

大数据工程技术专业

《大数据开发综合》考试大纲

一、考试基本要求

本考试为计算机相关专业专升本阶段的专业核心课程考试，核心考核 Java 语言编程基础、面向对象编程思想与实践、Java 核心类库应用、集合框架使用及数据库系统理论与 SQL 实操等内容。

通过本考试，旨在检验考生是否具备以下能力与认知：

1) 理解 Java 语言特性与跨平台运行机制，掌握 Java 基本语法并能编写基础应用程序；

2) 掌握面向对象编程的核心思想与三大特征，能熟练进行类与对象的设计、封装、继承及多态实现；

3) 熟悉 Java 核心 API 的使用，包括字符串处理、日期时间操作、随机数生成等常用功能；

4) 理解集合框架的设计理念，掌握 List、Set、Map 等核心集合的使用方法及迭代器操作；

5) 了解数据库系统的基本概念、体系结构，掌握关系型数据库理论基础与 SQL 标准语言的实操应用；

6) 具备基础的数据库设计思维与数据保护意识，了解数据库备份恢复、并发控制等核心维护知识。

二、考核范围与核心知识点

(一) Java 编程基础

1. 核心考核内容

- 1) Java 程序的基本结构与规范格式；
- 2) Java 注释（单行、多行、文档注释）的使用规范；
- 3) 标识符的命名规则与关键字的分类及使用限制；
- 4) 常量的定义（字面常量、final 常量）与使用；
- 5) 变量的定义、初始化及数据类型分类（基本数据类型、引用数据类型）；

- 6) 变量的类型转换（自动类型转换、强制类型转换）规则；
- 7) 变量的作用域划分（局部变量、成员变量）及生命周期；
- 8) 常用运算符：算术运算符、赋值运算符、比较运算符、逻辑运算符的使用及优先级规则；
- 9) 选择结构语句：if-else 条件语句、三元运算符、switch 多分支语句的语法与应用场景；
- 10) 循环结构语句：while 循环、do-while 循环、for 循环的语法及循环嵌套的实现；
- 11) 跳转语句：break 与 continue 的作用机制及区别；
- 12) 方法的定义、参数传递、返回值处理及方法重载的实现；
- 13) 一维数组与二维数组的定义、初始化、遍历及常见操作（元素访问、长度获取）。

2. 考核要求

- 1) 了解 Java 语言的发展历程、核心特点（跨平台性、面向对象、安全性等）；
- 2) 理解 Java 虚拟机（JVM）的运行机制与跨平台原理；
- 3) 熟练掌握 Java 基本语法格式与规范；
- 4) 熟练掌握常量与变量的定义、初始化及使用规则；
- 5) 熟练运用各类运算符，能准确判断运算符优先级并解决实际计算问题；
- 6) 熟练使用选择结构与循环结构语句，能根据业务需求设计分支与循环逻辑；
- 7) 熟练掌握方法的定义、调用及方法重载的实现，理解方法的参数传递机制；
- 8) 熟练掌握一维数组的定义、初始化、遍历及常见操作；
- 9) 了解二维数组的基本使用方法。

3. 典型例题示例

- 1) 填空题：在 Java 中，变量的数据类型分为两种，即基本数据类型和_____类型。
- 2) 简答题：简述跳转语句 break 与 continue 的作用及核心区别。

3) 编程题：编写程序，使用 for 循环计算 1 到 100 之间所有偶数的和。

(二) 面向对象编程（上）

1. 核心考核内容

- 1) 面向对象编程的核心思想与核心优势；
- 2) 类的定义语法（成员变量、成员方法、构造方法）；
- 3) 对象的创建（new 关键字）与使用（成员访问、方法调用）；
- 4) 对象的引用传递机制与内存模型；
- 5) 访问控制修饰符（private、default、protected、public）的作用范围；
- 6) 封装的设计思想、实现意义及实现方式（private 修饰+getter/setter 方法）；
- 7) 构造方法的定义规则、作用及重载实现；
- 8) this 关键字的核心作用（区分成员变量与局部变量、调用本类构造方法、表示当前对象）；
- 9) 代码块的分类与执行顺序（普通代码块、构造代码块、静态代码块）；
- 10) 静态变量与静态方法的定义、特点及调用方式；
- 11) 静态代码块的作用与执行时机。

