**《Java程序设计》课程教学大纲**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 英文课程名 | Java Programming | | 总 学 时 | | 48 | 学 分 | 3 |
| 课程编码 | G126007 | | 理论教学学时 | | 32 | 适用专业 | 软件工程 |
| 课程类别（请在课程所属类别栏注明选修或必修） | 通识课程 |  | 实践  教学  学时 | 实验学时 | 0 | 先修课程 | 程序设计基础C和C++程序设计 |
| 大类基础课程 |  | 上机学时 | 16 | 开课学院（部） | 计算机学院 |
| 专业课程 | 必修课 | 其它 |  | 基层教学组织 | JAVA系列课程群教学团队 |

**一、课程简介**

本课程是一门软件工程专业的本科生专业必修课，主要讲授Java程序设计开发过程及工作原理，Java语法基础，面向对象程序设计，数组和字符串、枚举，Java常用类和接口，异常处理，流和文件，图形用户界面编程，以及多线程等内容。该课程要求学生理解Java语言的语法、语义规则；掌握Java语言的数据类型、表达式及控制流程；运用面向对象程序设计思想，能阅读、分析程序和设计程序，掌握面向对象程序设计方法，提供程序设计能力，以适应计算机软件开发不断发展的需求；学会使用数组与字符串，掌握泛型、异常处理、输入输出流、图形用户界面、多线程等Java编程方法，能够运用这些知识开发一些应用程序和小型项目，从而掌握用Java开发语言解决复杂软件工程问题。从课程地位上来说，该课程是后续Web应用开发、JavaEE技术、移动应用开发等课程的先修课程，为学生学好后续课程尤其是掌握Web应用开发能力打下基础。

**二、教学目标**

**2.1 课程教学目标**

（1）了解Java程序设计语言的历史与特点，掌握Java程序设计语言的基本语法，熟悉各种Java开发环境及调试技巧，从而了解Java的使用方法，以及实际软件工程实践中Java开发工具的使用现状。

（2）理解面向对象程序设计的基本概念和设计思想，能够运用这些知识分析复杂的软件工程问题，从而掌握解决复杂软件工程问题的基本原理。

（3）掌握Java 常用标准类库中Object、Math、System、Random等类使用方法，学会使用数组与字符串，掌握泛型、异常处理、输入输出流、图形用户界面等Java编程方法，能够运用这些知识开发一些应用程序和小型项目，从而掌握用Java开发语言解决复杂软件工程问题。

