3’.mp4)4**

|  |  |  |
| --- | --- | --- |
| **课堂模块** | **教学内容/台词** | **教学标准/素材** |
| **案例分解（四）** | **1、案例分解（四）**  讲师：实现员工管理系统中工资汇总功能。  讲师：要求使用数组存放员工数据。  讲师：能够依次打印出所有员工姓名和当月工资，并返回工资总额。  讲师：那么大家考虑一下，这个数组能存储多个类型元素吗？  学生：【省略回答】  讲师：只能存储一种类型的数据元素，而员工根据岗位不同分为普通员工、司机和经理。  讲师：那如何才能将这三种不同类型的数据元素都保存在同一个数组中呢？  学生：【省略答案】  讲师：计算工资时，我们定义了一个抽象员工类，由不同岗位的员工去继承。  讲师：那么可以使用抽象员工类作为数组类型，然后将其子类作为数组元素进行存储。  讲师：具体的实现思路就是声明数组类型，存放子类对象。  讲师：将数组作为工资汇总的方法，通过遍历数组调用各自的计算工资方法。 |  |
| **2、代码分析**  讲师：那么具体的实现代码，我们来看一下。  讲师：首先定义一个财务管理类Finance类  讲师：在这个勒种有一个静态方法salarySummary()，这个方法的作用就是用来计算工资总额。  讲师：这个方法接收了一个什么类型的参数？  学生：【省略答案】  讲师：使用抽象员工类型的数组作为参数。通过遍历数组，得到每一个员工对象，然后调用改员工计算工资的方法，得到月工资。  讲师：将每次得到的工资进行求和累加，最后得到的就是所有员工的工资总额。  讲师：创建测试类，定义抽象员工类型的数组，并进行数组赋值。  讲师：由于在Finance类中，将计算工资总额的方法设置为static修饰，所以可以直接通过类名进行调用，并将赋值后的数组传进去。  讲师：通过运行结果可以看到，除了输出了每一个员工的工资外，还输出了工资总额。 |  |
| **3、多态概念图**  讲师：大家思考一下，在计算工资总额的方法中，使用的是抽象员工类型的数组，数组中存放的是具体实现类。  讲师：那么问题是，在遍历时为什么可以准确的知道其实现类是哪个呢？  学生：【省略答案】  讲师：没错，是因为多态。  讲师：下面通过概念图来回顾一下与多态相关的知识点。  讲师：多态是四大特征之一，也是面向对象的核心内容。多态分为编译时多态和运行时多态。  讲师：编译时多态是指在编译时就已经确定实现方式，例如方法重载，在编译时就知道总共有几个方法可以重载。  讲师：运行时多态是在运行时才能确定实现方式，我们总结过一个经典的语句是编译看左边，执行看右边。  讲师：运行时多态与继承联系紧密，例如普通类的继承，抽象类的继承、还有就是实现接口。  讲师：实际上在前面的代码中已经应用到了多态，知道在哪里吗？  学生：【省略答案】  讲师：在employees数组中，数组类型是AbstractEmployee类型。但是数组中存放的分别是Employee、Driver、Manager类型。  讲师：因为三个类型都是AbstractEmployee类型的子类，这就形成了一种运行时多态。  讲师：用父类的引用指向了子类对象，这种现象称为什么？  学生：向上转型。  讲师：非常好，那如果把已经向上转型的子类对象强制转换回子类类型是什么？  学生：向下转型。  讲师：非常好。运行时多态可以通过后期绑定来完成，后期绑定是指在程序运行时，根据所引用对象的实际类型来调用相应方法的方法绑定。  讲师：与后期绑定对应的是前期绑定，指在程序编译时便确定具体调用方法的方法绑定，例如构造方法的调用。  讲师：除了构造方法，属于前期绑定的还包括静态方法、私有方法，以及final方法。  讲师：以上就是与多态相关的主要概念回顾。 |  |
|  | **4、课堂编程**  讲师：下面给大家10分钟，完成工资总额计算的功能。 |  |