2. 考核要求

- 1) 深刻理解面向对象编程思想，掌握封装、继承、多态三大核心特征；
- 2) 熟练掌握类的定义语法与对象的创建、使用流程；
- 3) 熟练理解并运用对象的引用传递机制；
- 4) 熟练掌握四种访问控制修饰符的作用范围与使用场景；
- 5) 熟练掌握封装的实现方式，能设计符合封装规范的类；
- 6) 熟练掌握构造方法的定义、重载及调用规则；
- 7) 熟练掌握 this 关键字与 static 关键字的使用场景与核心作用；
- 8) 理解各类代码块的执行顺序，能准确分析程序运行结果。

3. 典型例题示例

- 1) 填空题：面向对象的三大特征是_____、_____、_____。
- 2) 单选题：下列关于 this 关键字的说法中，错误的是（ ）
只能在构造方法中使用 this 调用其他构造方法，不能在成员方法中使用

在构造方法中，使用 **this** 调用构造方法的语句必须位于第一行，且只能出现一次

this 关键字可以用于区分成员变量与局部变量

this 可以出现在任何成员方法中

3) 编程题：设计一个“学生”类 (**Student**)，包含学号 (**id**)、姓名 (**name**)、年龄 (**age**) 三个成员变量，通过封装实现成员变量的访问控制，提供无参构造、全参构造及 **showInfo()** 方法展示学生信息。

(三) 面向对象编程 (下)

1. 核心考核内容

1) 继承的定义语法 (**extends** 关键字)、核心作用及单继承限制；

2) 方法重写的定义规则、与方法重载的区别；

3) **super** 关键字的核心作用 (访问父类成员变量、调用父类成员方法、调用父类构造方法)；

4) **final** 关键字的使用：修饰类 (不可继承)、修饰方法 (不可重写)、修饰变量 (不可修改)；

5) 抽象类的定义 (**abstract** 关键字)、特点及使用规则；

6) 接口的定义 (**interface** 关键字)、特点及实现方式 (**implements** 关键字)；

7) 多态的实现条件 (继承、方法重写、父类引用指向子类对象) 及核心优势；

8) 对象类型转换：向上转型 (自动) 与向下转型 (强制，需结合 **instanceof** 判断)；

9) **instanceof** 关键字的作用 (判断对象类型)；

10) **Object** 类的核心方法 (**toString()**、**equals()**、**hashCode()**) 及重写；

11) 内部类的分类与使用：成员内部类、局部内部类、静态内部类、匿名内部类；

12) 异常的定义、分类 (编译时异常、运行时异常)；

13) 异常处理机制：**try-catch-finally** 语句、**throws** 声明异常、**throw** 抛出异常；

14) 自定义异常的实现 (继承 **Exception/ RuntimeException**) 与使用。

2. 考核要求

- 1) 熟练掌握类的继承语法与核心规则，理解继承的代码复用价值；
- 2) 熟练掌握方法重写的规则，能准确区分方法重写与方法重载；
- 3) 熟练掌握 `super` 关键字的使用场景与核心作用；
- 4) 熟练掌握 `final` 关键字修饰类、方法、变量的不同效果；
- 5) 熟练掌握抽象类与接口的定义、区别及使用场景；
- 6) 熟练掌握多态的实现条件与使用方式，能结合多态设计灵活的程序结构；
- 7) 熟练掌握对象类型转换规则与 `instanceof` 关键字的使用；
- 8) 了解 `Object` 类的核心方法，能根据需求重写 `toString()` 与 `equals()` 方法；
- 9) 了解各类内部类的定义与使用场景，重点掌握匿名内部类的应用；
- 10) 理解异常的产生机制，熟练掌握 `try-catch-finally` 异常处理流程；
- 11) 掌握 `throws` 与 `throw` 关键字的使用区别，了解自定义异常的实现流程。

3. 典型例题示例

- 1) 填空题：Java 中一个类最多可以继承_____个类；实现多态的三个必要条件是_____、_____、_____。
- 2) 简答题：简述抽象类与接口的核心区别。
- 3) 编程题：定义一个“动物”抽象类（`Animal`），包含抽象方法 `eat()`；定义“猫”（`Cat`）和“狗”（`Dog`）类继承 `Animal` 类并实现 `eat()` 方法；编写测试类，通过多态方式创建 `Cat` 和 `Dog` 对象并调用 `eat()` 方法。