（4）理解多线程的概念，掌握Java开发多线程程序的方法，并运用这些知识解决一些实际应用中的并发问题，从而掌握用Java开发语言解决复杂软件工程问题。

**2.2 课程目标与毕业要求（指标点）对应关系**

本课程支撑以下毕业要求和具体细分指标点：

**【毕业要求3】设计/开发解决方案**：能够设计针对复杂工程问题的解决方案，设计满足特定需求的系统、单元（部件）或工艺流程，并能够在设计环节中体现创新意识，考虑社会、健康、安全、法律、文化以及环境等因素。

支撑指标点3.1：掌握解决复杂软件工程问题的基本原理和相关开发语言。

**【毕业要求5】使用现代工具**：能够针对复杂工程问题，开发、选择与使用恰当的技术、资源、现代工程工具和信息技术工具，包括对复杂工程问题的预测与模拟，并能够理解其局限性。

支撑指标点5.1：了解现代工程工具和信息技术工具的使用方法，了解实际软件工程系统中各种现代工具的使用现状。

本课程目标与毕业要求（指标点）的对应关系如表1所示。

表1 课程目标与毕业要求（指标点）的对应关系

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 课程目标 | 毕业要求指标点 | 教学环节 | | | |
| 课堂授课 | 上机 | 作业 | 课堂讨论 |
| **目标1：**了解Java 程序设计语言的历史与特点，掌握Java程序设计语言的基本语法，熟悉各种Java开发环境及调试技巧，从而了解Java的使用方法，以及实际软件工程实践中Java开发工具的使用现状。 | **指标点5.1**：了解现代工程工具和信息技术工具的使用方法，了解实际软件工程系统中各种现代工具的使用现状。 | √ | √ | √ |  |
| **目标2**：理解面向对象程序设计的基本概念和设计思想，能够运用这些知识分析复杂的软件工程问题，从而掌握解决复杂软件工程问题的基本原理。 | **指标点3.1**：掌握解决复杂软件工程问题的基本原理和相关开发语言。 | √ | √ | √ | √ |
| **目标3**：掌握Java 常用标准类库中Object、Math、System、Random等类使用方法，学会使用数组与字符串，掌握泛型、异常处理、输入输出流、图形用户界面等Java编程方法，能够运用这些知识开发一些应用程序和小型项目，从而掌握用JAVA开发语言解决复杂软件工程问题。 | **指标点3.1**：掌握解决复杂软件工程问题的基本原理和相关开发语言。 | √ | √ | √ |  |
| **目标4**：理解多线程的概念，掌握Java开发多线程程序的方法，并运用这些知识解决一些实际应用中的并发问题，从而掌握用Java开发语言解决复杂软件工程问题。 | **指标点3.1**：掌握解决复杂软件工程问题的基本原理和相关开发语言。 | √ | √ | √ |  |