1. **脚本切片三：[视频切片3](file:///C:\\Users\\李巍\\AppData\\Roaming\\Microsoft\\Word\\2、回顾和作业点评3’.mp4)**

|  |  |  |
| --- | --- | --- |
| **课堂模块** | **教学内容/台词** | **教学标准/素材** |
| **案例扩展** | **1、需求描述**  讲师：完善司机类，以接口的形式规定司机必须要驾驶技能。  讲师：咱们同学有没有会开车，有驾照的？  学生：【省略回答】  讲师：有的同学已经考取了驾照，在实际生活中，驾照是一个司机驾驶能力的凭证。  讲师：规定了司机可以驾驶的车辆类型，例如C型驾照，只能驾驶小型车辆。  讲师：A2型驾照包含了C型驾照中允许的车辆、以及卡车。  讲师：A1驾照又包含了A2驾照允许的车型，并且还可以驾驶大客车。  讲师：因此，驾驶不同类型的车辆就需要考取不同类型的驾照。  讲师：那么按照对驾照类型的分析，这个需求如何实现呢？  讲师：可以通过接口，在接口中定义驾驶的方法。  讲师：由司机类实现接口，重写驾驶的方法。  讲师：具体的代码实现看一下。  讲师：定义一个接口，声明一个方法。  讲师：然后司机类在继承员工类的基础上，实现接口，并重写方法。  讲师：提到接口，大家思考一下接口和抽象类有那些相同点或不同点呢？  学生：【省略答案】  讲师：首先我们先看相同点。接口和抽象类中都可以有抽象方法，二者不能直接实例化，都可以用来体现多态的应用。  讲师：不同点表现在，首先对于接口而言，一个类可以实现多个接口，接口本身还可以继承接口，而抽象类只能是单继承。  讲师：其次，在接口中只能存放常量，而抽象类中既可以存放常量，也可以存放变量。  讲师：最后，接口中的方法可以包含抽象方法、静态方法和default方法，抽象类中则可以包含抽象方法、静态方法和普通方法。  讲师：以上就是对接口和抽象类的比较回顾。 |  |
|  | 1. **课堂编程**   讲师：下面给大家10分钟的时间来完成显示驾照类型的功能 |  |

1. **脚本切片三：[视频切片3](file:///C:\\Users\\李巍\\AppData\\Roaming\\Microsoft\\Word\\2、回顾和作业点评3’.mp4)**

|  |  |  |
| --- | --- | --- |
| **课堂模块** | **教学内容/台词** | **教学标准/素材** |
| **案例优化** | **1、需求描述**  讲师：大家已经基本完成了员工信息管理系统的主要功能。  讲师：那么大家思考一个问题。  讲师：员工的年龄有限制吗？工资可以是负数吗？如何进行检查和控制？  学生：【省略答案】  讲师：对于前两个问题，其实答案很明显，我国法律规定，60岁就可以退休了。如果一个公司的员工年龄有好几百岁，那大家想这个员工是什么？  讲师：至于工资，去除其特殊的原因，否则工资一定是一个大于0的数，不然，大家为什么还要工作。  讲师：回想之前的代码中，并没有对这些数据进行限制，那么有没有可能会出先类似的情况的呢？  学生：【省略答案】  讲师：存在隐患，那么就需要进行检查与控制。该如何处理呢？  学生：【省略答案】  讲师：非常好，使用异常处理机制  讲师：在Java中对程序运行中的不正常现象，分为异常（Exception）和错误（Error）两种情况  讲师：我们重点考虑的是Exception的处理。  讲师：异常可以分为系统异常和自定义异常。系统异常又分为了运行时异常和非运行时异常。  讲师：运行时异常是派生于RuntimeException类的，不需要显式处理的异常。  讲师：运行时异常的特点是在编译阶段不进行检查。  讲师：非运行时异常又称为检查型异常。  讲师：非运行时异常是派生于Exception(除RuntimeException及其子类)，需要显式处理的异常。  讲师：检查型异常的特点就是在编译阶段进行检查并强制要求进行处理。  讲师：在Java中使用5个关键字来进行异常处理，分别是，try、catch、finally、throw和throws。  讲师：try语句中执行可能产生异常的代码。  讲师：catch语句中进行捕获异常。  讲师：finally语句的代码无论是否发生异常代码总能执行  讲师：大家还记得如何使用try-catch-finally进行异常处理吗？  学生：【省略答案】。  讲师：在程序中try-catch-finally在使用中有三种不同的方式。  讲师：第一种是try-catch结构。  讲师：第二种是try-catch-finally结构。  讲师：第三种是try-finally结构。  讲师：通过仔细观察可以发现，在三种结构中都有try语句，但是可以没有catch语句，也可以没有finally语句。  讲师：这三种结构大家需要熟练掌握。  讲师：throws是在声明方法时，指定可能会抛出哪些异常的关键字。  讲师：throws关键字的位置，必须在方法声明后，指定声明抛出异常的类型。  讲师：使用throws的作用就是通知方法的调用者知道要捕获异常，并由调用者对异常进行处理。  讲师：使用throws声明异常，表示方法在执行时可能会抛出某种异常，但是不等于就一定会抛出异常。  讲师：以上就是异常相关的概念回顾。 |  |
|  | 1. **课堂编程**   讲师：下面给大家10分钟的时间来优化代码，添加异常处理。 |  |

