

Java

黄浩

2021-2-8 19:27

目录

第一部分 Java 基础部分	1
1 线程	2
2 集合	2
2.1 迭代器	2
2.1.1 iterator 迭代器	2
2.2 map 集合	2
2.3 哈希表及其相关知识	3
2.3.1 hashcode	3
2.3.2 哈希表	3
2.3.3 hashmap	3
2.4 set	4
3 备忘	4

第一部分 Java 基础部分

摘要

java 基础笔记

1 线程

线程关闭，不可调用 stop 方法，因为可能有资源没有释放掉，可以设置一个变量通知线程，当值为-1 时，return run 方法（合理关闭）

线程分类：daemon 守护线程，用户线程

2 集合

2.1 迭代器

2.1.1 iterator 迭代器

失败：集合中数据被改了，数据肯定就不对了，我们就称为失败；

快速失败：遍历数据是集合本身，会抛出异常；

安全失败：遍历的是集合数据的副本，不会出现异常；

如果 API 不特别说明，默认安全失败；

2.2 map 集合

map (interface): mapping 一个键只能对应一个值

2.3 哈希表及其相关知识

2.3.1 hashCode

hashCode: int 类型值

值会因对象、值变化，而分布的比较均匀；相同属性也不建议值一样

2.3.2 哈希表

哈希表：对象数组 + 链表 (java 中) 依据对象的 hashCode 值来和数组长度取余运算，得到的数字作为下标，放入数组，这样查找快余数相等就用链表存，数组中的每一个下标，叫哈希桶哈希桶的长度大于 8 时，转换成红黑树当哈希桶中的数据量减少到 6 时，从红黑树转换成链表

问：如果哈希桶中数据为 7 个，一定会从红黑树转为链表么？答：不一定，原来 7，可能还是链表结构，不用转

2.3.3 hashmap

hashmap：影响性能的两个参数：1. 初始（桶）数量：16（扩容为 2 倍原长度）！注意，一旦桶的数量更改了（即下标范围变更了），**测试** 需要重新取余计算 2. 散列因子（加载因子）：0.75（有 75% 的桶都装有数据了，进行扩容即散列（通过重建内部数据结构））；该参数反映在存储空间和查找效率上；有数组，就算哈希表了，不一定非要有链表存在。存储值时，先依据键计算哈希值，确定存放位置；源码中用（数组长度-1）和哈希值与运算就等于哈希值取余长度里面的 key 值尤其是自定义类型，不要乱改值，不然就找不到了，确认两个键相等，需要满足哈希值相等，还有满足 equals。

list, map, set 接口提供了固定长度的集合，of () 重载方法

HashMap：不保证存储有序 TreeMap：不保证存储有序，自动排序
LinkedHashMap：保证存储有序，通过使用双向链表和哈希表来完成，既保证顺序，也保证查找高效

2.4 set

set 不保证顺序，属性更改会导致不同位置

3 备忘

需要创建一个 id，setting 中，复制，已完成

编译速度太快，更改，已完成

VSCode 默认单击文件是预览，双击文件是用一个新的 Tab 打开。如果想关掉预览模式，单击打开文件，在设置里加入 [参考文档](#) 或者更改 JSON 文件 "workbench.editor.enablePreview": false 这样单击文件就是直接用新 Tab 打开了。