

怀化学院专升本考试

《Java 语言程序设计》课程考试大纲

一、考试目的

Java 语言程序设计考试旨在全面评估考生对于 Java 语言及其平台(包括 JavaSE)的基础知识和应用技能的掌握程度。此考试不仅测试考生对 Java 语言本身的了解，还注重考查其在面向对象程序设计、类库使用、程序分析与编写以及复杂工程问题解决等多方面的实际能力。

二、考试要求

考生在完成本课程学习后，应达到以下知识和技能两方面的考试要求：

知识要求：

1. 熟练掌握 Java 语言的基础语法和编程规范，能够准确使用 Java 语言的各种语法成分。
2. 深入理解面向对象程序设计的基本概念，包括类、对象、封装、继承和多态等。
3. 熟悉 Java 标准类库中的常用类及其功能，能够在实际编程中灵活运用。
4. 了解 Java 多线程编程的基本概念和方法。

技能要求：

1. 能够独立阅读、分析和理解简单的 Java 应用程序，包括其程序结构、逻辑和代码实现。
2. 具备良好的编程实践能力，能够编写和调试具有一定复杂度的 Java 程序。
3. 熟练掌握至少一种 Java 程序的编译和调试环境，如 Eclipse 或 IntelliJ IDEA 等，能够高效地进行程序调试和问题排查。

4. 具备使用 Java 语言分析和解决复杂工程问题的能力，能够运用所学知识进行实际项目的开发和实施。

三、考核知识范围及考核要求

(一) 基础知识

【考试要求】

了解 Java 语言的历史和特点；掌握 Java 的程序结构及开发环境的搭建；熟悉 Java 语言的各种数据类型；掌握变量和常量的定义方法；掌握基本数据类型的转换；掌握运算符与表达式的运用；熟悉 Java 的编码规范。

【主要考点】

1. Java 语言的历史、现状、特点、基本程序结构以及编程环境的搭建
2. Java 语言的数据类型：(1)基本数据类型：byte、short、int、long、float、double、char、boolean；(2)引用数据类型：类、接口、数组
3. 常量和变量：常量的各种表示形式和类型，变量的定义和初始化
4. 数据类型转换：显示转换，隐式转换
5. 运算符：赋值运算符、算术运算符、自增/减运算符、关系运算符、逻辑运算符、位运算符、条件运算符等
6. 表达式：表达式的组成规则，表达式运算中的类型转换、优先级和结合性
7. Java 编程规范

(二) 流程控制结构

【考试要求】

熟悉流程控制分为顺序结构、选择结构和循环结构，掌握选择结构和循环结构的具体实现。

【主要考点】

1. Java 程序的基本结构

2. 选择结构：if 语句、switch 语句
3. 循环结构：for 语句、while 语句、do-while 语句
4. 跳转语句：break 语句、continue 语句、return 语句
5. 循环语句与分支语句的嵌套

(三) 面向对象程序设计

【考试要求】

掌握面向对象程序设计的特点和基本概念；掌握类的声明；掌握对象的创建和使用；掌握继承和多态的概念和应用；掌握接口的声明和实现；掌握包的声明、包与类的引入；掌握 Java 类库常用类的使用。

【主要考点】

1. 面向对象程序设计的基本概念：类、对象、成员变量、成员方法、封装、继承、多态
2. 类的基本组成：类的声明、成员变量的声明、成员方法的声明与调用（参数传递与返回值）、方法的重载、静态变量和静态方法
3. 对象的创建和使用：对象的创建、构造方法与对象的初始化、成员变量/方法的访问、this 关键字、引用的赋值
4. 类的继承：子类的声明、super 关键字、变量覆盖和方法覆盖、final 类和 final 方法
5. 类及成员的四种访问权限：类（默认、public）；类成员（public、private、protected、默认）
6. 抽象类与接口：abstract 类和 abstract 方法的声明、接口的声明与实现
7. 包：包的声明及 package 语句、包和类的引入及 import 语句
8. 常用类的使用：String 类、StringBuffer 类、Math 类、Random 类、基本类型包装类、Object 类、Class 类