1. **脚本切片六：[视频切片](file:///C:\\Users\\李巍\\AppData\\Roaming\\Microsoft\\Word\\2、回顾和作业点评3’.mp4)6**

|  |  |  |
| --- | --- | --- |
| **课堂模块** | **教学内容/台词** | **教学标准/素材** |
| **小结** | **1、小结**  讲师：通过综合案例，我们回顾了面向对象的核心内容，下面通过几个问题来做个小结。  讲师：第一个问题什么是面向对象？  学生：【省略答案】  讲师：第二个问题简述你对面向对象四大特征的理解  学生：【省略答案】  讲师：重载与重写的区别？  学生：【省略答案】  讲师：抽象类与接口的区别？  学生：【省略答案】 |  |

1. **教学过程脚本设计——第四课时**
2. **脚本设计思路**



1. **脚本切片设计思路**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **内容** | **目标** | **设计亮点** | **时间** |
| 1 | 本单元贯穿案例和知识目标 | 1. 讲解本单元的贯穿案例 2. 通过本单元概念图讲解明确学习目标 |  |  |
| 3 | 为什么使用枚举 | 理解使用枚举的原因 | 1. 结合案例代码讲解。 |  |
| 4 | 如何使用枚举 | 1. 掌握枚举的语法。 2. 能够正确的定义枚举类型。 3. 正确调用枚举类型。 | 1. 实践知识，学生为主，教师巡视辅导和答疑。 |  |
| 5 | 枚举的应用 | 1. 掌握枚举的最基本应用 | 1. 讲解枚举使用注意事项。 |  |
| 6 | 深入理解美军 | 1. 掌握枚举在switch中使用方法 | 1. 案例引入。 2. 通过代码展示应用 |  |
| 7 | 如何使用枚举 | 1. 掌握如何在枚举中添加成员 2. 枚举使用注意事项 | 1. 练习前进行思路讲解。 2. 练习中进行辅导答疑。 3. 练习后进行共性问题分析。 |  |
| 8 | 枚举的应用 | 1. 枚举在switch结构中应用 | 1. 案例引入。 2. 通过代码展示应用 |  |
|  | 什么是枚举 | 理解枚举的定义 |  |  |
|  | 总结 |  |  |  |

1. **脚本切片一：**

|  |  |  |
| --- | --- | --- |
| **课堂模块** | **教学内容/台词** | **教学标准/素材** |
| **本单元贯穿案例** | **1、本单元贯穿案例**  讲师：在正式学习枚举之前，我们来看一下本单元的贯穿案例  讲师：完成开心聊天室的主菜单功能。  讲师：使用枚举作为switch的参数完成分支的选择。  讲师：也就是根据菜单选择，用户可以执行相应的操作。执行效果如图所示。 |  |
| **本单元知识目标** | **2、知识目标**  讲师：那么既然需要使用枚举来完成聊天室的菜单功能。  讲师：我们就来看一下与枚举相关知识概念有那些。  讲师：在Java基础时，我们已经知道引用数据类型包括了数组、接口、类，还有就是枚举。  讲师：所以枚举也是一种引用数据类型。  讲师：枚举是继承自Enum类，Enum类又默认继承了Object类。  讲师：作为引用数据类型，在枚举中可以添加成员，例如方法和变量。  讲师：枚举的一个核心概念就是枚举常量，枚举常量相对于普通常量而言，数据更安全，代码更直观。  讲师：这就是枚举涉及的相关概念知识 |  |

