**一、举例 ﻿**

**1. 例题1: TestArgsWords程序解析**

A computer screen with a white screen

AI-generated content may be incorrect.

* 程序功能：统计命令行参数中每个单词出现的次数
* 核心实现：
  + 成员变量：private static final Integer ONE = new Integer(1)，使用final修饰表示该引用不可更改
  + Map初始化：Map m = new HashMap()创建哈希映射用于存储单词计数
  + 循环处理：通过for循环遍历所有输入参数
  + 计数逻辑：
    - 使用m.get(args[i])获取当前单词的计数
    - 若单词不存在（返回null），则存入初始值1
    - 若单词已存在，则将其计数值加1后重新存入
  + 输出结果：打印不同单词的数量及详细统计

A computer screen with a black and white screen

AI-generated content may be incorrect.

* 运行示例：
  + 输入：java TestArgsWords aaa bbb ccc aaa ddd ccc ccc
  + 输出：4 distinct words detected:{ccc=3, aaa=2, ddd=1, bbb=1}

**2. 例题2: TestArgsWords程序运行与结果分析 ﻿**

* 执行流程分析：
  + 首次处理单词：当处理第一个参数"aaa"时，Map为空，m.get("aaa")返回null，因此存入初始值1
  + 重复单词处理：当再次遇到"aaa"时，m.get("aaa")返回1，执行new Integer(freq.intValue()+1)后存入2
  + 新单词处理：遇到新单词时重复首次处理流程
* 自动装箱优化建议：
  + 原始代码中显式创建Integer对象（如new Integer(1)）
  + 可简化为自动装箱方式：Integer ONE = 1和m.put(args[i], (freq == null ? ONE : freq + 1))
* 类型安全警告：
  + 编译时提示"未经检查或不安全的操作"
  + 建议使用泛型声明Map：Map<String, Integer> m = new HashMap<>()

A computer screen shot of a computer

AI-generated content may be incorrect.

* 关键步骤说明：
  + 初始化空Map
  + 遍历每个输入参数：
    - 检查Map中是否已存在该单词
    - 不存在则计数设为1
    - 存在则计数加1
  + 循环结束后输出统计结果
* 数据结构变化示例：
  + 输入"aaa" → Map变为{"aaa":1}
  + 输入"bbb" → Map变为{"aaa":1, "bbb":1}
  + 再次输入"aaa" → Map变为{"aaa":2, "bbb":1}

**二、知识小结**

|  |  |  |  |
| --- | --- | --- | --- |
| 知识点 | 核心内容 | 考试重点/易混淆点 | 难度系数 |
| 自动打包与解包 | 通过Integer自动装箱/拆箱简化代码，如Integer one = 1直接赋值 | final修饰的变量不可变，但对象内容可修改（如Map值更新） | ⭐⭐ |
| HashMap统计词频 | 使用HashMap<String, Integer>统计单词出现次数，键为单词，值为频次 | 初始值为null时需处理（三元运算符赋默认值one） | ⭐⭐⭐ |
| 三元运算符应用 | frequency == null ? one : new Integer(frequency + 1)动态更新频次 | 注意自动拆箱（frequency.intValue()）与重新装箱（new Integer） | ⭐⭐ |
| 循环遍历参数 | for循环遍历命令行参数数组args，逐个处理单词 | 数组越界风险（需确保args非空） | ⭐ |
| Map操作逻辑 | map.get(key)获取值→判断null→map.put(key, newValue)更新 | 键值覆盖机制：重复put会更新原有值 | ⭐⭐⭐ |