（四）Java 核心 API 应用

1. 核心考核内容

- 1) `String` 类的初始化方式（直接赋值、`new` 关键字）与字符串常量池；
- 2) `String` 类的常见操作：字符串拼接、截取、替换、查找、比较、大小写转换；
- 3) `StringBuffer` 与 `StringBuilder` 类的特点、区别及常见操作；
- 4) `System` 类的核心方法（`out.println()`、`currentTimeMillis()`、`arraycopy()`）；

- 5) `Runtime` 类的作用与常用方法 (`exec()`、`availableProcessors()`);
- 6) `Math` 类的常用静态方法 (绝对值、平方根、三角函数、随机数生成);
- 7) `Random` 类的使用 (生成指定范围的随机数);
- 8) 日期时间类的使用 (`Date`、`SimpleDateFormat`、`Calendar/LocalDateTime`);
- 9) 包装类的定义、自动装箱与自动拆箱机制;
- 10) 正则表达式的基本语法与应用 (字符串匹配、替换、分割)。

2. 考核要求

- 1) 熟练掌握 `String` 类的初始化方式与字符串常量池的原理;
- 2) 熟练运用 `String` 类的常见操作方法解决字符串处理问题;
- 3) 熟练掌握 `StringBuffer` 与 `StringBuilder` 类的使用, 能根据场景选择合适的字符串拼接类;
- 4) 熟练掌握 `System` 类与 `Runtime` 类的常用方法;
- 5) 熟练掌握 `Math` 类与 `Random` 类生成随机数的方法;
- 6) 熟练掌握日期时间类的使用, 能实现日期的格式化、解析及日期计算;
- 7) 理解包装类的核心作用, 掌握自动装箱与拆箱机制;
- 8) 了解正则表达式的基本语法, 能运用正则表达式解决简单的字符串匹配问题。

3. 典型例题示例

- 1) 编程题: 编写程序, 每次随机生成 10 个 0 (包含) 到 100 之间的整数并打印输出。
- 2) 编程题: 编写程序实现字符串大小写转换并倒序输出。要求: ①遍历字符串“ITcastHeiMa”; ②将大写字符转为小写, 小写转为大写; ③使用 `StringBuffer` 拼接遍历结果并输出。
- 3) 简答题: 简述 `String`、`StringBuffer`、`StringBuilder` 三者的核心区别。

(五) 集合框架

1. 核心考核内容

- 1) 集合框架的核心作用、与数组的区别;
- 2) `Collection` 接口的核心方法 (添加、删除、判断、遍历);

- 3) List 接口的特点（有序、可重复）及实现类：ArrayList（数组实现）、LinkedList（链表实现）；
- 4) ArrayList 与 LinkedList 的特点对比及适用场景；
- 5) Iterator 迭代器的使用（hasNext()、next()、remove()）；
- 6) foreach 循环的语法与适用场景；
- 7) Set 接口的特点（无序、不可重复）及实现类：HashSet（哈希表实现）、TreeSet（红黑树实现）；
- 8) Map 接口的特点（键值对存储、键唯一）及实现类：HashMap（哈希表实现）、TreeMap（红黑树实现）；
- 9) Map 接口的常用方法（添加键值对、获取值、遍历键/值/键值对）；
- 10) Properties 类的特点与使用（读取配置文件）；
- 11) 泛型的核心思想、作用（类型安全、避免类型转换）；
- 12) 泛型类、泛型方法、泛型接口的定义与使用；
- 13) 类型通配符（?、? extends T、? super T）的使用；
- 14) Lambda 表达式的基本语法与在集合遍历中的应用。

2. 考核要求

- 1) 理解集合框架的设计理念，掌握集合与数组的核心区别；
- 2) 熟练掌握 Collection 接口的核心方法；
- 3) 熟练掌握 ArrayList 与 LinkedList 的使用方法，理解其底层实现差异及适用场景；
- 4) 熟练掌握 Iterator 迭代器与 foreach 循环的使用；
- 5) 熟练掌握 HashSet、TreeSet 的使用方法，理解其去重机制；
- 6) 熟练掌握 HashMap、TreeMap 的使用方法，掌握其键值对的遍历方式；
- 7) 了解 Properties 类的使用场景，能运用其读取简单配置文件；
- 8) 熟练掌握泛型的核心作用，能定义并使用泛型类、泛型方法、泛型接口；
- 9) 理解类型通配符的使用场景；
- 10) 了解 Lambda 表达式的基本语法，能运用其简化集合遍历代码。