**三、课程教学内容及学时分配**

**1．理论教学安排**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 章节或知识点(模块) | 教学内容 | 学时分配 | 教学要求  (应明确教学重点、难点和教学方法) | 学生任务 | |
| 作业要求 | 其他要求  (自学/讨论） |
| 1 | Java程序设计基础 | 1.Java的历史和主要特征；  2.Java程序开发过程及工作原理  3.Java语法基础：基本数据类型和类型转换，常量和变量，运算符，分支语句，循环语句，跳转语句 | 6 | 教学要求：  1.要求学生了解Java的历史和主要特征，掌握学生理解Java程序的工作原理和基本语法，了解与C++的区别；  2.要求学生学会搭建Java的开发和运行环境，学会使用Java的开发工具，能简单编写JAVA的应用程。  教学重点：  Java的特点，Java程序的执行工作和工作原理，开发工具，基本数据类型和类型转换，常量和变量，运算符与表达式，程序流程控制。  教学难点：  Java程序的工作原理，数据类型转换，程序流程控制语句。  教学方法：讲授+讨论+代码演示。 | 1.搭建Java的开发和运行环境  2.编写程序流程控制的程序，如编写一个所有课程的成绩等级和平均成绩 | 讨论：  Java和C++在语法上的区别 |
| 2 | 面向对象程序设计 | 1.面向对象的基本概念以及面向对象程序设计的基本思想  2.类的定义、成员变量与成员方法，构造方法和main方法  3.对象的生成与使用，变量的作用域，对象的内存分配机制，对象参数的传递  4.类的封装性、继承性、多态性和接口以及抽象类 | 8 | 教学要求：  1.要求学生了解面向对象的基本概念和程序设计思想，学会用面向对象的程序设计思想分析实际的软件工程问题能力；  2.要求学生掌握类的封装性、继承性、多态性和接口以及抽象类，学会用JAVA语言编写用面向对象的程序设计思想设计的程序。  教学重点：  类的定义，对象的创建，构造方法的定义与使用，方法的重载与覆盖，类的继承，包，抽象类和接口。  教学难点：  类与对象的概念、创建和使用，类的继承与运行时多态，访问控制权限的使用，抽象类的应用，接口的应用。  教学方法：讲授 + 讨论+ 任务驱动式教学+代码演示。 | 1.用面向对象的程序设计思想分析实际应用问题，并设计相关的类，如学生成绩查询系统项目分析，包括确定对象、测试对象、为对象确定属性和操作、为对象建模等  2.采用类和对象的特性实现相关的代码解决实际问题 | 讨论：  用面向对象程序设计的基本思想如何分析实际的工程问题  自学：  撰写系统需求分析文档的要求 |
| 3 | 数组和字符串、枚举 | 1.数组的应用  2.String和StringBuffer的构造方法和常见方法的使用  3.枚举 | 2 | 教学要求：  1.要求学生掌握数组和枚举类型的应用；  2.掌握String和StringBuffer使用的方法和区别，学会用Java语言解决软件项目中的数组和字符串处理问。  教学重点：  一维数组，String和StringBuffer的构造方法和常见方法的使用。  教学难点：  String和StringBuffer的使用方法的区别。  教学方法：讲授 +任务驱动式教学+代码演示。 | 1.熟练运用数组和字符串解决一些实际应用问题，如统计学生的平均成绩和各等级学生人数 | 自学：  字符串处理的常见方法 |
| 4 | Java常用类和接口 | 1.Object类、Math类、System类、Date类、Calendar类和Random类的使用  2.集合类和接口的使用  3.for语句的简化写法 | 2 | 教学要求：  1.要求学生掌握Object、Math、System、Random等类的应用；  2.掌握集合类List、ArrayList、Vector、Map的使用，学会使用集合类对软件工程中的数据进行存储和访问。  教学重点：  常见的集合类和接口，for语句的简化写法。  教学难点：  集合类和接口，如List、ArrayList和Vector、Map的熟练应用。  教学方法：讲授 + 代码演示 + 自主学习。 | 1.熟练应用集合类和接口如List、ArrayList和Vector、Map编程解决实际应用问题，如输入学生的学号、姓名、课程成绩，并将成绩按高到低排序打印输出。 | 自学：  集合类的使用方法，查阅资料自学其它相关内容 |
| 5 | 异常处理 | 1.Java中的异常类  2.异常处理模式try-catch-finally  3.重新抛出异常。 | 2 | 教学要求：  1.要求学生了解异常的处理机制，掌握如何编写自定义异常类；  2.学会对软件工程中对程序异常情况的处理  教学重点：  异常的基本概念，异常的抛出捕获和处理，自定义异常类。  教学难点：  异常的处理机制，自定义异常类。  教学方法：讲授 + 代码演示。 | 1.编写Java程序时能熟练运用异常处理机制 | 自学：  网上查询异常处理的应用 |
| 6 | 流和文件 | 1.流的基本概念  2.字节流和字符流  3.文件类File，随机访问文件类RandomAccessFile  4.对象序列化（可选讲） | 4 | 教学要求：  1.要求学生掌握字节流和字符流读写文件，掌握随机读写文件；  2.学会对软件工程中对文件数据的读、写操作。  教学重点：  输入输出流的基本概念，文件与目录管理，文件输入流和文件输出流，随机存取文件。  教学难点：  数据输入流与数据输出流，随机读写文件。  教学方法：讲授 + 案例式教学 + 代码演示。 | 1.采用文件输入输出流实现简单的读写文件，如输入学生成绩（包括学号、姓名和成绩），并将成绩写入一个文件中  2.采用文件类对目录进行处理  3.采用输入输出流解决一些复杂的应用 | 讨论：  如何实现某个工程下有多少行有效代码 |
| 7 | 图形用户界面编程 | 1.容器组件  2.基本组件  3.布局管理器  4.事件处理机制 | 4 | 教学要求：  1.要求学生熟悉常见的容器组件和基本组件以及布局管理器，理解事件处理模型;  2.学会用图形用户界面编程实现一些软件程序，从而掌握用JAVA开发语言解决复杂软件工程问题。  教学重点：  常用的GUI组件和容器，布局管理器，事件处理机制  教学难点：  布局管理器的灵活应用，事件处理机制。  教学方法：讲授 + 案例式教学 +代码演示。 | 1.采用swing组件设计图像用户界面程序  2.综合运用已学的内容设计复杂的界面交互程序，如实现学生成绩查询系统基本功能，包括用户登录、教师成绩输入与统计和学生成绩查询 | 讨论：  如何设计合理的图形用户界面 |
| 8 | 多线程 | 1.进程与线程  2.线程的生命周期  3.编写线程的方法：继承Thread类和实现Runnable接口  4.线程的基本控制方法  5.线程互斥与同步 | 4 | 教学要求：  1.要求学生掌握线程的运行机制和生命周期，掌握编写线程的方法和线程同步处理的方法，并运用这些知识解决一些实际应用中的并发问题，从而；  2.掌握用JAVA开发语言解决复杂软件工程问题。  教学重点：  线程的创建、线程的控制与调度，多线程同步问题。  教学难点：  多线程同步问题和处理方法。  教学方法：讲授 + 案例式教学 + 代码演示。 | 1.采用2种实现线程的方法设计多线程程序  2.利用同步处理的机制设计多线程程序解决并发问题，如模拟实现解决生产者和消费者问题 | 讨论：  如何用同步处理的方法解决多线程产生数据不一致问题 |