1. **脚本切片三：**[视频切片3](file:///C:\\Users\\李巍\\AppData\\Roaming\\Microsoft\\Word\\2、回顾和作业点评3’.mp4)

|  |  |  |
| --- | --- | --- |
| **课堂模块** | **教学内容/台词** | **教学标准/素材** |
| **为什么使用枚举** | **1、枚举引入**  讲师：一年是有四季之分，如何用代码来体现呢？  讲师：定义一个接口，通过接口定义整型常量表示春夏秋冬，并且赋值为1234。  讲师：用一个类去实现接口。  讲师：这里定义私有属性season。  讲师：通过set方法，调用接口中的常量为属性赋值。  讲师：输出结果为1。  讲师：这句话好理解吗？  学生：【省略答案】  讲师：既然不容易理解，我们先用枚举来进行优化，看看枚举的使用效果。  讲师：第一步定义枚举类型SeasonEnum。  讲师：如何定义枚举后面会详细讲解，这里大家先看效果。  讲师：第二步是调用枚举并赋值。  讲师：通过枚举类型SeasonEnum.枚举常量进行赋值。  讲师：打印结果为：当前的季节是：SPRING。  讲师：这样更容易让人看懂和理解。  讲师：另一方面，大家看代码，由于在赋值时只规定了数据的类型，并没有限制数据的取值范围，从而导致最后的运行结果错误。  讲师：如果使用枚举来限制数据的输入范围，这样就可以提升代码的可靠性。  讲师：大家注意看，在类中定义了一个枚举类型的属性，在赋值时只能输入枚举所包含的内容，否则无法通过编译。  讲师：通过这一点，我们可以看到使用枚举后，数据类型与数值范围都收到了限制。  讲师：基于结果可以看到使用枚举后的效果，这也是为什么使用枚举的原因。 | 1. 通过静态常量引入。 2. 结合案例分析静态常量的问题。   C:\Users\sw\AppData\Local\Temp\ksohtml\wps25A5.tmp.jpg |

1. **脚本切片四：**

|  |  |  |
| --- | --- | --- |
| **课堂模块** | **教学内容/台词** | **教学标准/素材** |
| **如何使用枚举** | **1、枚举的语法**  讲师：声明枚举类型，首先是访问修饰符，然后是enum关键字，最后是枚举类型的名称。  讲师：在大括号内，是枚举常量的声明，每一个枚举常量之间用逗号分隔。  讲师：例如SeasonEnum枚举类型的声明；  讲师：public-访问修饰符，enum关键字，SeasonEnum枚举类型。  讲师：大括号内为枚举常量声明，也称为是枚举项。 |  |
|  | **2、获取枚举实例的方式**  讲师：声明枚举后，在使用枚举有三种方式。  讲师：第一种是通过枚举常量获取枚举实例。通过枚举类型调用枚举常量，返回结果就是一个枚举实例。  讲师：第二种是通过指定名称获取枚举实例。通过枚举类型调用valueOf（），并给出一个名称参数。  讲师：valueOf()方法会按照名称参数查找并返回一个与名称相同的枚举实例。  讲师：第三种是得到包含所有枚举实例的数组。通过枚举类型调用values()方法，就会返回一个枚举类型的数组，该数组中存放了所有枚举的实例。 |  |

1. **脚本切片五：视频切片5**

|  |  |  |
| --- | --- | --- |
| **课堂模块** | **教学内容/台词** | **教学标准/素材** |
| **枚举的基本应用**  **120s** | **1、枚举的基本应用**  讲师**：**掌握了如何使用枚举，下面通过案例看一下枚举的基本应用。  讲师**：**编码模拟红绿灯状态显示。  讲师：同学们红绿灯的状态是不是固定的？  学生：是。  讲师**：**所以可以使用枚举来进行表示。  讲师：先定义枚举类型，设置三个枚举常量值分别是red、yellow、green。  讲师：调用时，通过枚举类型访问枚举常量，获得枚举常量的值，然后输出结果。  讲师：这就是枚举的一个最基本的应用。 |  |
|  | **2、课堂编程**  讲师：使用枚举来表示一周7天，并输出显示。  讲师：在实现时，首先用enum关键字进行枚举声明。  讲师：将一周7天定义为枚举常量。  讲师：最后获取枚举类型调用枚举常量输出结果。  讲师：下面给大家5分钟的时间来编码实现 |  |