3. 典型例题示例

- 1) 简答题：简述 List、Set、Map 三大接口的核心区别。
- 2) 简答题：为什么 ArrayList 的增删操作效率较低，而查找操作效率较高？
- 3) 编程题：使用 ArrayList 存储 5 个学生对象，通过 Lambda 表达式遍历并打印学生信息。

（六）数据库系统概述

1. 核心考核内容

- 1) 核心概念：信息、数据、数据处理、数据库（DB）、数据库管理系统（DBMS）、数据库系统（DBS）；
- 2) 数据库技术的发展历程（人工管理阶段、文件系统阶段、数据库系统阶段）及各阶段特点；
- 3) 数据库管理系统的核心功能与组成；
- 4) 数据库系统的体系结构（三级模式：外模式、模式、内模式；两级映像）；
- 5) 数据的三个世界：现实世界、信息世界（实体、属性、关系）、计算机世界（字段、记录、表）。

2. 考核要求

- 1) 熟练掌握信息、数据、DB、DBMS、DBS 等核心概念的定义及相互关系；
- 2) 了解数据库技术的发展历程及各阶段的核心特点；
- 3) 理解数据库管理系统的核心功能；
- 4) 熟练掌握数据库系统的三级模式与两级映像体系结构；
- 5) 理解数据三个世界的对应关系及核心概念。

3. 典型例题示例

- 1) 填空题：数据管理技术经历了_____、_____、_____三个发展阶段。
- 2) 简答题：简述数据库系统（DBS）的组成部分。

3) 填空题：在关系型数据库中，数据以二维表形式存储，每个二维表称为_____。

(七) 关系型数据库理论基础

1. 核心考核内容

- 1) 关系模型的核心概念（关系、元组、属性、主键、外键、域、关系模式）；
- 2) 关系模型的组成（数据结构、数据操作、完整性约束）；
- 3) 关系的基本性质（行无序、列无序、元组唯一、属性原子性）；
- 4) 关系完整性约束：实体完整性、参照完整性、用户定义完整性；
- 5) 关系代数运算：集合运算（并、交、差、笛卡尔积）、专门关系运算（选择、投影、连接、除）。

2. 考核要求

- 1) 熟练掌握关系模型的核心概念；
- 2) 熟练掌握关系模型的三大组成部分；
- 3) 理解关系的基本性质；
- 4) 熟练掌握三类关系完整性约束的定义与应用；
- 5) 熟练掌握关系代数的集合运算与专门关系运算的规则，能完成简单的关系代数运算。

3. 典型例题示例

- 1) 填空题：消除了主属性对主键的部分函数依赖和传递函数依赖的范式是_____。
- 2) 简答题：简述关系模型的三类完整性约束及其核心要求。
- 3) 计算题：已知关系 R 和关系 S，计算 $R \cup S$ 、 $R \cap S$ 、 $R - S$ 及 R 与 S 的自然连接结果。

(八) SQL 标准语言实操

1. 核心考核内容

- 1) SQL 语言的核心特点（综合统一、高度非过程化、面向集合、语法简洁）；

2) 数据定义语言 (DDL): 数据库创建 (CREATE DATABASE)、表创建 (CREATE TABLE)、表结构修改 (ALTER TABLE)、表删除 (DROP TABLE)、索引创建与删除 (CREATE/DROP INDEX);

3) 数据操纵语言 (DML): 插入数据 (INSERT)、修改数据 (UPDATE)、删除数据 (DELETE);

4) 数据查询语言 (DQL): 简单查询 (SELECT)、条件查询 (WHERE)、排序 (ORDER BY)、分组统计 (GROUP BY、HAVING)、多表连接查询 (内连接、外连接、交叉连接)、子查询 (相关子查询、非相关子查询)、聚合函数 (COUNT、SUM、AVG、MAX、MIN);

5) 数据控制语言 (DCL): 用户授权 (GRANT)、权限回收 (REVOKE)、事务控制 (COMMIT、ROLLBACK)。

2. 考核要求

- 1) 了解 SQL 语言的核心特点;
- 2) 熟练掌握 DDL 语句的使用, 能完成数据库、表、索引的创建与管理;
- 3) 熟练掌握 DML 语句的使用, 能完成数据的增删改操作;
- 4) 熟练掌握 DQL 语句的使用, 能完成简单查询、条件查询、排序、分组统计及多表连接查询;
- 5) 理解子查询的执行机制, 能完成简单的相关子查询与非相关子查询;
- 6) 了解 DCL 语句的使用, 掌握基本的权限管理与事务控制方法。