(四) 数组

【考试要求】

掌握一维数组、二维数组的声明、创建与使用。

【主要考点】

1. 数组的声明、创建和初始化
2. 数组的大小和元素默认值
3. 数组元素的使用

(五) 图形用户界面程序设计

【考试要求】

掌握 AWT 和 Swing 常用组件与容器的使用方法；掌握窗口应用程序的构建方法；掌握常用布局管理器的使用方法；掌握常用事件的处理方法。

【主要考点】

1. 常用组件和容器：标签、文本框、按钮、文本区域、复选框、单选钮、下拉列表、窗口、面板、JFileChooser、JColorChooser 等
2. 常用布局管理器：FlowLayout、BorderLayout、GridLayout、CardLayout
3. 事件处理：事件模型（事件、事件源、事件处理者）、WindowEvent、ActionEvent、KeyEvent、MouseEvent 等常见事件处理

(六) 异常处理

【考试要求】

理解异常处理机制；熟悉常用异常类；掌握用 try-catch-finally 语句捕获异常；掌握抛出异常和声明抛出异常；掌握自定义异常类的声明和使用。

【主要考点】

1. 异常的概念和分类

2. 常用的异常类
3. 异常的捕获与处理：try-catch-finally 语句
4. 异常的抛出：throw 语句、throws 子句
5. 自定义异常类的声明和使用

(七) 输入输出流及文件操作

【考试要求】

理解输入输出流的概念；熟悉流的分类；掌握常用流类的使用方法；掌握系统标准输入和输出的使用方法；掌握用 Scanner 类输入数据的方法；掌握用 File 类管理文件和目录的方法；掌握用 RandomAccessFile 类读写随机访问文件的方法。

【主要考点】

1. 流的分类：输入/输出流、字节/字符流、节点/过滤流
2. 常用流类：FileInputStream/OutputStream、BufferedInputStream/OutputStream、FileReader/Writer、BufferedReader/Writer 等
3. 系统标准输入输出操作
4. File 类：文件/文件夹的创建、删除、属性获取、重命名、目录列表
5. RandomAccessFile 类的读写操作

(八) 高级编程基础

【考试要求】

理解多线程程序设计的概念；掌握线程的创建、生命周期、调度和控制；了解线程的同步。

【主要考点】

1. 多线程的基本概念

2. 线程的创建方式
3. 线程的生命周期、调度和基本控制方法
4. 线程的同步机制

四、考试形式与试卷结构

1. 试卷分数及考试时间：满分为 200 分，考试时间为 150 分钟。
2. 答题方式：闭卷、笔试。
3. 试卷题型结构：

题型	题量	分值
单项选择题	30	60
程序分析题	10	40
应用编程题	5	100
总计	45	200

五、主要参考书目

1. 《Java 基础入门》，黑马程序员著，2022 年（第三版），清华大学出版社。

六、考试样题

单项选择题

1. 以下代码的输出结果是（ ）

```
int a = 10;  
a *= a;  
a -= 100;  
a += -90;  
System.out.println(a);
```

A. -100 B. -90 C. 10 D. 100
2. 下面关于数组声明和初始化的语句哪个有语法错误（ ）

- A. int a1[]={3,4,5}; B. String a2[]{"string1","string1","string1"};
C. String a3[]=new String(3); D. int[][] a4=new int[3][3];

程序分析题

1. 分析下面的程序，然后给出它的执行结果 ()

```
class Main {
    private static int cnt=0;
    public void run() {
new Thread()->{
        try {
            Thread.sleep(1000);
        } catch (InterruptedException e) {
        }
        synchronized(this) {
            cnt++;
        }
    }).start();
}
    public static void main(String [] args) {
        try {
            Main m = new Main();
            for ( int i=0; i<3; i++ )
                m.run();
            Thread.sleep(5000);
            System.out.println(cnt);
        } catch (InterruptedException e) {
        }
    }
}}
```

应用编程题

1. 设计一个学生信息管理系统的一部分。

要求:

- (1) 定义一个Student类，包含学号(id)、姓名(name)、年龄(age)和成绩(score)等属性
- (2) 实现构造方法、getter和setter方法
- (3) 重写toString()方法，返回学生的完整信息
- (4) 在主类中创建一个包含3个学生对象的数组，并实现按照成绩从高到低排序的功能
- (5) 输出排序前和排序后的学生信息