2．实践教学安排

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 项目名称 | 学时 | 类型 | 每组人数 | 教学要求  (应明确教学重点、难点和教学方法) | 学生任务 | |
| 作业要求 | 其他要求(自学/讨论） |
| 1 | Java结构化程序设计 | 2 | 设计型 | 1 | 教学要求：  熟悉Java开发环境和Java的基本语法，并根据某特定需求设计并实现结构化程序。  教学重点：  Java开发环境和工具，程序流程控制。  教学难点：  程序流程控制语句。  教学方法：代码演示 | 撰写作业报告，内容包括题目、问题分析（如思路、流程图、实现步骤等）、实现的代码、运行结果以及实验体会（如编写过程中出现的问题及解决方法） | 自学：自行下载JDK、Eclipse |
| 2 | 编写一个典型的Java类 | 2 | 设计型 | 1 | 教学要求：  根据某特定需求设计并实现一个Java类，掌握设计Java类的基本方法。  教学重点：  类的定义，对象的创建，构造方法定义与使用  教学难点：  类与对象的关系，对象的创建和使用。  教学方法：代码演示。 | 自学：  撰写软件需求分析文档的要求 |
| 3 | 类的继承与多态、访问权限的控制 | 2 | 设计型 | 1 | 教学要求：  运用面向对象的专业知识，对复杂软件工程问题进行面向对象设计，将复杂问题形式化描述，并实现相关的Java程序。  教学重点：  方法的重载与覆盖，类的继承，包，抽象类和接口。  教学难点：  类的继承与运行时多态，访问控制权限的使用，抽象类和接口的应用。  教学方法：代码演示。 | 自学：  查询如何采用面向对象的程序设计思想解决实际问题 |
| 4 | 数组、字符串处理与Java常见类及接口应用程序的开发 | 2 | 设计型 | 1 | 教学要求：  运用数组、字符串以及合集类，根据某特定需求设计实现一个字符串处理的程序。  教学重点：  一维数组，String方法、List、ArrayList和Vector、Map的应用。  教学难点：  字符串的处理，List、ArrayList和Vector、Map的熟练应用。  教学方法：代码演示。 | 自学：  1.查询处理字符串的常规方法及应用。  2.查询如何用集合类和接口。 |
| 5 | 异常处理程序设计 | 2 | 设计型 | 1 | 教学要求：  根据某特定需求设计并实现一个Java异常类，掌握如何处理程序异常的方法。  教学重点：  异常的抛出捕获和处理，自定义异常类。  教学难点：  自定义异常类。  教学方法：代码演示 | 自学：  baidu中搜索如何处理异常。 |
| 6 | 输入输出程序设计 | 2 | 设计型 | 1 | 教学要求：  根据某特定需求设计读写文件处理的程序，如文件拷贝，将键盘输入的数据经处理后存入文件。  教学重点：  输入输出流的基本概念，文件与目录管理，文件输入流和文件输出流。  教学难点：  数据输入流与数据输出流，随机读写文件。  教学方法：代码演示。 | 自学：  查询读写文件的方法。  讨论：  如何实现读写文本型文件（中文不乱码）和二进制文件 |
| 7 | 图形用户界面程序设计 | 2 | 设计型 | 1 | 教学要求：  根据某特定需求设计图形用户界面程序，如用户登录程序，掌握用户界面的实现方法。  教学重点：  常用的GUI组件和容器，布局管理器，事件处理机制。  教学难点：  事件处理机制。  教学方法：代码演示。 | 自学：  1.在baidu中搜索如何合理地设计图形用户界面  2.自学表格JTable和菜单JMenu、JMenuBar等组件设计复杂图像界面 |
| 8 | 多线程程序设计 | 2 | 设计型 | 1 | 教学要求：  根据某特定需求设计多线程程序，实现多线程的处理过程。  教学重点：  线程的创建、线程的控制与调度，多线程同步问题。  教学难点：  多线程同步问题和处理方法。  教学方法：代码演示。 | 自学：  1.查询进程和线程的概念；线程死锁的原因  2.基于TCP的socket多线程编程 |