3. 典型例题示例

- 1) 填空题: 在 SQL 的 SELECT 语句中, 要去除查询结果中的重复记录, 应使用_____关键字。
- 2) 简答题: 简述相关子查询与非相关子查询的核心区别。
- 3) 编程题: 已知“学生表”(student: 学号、姓名、性别、年龄、班级编号)和“班级表”(class: 班级编号、班级名称), 编写 SQL 语句查询每个班级的班级名称及该班级的学生人数(要求显示人数为 0 的班级)。

(九) 数据库保护

1. 核心考核内容

- 1) 事务的定义、核心特征 (ACID: 原子性、一致性、隔离性、持久性);

- 2) 数据库完整性控制：完整性约束的定义与实现；
- 3) 并发操作带来的问题：丢失修改、不可重复读、脏读；
- 4) 并发控制技术：封锁机制（共享锁、排他锁）、隔离级别；
- 5) 数据库备份与恢复的核心原理；
- 6) 常见的备份方式：完全备份、差异备份、日志备份；
- 7) 数据恢复的实现流程。

2. 考核要求

- 1) 熟练掌握事务的定义与 ACID 核心特征；
- 2) 理解数据库完整性控制的核心意义；
- 3) 理解并发操作带来的三类问题及产生原因；
- 4) 了解封锁机制与隔离级别的核心作用；
- 5) 了解数据库备份与恢复的核心原理，掌握常见的备份方式。

3. 典型例题示例

- 1) 填空题：解决并发控制带来的数据不一致问题，普遍采用的技术是_____。
- 2) 简答题：简述事务的 ACID 特征。
- 3) 填空题：并发操作带来的异常包括丢失更新、脏读和_____。

（十）数据库设计

1. 核心考核内容

- 1) 数据库设计的基本步骤：需求分析、概念结构设计、逻辑结构设计、物理结构设计、数据库实施、数据库运行与维护；
- 2) 需求分析的核心任务与输出（需求规格说明书）；
- 3) 概念结构设计：E-R 模型的设计（实体、属性、关系的识别与绘制）；
- 4) 逻辑结构设计：E-R 模型向关系模式的转换规则；
- 5) 物理结构设计：存储结构选择、索引设计；
- 6) 数据库运行与维护的核心内容（性能监控、优化、数据备份）。

2. 考核要求

- 1) 熟练掌握数据库设计的基本步骤；

- 2) 理解需求分析的核心任务；
- 3) 熟练掌握 E-R 模型的设计方法，能根据业务需求绘制简单的 E-R 图；
- 4) 熟练掌握 E-R 模型向关系模式的转换规则，能完成简单 E-R 图的关系模式转换；
- 5) 了解物理结构设计的核心内容；
- 6) 了解数据库运行与维护的核心任务。

3. 典型例题示例

- 1) 编程题：设计学生借书还书系统的 E-R 图，包含学生、班级、图书三个实体（属性如下），并将其转换为关系模式，指出每个关系模式的主键和外键。
- 学生实体：学号（唯一）、姓名、性别、出生日期；
 - 班级实体：班号（唯一）、班名；
 - 图书实体：图书号（唯一）、书名、作者、单价、出版社。
- 2) 简答题：简述数据库设计的基本步骤。

三、考试题型与分值分布

题型	题量	每题分值	总分值
单选题	10 题	4 分	40 分
判断题	5 题	4 分	20 分
填空题	10 空	2 分	20 分
简答题	4 题	15 分	60 分
编程题	3 题	20 分	60 分
合计			200 分

四、考试形式

笔试（闭卷），考生需携带黑色签字笔、2B 铅笔、橡皮等文具，禁止携带任何电子设备、参考资料及存储介质。

五、考试时间

150 分钟（具体考试时间以官方公布的考试安排为准）。

六、参考书目

（一）核心教材

1) 《Java 编程从入门到实战》，水利水电出版社，ISBN：978-7-5170-9933-8

2) 《数据库原理及应用（双色版）》，上海交通大学出版社，ISBN：978-7-3132-7388-8

（二）推荐参考书

1) 《Java 核心技术卷 I 基础知识》（原书第 11 版）

2) 《Java 编程思想（第 4 版）》

3) 《SQL Server 2012 数据库应用与开发教程》，清华大学出版社，ISBN：978-7-3024-0008-0

4) 《关系数据库与 SQL Server 2012 第 3 版》，机械工业出版社，ISBN：978-7-1115-1590-6