**12\_示例\_2\_笔记**

**一、自动类型转换**

A computer screen shot of a program

AI-generated content may be incorrect.

* 类型转换优化：将Integer对象转换为基本int类型，简化代码写法。原代码Integer frequency = m.get(xi)改为int frequency = m.get(xi)
* 空值处理：当第一次从HashMap取值时可能遇到空值，需要特殊处理：
  + 若值为空则赋值为0
  + 否则保持原值
* 改进写法：使用三元运算符简化空值判断：int freq = (Integer)m.get(args[i]) == null ? 0 : (Integer)m.get(args[i])
* 常量使用：定义private static final int ONE = 1作为常量，但实际可以直接使用字面值1

**二、编译运行**

A computer screen shot of a computer screen

AI-generated content may be incorrect.

* 词频统计逻辑：
  + 使用m.put(args[i], freq==0 ? ONE : freq + 1)更新词频
  + 若词频为0（首次出现）则设为1，否则在原值基础上加1
* 编译错误：
  + 注意方法名拼写：m.put()而非m.pub()
  + 未经检查操作警告：需使用-Xlint:unchecked查看详情
* 运行示例：
  + 输入：aaa bbb ccc aaa ddd ccc ccc
  + 输出：统计不同单词出现次数
  + 空指针异常处理：确保在取值时处理可能的null值

**三、知识小结**

|  |  |  |  |
| --- | --- | --- | --- |
| 知识点 | 核心内容 | 易混淆点 | 难度系数 |
| Java自动类型转换 | 将Integer对象转换为int基本类型时需处理null值情况 | 空值无法直接转换为int类型 | ⭐⭐ |
| Map集合操作优化 | 使用m.put(xi, frequency == null ? 1 : frequency + 1)简化空值判断逻辑 | 原始写法需要显式判断null并赋默认值 | ⭐⭐⭐ |
| 泛型编程优势 | 后续课程将介绍泛型可简化类型转换操作 | 当前非泛型写法需要手动处理类型安全 | ⭐⭐ |
| 代码调试演示 | 通过单词统计案例验证修改效果（aaa出现2次，bbb出现3次） | 原始输出显示"0个"需注意初始化问题 | ⭐⭐ |