

河南中医药大学课堂教学设计

授课章节	第 11 章：多线程		授课学时	2 学时
所属课程	计算机程序设计	授课年级	2023 级	
设计者	阮晓龙、黄子杰	授课专业	智能医学工程专业	
1.教学目标：含知识、技能（能力）、学习态度与价值观（情感）目标				
知识目标： 1. 理解多进程与多线程； 2. 掌握 Java 多线程的编程方法； 3. 掌握线程资源的同步处理； 4. 掌握 Java 的任务定时处理。 能力目标： 1. 逻辑推导能力； 2. 语言表达能力； 3. 复杂问题简化分析能力。 素质目标： 1. 提升自主学习能力； 2. 激发创新思维与问题解决能力； 3. 强调团队合作与沟通能力； 4. 培养逻辑思维与清晰表达能力。 思政目标： 1. 弘扬科学精神，推动社会进步：通过学习 Java 编程语言，激发学生对科学技术的兴趣和追求，树立终身学习的理念，鼓励学生以科学的态度探索技术问题，积极参与到科技创新中去； 2. 培养学生的软件工匠精神，在潜移默化中培育社会主义核心价值观，提高综合职业素养，树立社会主义职业精神。				
2.教学内容：依据教学大纲；含教学重点难点				
教学重点： 1. 多进程与多线程； 2. 线程调度与优先级； 3. Java 多线程编程方法； 4. Thread 类实现多线程。 教学难点： 1. Java 多线程编程方法； 2. Runnable 接口编写多线程。				

课堂教学内容：

1、Java 线程的概念（25 分钟）

Java 多线程是一种使单个程序中的多个任务同时执行的机制。通过创建多个线程，可以充分利用系统资源，提高程序的执行效率。在 Java 中，线程可以通过继承 Thread 类或实现 Runnable 接口来创建。

（1）多进程与多线程

多进程和多线程是计算机并发执行任务的两种方式，进程是操作系统资源分配的基本单位，拥有独立的内存空间；线程是进程的执行单元，是 CPU 调度和执行的单位，多线程共享同一进程的资源。

（2）线程的状态

Java 线程在生命周期中可以处于五种状态：新建、就绪、运行、阻塞、终止。

（3）线程调度与优先级

线程调度是操作系统决定何时以及哪一个线程应当运行的过程，确保系统资源的有效利用和任务的合理分配；线程优先级是一种机制，用于向操作系统的调度器建议哪些线程应优先获得 CPU 时间，通过设置不同的优先级，可以影响线程的执行顺序，但并不保证执行顺序，还需考虑调度策略如时间片轮转或优先级调度。

2、Java 多线程编程方法（20 分钟）

Java 多线程编程方法概念涉及在 Java 语言中实现并发执行的技术，这包括创建和管理线程，使用线程池来优化资源利用，以及同步机制来控制对共享资源的访问，Java 提供了多种方式来实现多线程，如继承 Thread 类或实现 Runnable 接口。

（1）Thread 类简介

Thread 类是 Java 中的主要多线程类，它提供了创建和控制线程的方法，包括启动、运行和终止线程的机制，是实现并发执行的关键。

（2）继承 Thread 类实现多线程

通过继承 Thread 类，可以定义自己的线程类，并在其中重写 run 方法来指定线程执行的任务。

（3）实现 Runnable 接口编写多线程

实现 Runnable 接口是另一种创建线程的方法，通过定义一个类实现 Runnable 接口，并重写其 run 方法，可以创建一个线程任务。

3、线程资源的同步处理（20 分钟）

线程资源的同步处理是指在多线程环境下，对共享资源进行访问控制的一种机制，这种机制可以确保在同一时间只有一个线程能够访问共享资源，从而避免数据不一致的问题，同步处理可以通过互斥锁、信号量等技术实现。

（1）临界资源问题

在多线程环境中，临界资源是那些可能同时被多个线程访问和修改的共享资源，若未妥善处理，这些资源可能会引发数据不一致或竞态条件的问题。

（2）wait()和 notify()方法

wait()和 notify()是用于线程间通信的两种方法。

wait()使当前线程等待，直到另一个线程调用 notify()或 notifyAll()方法来唤醒它。

4、Java 的任务定时处理（15 分钟）

Java 提供了多种方式进行任务的定时处理，例如 Timer 类和 ScheduledExecutorService 接口，可以用来在未来某个时间或以固定间隔重复执行任务。

课堂教学内容：

3.思政知识点：

课程思政案例	思政点映射
物流公司使用的物流配送系统同时处理包裹分拣、运输调度、车辆监控等任务。如分拣中心多个分拣线程并行处理包裹，运输调度线程根据路况和订单分配车辆，车辆监控线程实时跟踪车辆位置。各线程协同确保包裹快速准确送达	培养效率意识、增强责任担当

4.学情分析及教学预测：	
学生的知识基础： 1. 编程基础； 学生的认知特点： 1. 对多进程和多线程没有清晰的理解。 学生的学习风格： 1. 不熟悉 Java 程序的基本知识，但学习的热情和积极性较高； 2. 不理解多线程的基本概念与原理，对计算机程序设计课程热情高。 教学预测： 1. 通过案例式教学和探究式教学等方法，培养学生的创新意识和思维能力； 2. 学生的学习兴趣 and 动机提升：通过引导学生进行实际的操作和互动交流； 3. 学生的合作与沟通能力培养：在课程中鼓励学生进行小组合作，分享经验和解决问题。	
5.教学策略与方法：	
教学策略： 1. 通过多媒体演示文稿进行讲解，并结合板书进行关键难点的介绍和原理过程的讲解； 2. 课后留练习题或作业，引导学生对课程内容进一步巩固和复习。 教学方法： 1. 通过课前预习，让学生对相关基础知识及概念有基本的了解。 2. 理论课通过讲解、与学生互动了解学生知识掌握情况，对学生较为薄弱的环节进一步强化介绍。	
6.板书设计：	
① 黑板（白板）设计： 多进程与多线程 线程调度与优先级 Thread 类实现多线程 Runnable 接口编写多线程	② 现代信息媒体设计： 使用多媒体教学课件开展。
7.教学互动环节设计：	
课堂上的提问和互动交流： 1. 问题一：多进程和多线程有哪些区别？ 2. 问题二：什么是线程调度？ 3. 问题三：Thread 类的功能是什么？	

8.学习资源，课外自主学习设计：

自建学习资源：

1. 课程学习平台：<https://webdev.hactcm.edu.cn/program>
2. 课堂派：<https://www.ketangpai.com>

网络学习资源：

1. Java 官网：<https://www.java.com/zh-CN/>

9.教学测量与评价：

课堂教学测量评价：

1. 课堂测试：使用课堂派开展阶段性测试。
2. 课堂提问：通过提问及利用课堂派与学生互动，及时了解学生知识点掌握情况。

课外学习测量评价：

1. 课前预习：通过课程学习平台开展预习。
2. 课后作业：通过课堂派布置作业，每个章节 1 个作业，内容见课堂派。

10.教学反思与改进：（授课后教师总结）

11.授课教师认为尚未包含在内的设计内容：（授课后教师总结）