1. **脚本切片六：**

|  |  |  |
| --- | --- | --- |
| **课堂模块** | **教学内容/台词** | **教学标准/素材** |
| **深入理解枚举**  **180s** | **1、枚举的本质**  讲师：声明枚举类型后，会自动编译成一个final类。  讲师：并且默认继承Enum类，Enum类在后面给大家详细介绍。  讲师：在final类中枚举常量会被编译成public static final。  讲师：所以可以使用枚举类型直接调用。  讲师：枚举常量的类型与声明时枚举类型相同，即为本类型常量。  讲师：提供一个私有的构造器，所以不能在外部显示实例化。  讲师：因为编译时已经在内部进行了实例化。  讲师：即每一个枚举常量对应一个枚举实例。  讲话：所有枚举实例会按照声明顺序生成一个枚举类型的数组。  讲师：数组的下标即为声明枚举常量时默认的序号。 | 1. 展示编译后的源码，结合讲解。   C:\Users\sw\AppData\Local\Temp\ksohtml\wpsD3A6.tmp.jpg |

1. **脚本切片七：**

|  |  |  |
| --- | --- | --- |
| **课堂模块** | **教学内容/台词** | **教学标准/素材** |
| **了解Enum类** | **1、什么是Enum类**  讲师：枚举自动继承Enum类，那什么是Enum类？  讲师：首先Enum类是Java1.5后新增的类。  讲师：这个类也直接继承了Object类。  讲师：同时它也是一个抽象类。  讲师：它是作为所有枚举类型的公共父类。  讲师：在声明枚举类型的时，会自动继承该类。  讲师：对于Enum类，大家了解一下就可以了。 | 1. 学生了解即可 2. Enum的特点，final、不能被继承   C:\Users\sw\AppData\Local\Temp\ksohtml\wps8819.tmp.jpg |

1. **脚本切片八：**

|  |  |  |
| --- | --- | --- |
| **课堂模块** | **教学内容/台词** | **教学标准/素材** |
| **枚举中的成员**  **150s** | **1、枚举中添加变量和方法**  讲师：枚举中除了可以定义常量，还可以添加其他成员。  讲师：首先来看一下如何添加变量和方法。  讲师：由于需要在枚举添加其他的内容，所以在枚举常量后要以分号进行结束，这一点大家在编写代码时需要注意。  讲师：添加变量的方式很简单，与普通类中添加变量的方法相同，同学们还记得在类中定义变量吗？  学生：记得。【参考回答】  讲师：设置访问修饰符，数据类型，变量名。  讲师：将变量设置为private，设置public的set和get方法。  讲师：在实现时，获取枚举实例，通过实例得到枚举值后进行判断。  讲师：然后通过枚举实例来调用变量的set或者get方法，完成赋值和读取。  讲师：这就是如何在枚举中添加变量和方法。 |  |
| **2、构造器**  讲师：在枚举中可以添加构造器，也就是构造方法。  讲师：在枚举中，构造器只能是private，不能是public。  讲师：除了修饰符必须是private，其余方面与普通类中的构造方法的使用是一样的。  讲师：继续使用上一个案例，在定义常量时，分别给每一个常量都增加一个字符串说明，放在小括号中。  讲师：每一个常量的字符串说明是如何使用的呢？  学生：构造。  讲师：非常好，因为每一个常量都是一个实例，那么通过构造方法就可以实现对实例初始化。  讲师：在使用时，通过声明带参数的构造方法就可以将指定字符串传进去。  讲师：加入我们要是输出所有的枚举常量，那么在调用时如何实现呢？  讲师：调用values()方法，可以得到一个包含所有枚举实例的数组。  讲师：然后对这个数组进行遍历。  讲师：由于已经使用构造方法对枚举进行了初始化，所以输出时就会直接输出每个常量对应的字符串内容。 |  |
|  | **3、课堂编程**  讲师：手机店促销推出特价机，编码实现特价机的输出显示。  讲师：实现这个需求，思路分析如下。  讲师：店庆手机品牌固定，选择枚举来表示  讲师：手机价格作为枚举项参数，可定义一个变量进行存储  讲师：通过声明私有构造器实现变量初始化  讲师：通过自定义方法实现对应品牌手机的价格输出  讲师：下面给大家10分钟来完成. |  |