**四、考核方式及成绩评定方式**

该课程的考核强调过程性考核。其总成绩分为平时成绩和期末考核成绩两部分，分别占50%。

期末考试占总成绩的50%。试卷难度适中，选择、判断、名词解释、简答等基础性题目的比例不超过30%，加大程序阅读题和编程题等综合性题目的比例，重在考查学生运用Java程序设计语言和面向对象程序设计思想的知识分析和解决复杂工程问题的能力。

平时成绩采用进程性考核，主要考核学生的作业（或上机实验）、课内测试、课程报告等几个方面。各部分所占的考核比例及基本要求如下：

（1）作业（或上机实验）：占总成绩的40%。要求：教师针对某些知识模块布置一定数量的课后作业或课外编程题，也可以将上机实验的内容进行拓展布置给学生作为课后作业，以巩固知识或拓展思维。对于作业中的共性问题，教师须在课堂讲解，以帮助学生提高和进步。

（2）课内测试：占总成绩的5%。要求：教师根据课程进度设计一些课内测试题，要求学生当场完成提交，以了解学生的学习情况。对于测试中存在的问题，教师必须课堂讲解，以帮助学生理解相关内容。

（3）课程报告：占总成绩的5%。要求：教师可以设计一个或若干个主题，要求学生查找和阅读资料，按要求完成一份课程报告。

期末考核采用闭卷或“一页开卷”考试的形式，一些难以记忆的类、接口和方法等学生可借助一页开卷纸进行记录，以便能更真实地考核学生的实际水平。期末考核主要考查学生掌握基础知识的程度和程序设计方法的能力，将按照本课程的教学目标全面考核学生课程学习的效果，分析课程对毕业要求的达成度。

**五、教材、课程网址及参考书目**

教 材：《Java程序设计实用教程》，高飞，清华大学出版社，2013.6，第1版

课程在学校网络教学平台的地址：http://mooc1.chaoxing.com/course/92750019.html

参考书：

【1】Java程序设计(第2版)，马世霞，机械工业出版社，2014.1

【2】Java程序设计，朱庆生，古平，清华大学出版社，2011

【3】Java语言程序设计（第3版），郎波，清华大学出版社，2016.7

【4】面向对象与Java程序设计(第2版)，朱福喜，清华大学出版社，2015.7

**执笔者：赵小敏**

**审核者：田贤忠**

**课程教学团队成员：黄洪、高飞、赵小敏、叶蕾、戴小春、陆佳炜、王春平、周乾伟、程珍、徐利锋、沈瑛等**