**01\_容器API\_Collection\_2**

**1. 编译问题**

* A screenshot of a computer

  AI-generated content may be incorrect.
* 解决方法：将Name类与测试类放在同一个Java文件中才能成功编译
* 关键点：Java要求内部类或关联类需在同一文件或包中才能相互调用

**2. Collection方法举例**

**﻿1）小程序演示**

A screenshot of a computer program

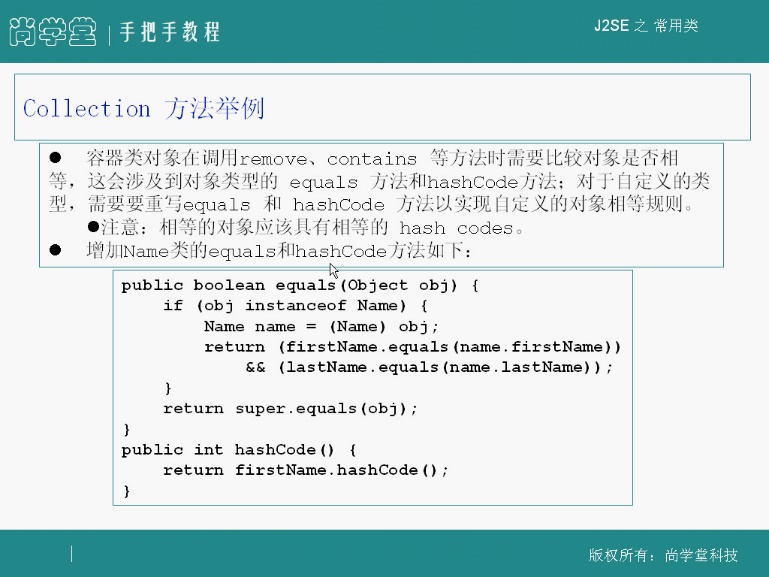
AI-generated content may be incorrect.

* 容器类型：
  + Set：无序不可重复，示例中使用HashSet实现
  + List：有序可重复，可替换为ArrayList/LinkedList
* 基本操作：

**2）注意点分析**

* 对象比较机制：
  + equals方法：容器通过equals方法比较对象是否相等
  + 示例分析：
    - String和Integer类已重写equals方法，能正确比较内容
    - 未重写equals的自定义类（如Name）仅比较引用地址
* remove操作原理：
  + 遍历容器内所有对象，调用equals方法进行比较
  + 匹配成功则移除，返回true；否则返回false

**3）增加equals和hashCode方法**



* 重写规范：
  + equals方法：
  + hashCode方法：
* 黄金法则：
  + 相等的对象必须具有相等的hashCode
  + 重写equals必须同时重写hashCode
  + 特别当对象作为Map的key使用时

**4）动手实验**

**﻿**

A computer screen shot of a computer screen

AI-generated content may be incorrect.

* 实验结果：
  + 未重写equals时：remove返回false，容器保留Name对象
  + 重写equals后：remove返回true，成功移除Name对象
* 哈希机制：
  + 作用原理：通过hashCode快速定位对象存储位置
  + 优化建议：不同对象尽量产生不同hashCode提升性能
  + 典型实现：String的hashCode已优化，可直接复用

**二、知识小结**

|  |  |  |  |
| --- | --- | --- | --- |
| 知识点 | 核心内容 | 考试重点/易混淆点 | 难度系数 |
| Java类文件组织 | 强调类文件需放在同一文件中才能编译通过 | name类未与主类同文件导致编译失败 | ⭐⭐ |
| 集合框架基础 | HashSet/LinkedList/ArrayList都是容器实现 | 不同集合实现的底层差异(哈希表/链表/数组) | ⭐⭐⭐ |
| 对象相等性判断 | remove操作依赖equals方法比较对象 | Integer重写equals而自定义类未重写导致不同结果 | ⭐⭐⭐⭐ |
| equals与hashCode契约 | 重写equals必须重写hashCode | 集合作为键使用时依赖hashCode | ⭐⭐⭐⭐⭐ |
| 泛型警告处理 | 未使用泛型时出现编译警告 | 后续泛型课程会解决该问题 | ⭐⭐ |
| 字符串哈希机制 | String类已实现规范的hashCode | 自定义类可委托字符串计算哈希值 | ⭐⭐⭐ |
| 集合操作演示 | add/remove方法实际效果验证 | 相同值不同Integer对象仍被移除 | ⭐⭐⭐ |
| 面向对象基础 | 所有类默认继承Object | Object.equals默认比较对象地址 | ⭐⭐ |

该部分主要讲述了Java中对象比较和移除的机制以及hash code方法的重要性。首先介绍了如何将对象放入HashSet和LinkedList等容器中，并重写equals()和hashCode()方法以比较对象。当使用remove()方法移除对象时，容器会遍历每个对象并使用equals()方法进行比较，如果找到相等的对象则将其移除。对于使用哈希表的容器，如HashSet，还会使用hashCode()方法以提高效率。此外还提到了泛型的使用和消除警告的机制。最后强调了重写equals()和hashCode()方法的重要性，以确保对象的正确比较和移除。

Name类和容器类的关系

1.Name类应该与包含它的类放在同一个文件中，以便编译。

2.容器类（如HashSet）用于存储对象，并提供添加、删除和访问元素的方法。

3.HashSet基于哈希表实现，具有快速的查找和插入性能。

HashSet和LinkedList的区别

1.HashSet和LinkedList是两种不同的容器实现，用于存储和操作对象。

2.HashSet基于哈希表实现，使用哈希码进行快速查找和插入。

3.LinkedList基于链表实现，适用于需要按顺序访问元素的场景。

HashSet的remove方法

1.HashSet的remove方法用于从容器中删除指定的对象。

2.删除对象时，通过比较对象的equals方法来确定是否匹配。

3.如果对象成功删除，remove方法返回true；否则返回false。

Name类的equals和hashCode方法重写

1.重写equals方法用于比较Name对象的名字和姓氏是否相等。

2.重写hashCode方法确保相等的对象具有相同的哈希码。

3.equals方法和hashCode方法应一起重写，以确保容器的正确工作。

equals方法和hashCode方法的重要性

1.当对象用作容器类的键时，equals方法和hashCode方法至关重要。

2.equals方法用于比较对象是否相等，hashCode方法用于生成对象的哈希码。

3.相等的对象必须具有相同的哈希码，以确保容器（如Map）的正确工作。