1. **脚本切片九：**

|  |  |  |
| --- | --- | --- |
| **课堂模块** | **教学内容/台词** | **教学标准/素材** |
| **枚举注意事项** | **1、枚举使用注意事项**  **讲师**：对于枚举，总结一下使用注意事项  **讲师**：首先，定义枚举常量时，枚举常量必须是在枚举的第一行进行定义。  **讲师**：其次，如果在枚举中继续添加成员，则需要在最后一个枚举常量定义完毕后，使用分号作为结束。  **讲师**：枚举中允许添加构造方法，但是构造方法只能是private。  **讲师**：枚举类型由系统编译并进行实例化，因此不能通过new关键字实例化。 |  |

1. **脚本切片十：**

|  |  |  |
| --- | --- | --- |
| **课堂模块** | **教学内容/台词** | **教学标准/素材** |
| **在switch结构中使用枚举** | 讲师：switch结构是用于进行选择判断的，使用与什么场合呢？  学生：等值判断。【参考答案】  讲师：下面我们就通过代码来看一下如何在switch结构中使用枚举。  讲师：依然是用枚举来实现对春夏秋冬季节的描述。  讲师：使用枚举定义春夏秋冬四个常量。  讲师：在使用时，将枚举类型的实例作为参数，传递给switch结构。  讲师：注意，在判断枚举实例时，每一个case语句后面是枚举常量名称。  讲师：在调用switch时，将枚举实例直接进行传递。 |  |
| **2、使用注意事项**  **讲师**：在switch结构中使用枚举时，需要注意。  **讲师**：每一个case标签，后面一定是枚举常量，不能是采用全限定名方式，否则无法通过编译。  **讲师**：大家看一下编译器是如何提示的。  讲师：编译器提示必须要使用一个常量。  讲师：所以在使用时，case语句后面一定是非限定名的常量名。 |  |
|  | **3、课堂编程**  讲师：使用switch结构，结合枚举实现生肖对应的英文单词输出。  讲师：声明带参数的枚举项，并添加变量。  讲师：通过构造器进行初始化。  讲师：在自定义方法中输出显示。  讲师：给大家10分钟来完成。 |  |

1. **脚本切片十一：**

|  |  |  |
| --- | --- | --- |
| **课堂模块** | **教学内容/台词** | **教学标准/素材** |
| **什么是枚举** | **1、枚举的作用**  讲师：通过枚举的应用，我们可以总结出来枚举的作用  讲师：第一个就是提高数据的可靠性，因为限制数据范围，所以更加可靠。  讲师：第二个是提高代码的可读性，代码更容易裂解，维护更加方便。 |  |
|  | **2、枚举的定义**  讲师：那么到目前为止枚举的定义是什么呢？  讲师：我们尝试用属+种差的方式来下个定义。  讲师：首先，大家思考一下属是什么？可以在看一下概念图。  学生：【省略答案】。  讲师：那么同属的种有那些？  学生：。。。。。  讲师：那尝试说下种差。  学生：。。。。。。  讲师：很好，总结一下种差是用enum关键字修饰，其次枚举常量类型就是枚举类型。  讲师：一个枚举常量就是一个实例，最后枚举是一个特殊的类。  讲师：那么综合起来，枚举的定义就是使用enum声明的，由一组预定义的本类型常量组成的引用数据理性。 |  |

1. **脚本切片十二：**

|  |  |  |
| --- | --- | --- |
| **课堂模块** | **教学内容/台词** | **教学标准/素材** |
| **单元总结** | 1. **内容小结**   讲师：现在对本单元内容进行总结。  讲师：枚举的定义是用enum声明，由一组预定义的本类型常量组成的引用数据类型。  讲师：内涵是enum关键字声明，常量类型与枚举类型相同，一个枚举常量就是一个枚举实例，枚举是一个特殊的类。  讲师：枚举的语法是访问修饰符、enum关键字，枚举名称。每一个枚举值之间用逗号分隔。  讲师：枚举在应用时，可以通过常量获取实例，也可以指定名称获取实例，还可以获取包含所有实例的数组。  讲师：枚举的作用就是提高数据可靠性和代码的可读性。 |  |