前置

讲师：景天

上课时间：20:00

分享资料：课堂笔记

HashMap核心知识点分析

1、使用

就是最常见的一个内存存储结构，

- 创建对象，HashMap map = new HashMap()
- 写：map.put("zs",18)
- 读:map.get("zs");

re

2、如何实现它的核心功能

核心功能就是存数据的

1、在计算机里面，如何实现数据的存储

有内存存储、有磁盘存储、缓存、寄存器。。。

2、在内存里面如何存

通过数据结构：

- 数组
- 链表
- 队列
- 树
- 图
- 堆
- 栈
- Hash表
- 对象

3、在HashMap里面，到底是由什么数据结构来存的呢

Hash表

4、Hash表到底是什么？

特殊的数组

5、特殊在哪里呢?

Hash表在存数据的时候，数组的下标不是人为给的

i=hash值 & （数组长度-1）

- 因为数组的长度是有限的，而且不会太长，但是hash值是一个非常大的数字（在Java中，hash值是一个int类型的数字。是32位）

  01001010 11000010 10100110 11011001

- 要根据数组的长度来取下标

  如果数组的长度是8，那最大的下标是7，111

  如果数组的长度是16，最大的下标是15,1111

- 就是通过&运算能截取hash值的后几位

  01001010 11000010 10100110 11011001

  111

6、正式因为上面的取模运算会造成问题

hash冲突问题

7、什么叫hash冲突问题呢?

girl: 01001010 11000010 10100110 11011001

1111

1001

boy: 01011110 11011010 10101110 11001001

1001

8、一旦发生了Hash冲突，怎么解决?

- 在同一个房间加床：链式地址法；它是HashMap的默认方案
- 找隔壁房间：线性探测（for循环），它是ThreadLocal的默认解决方案
- rehash法

9、有没有一种方案，在发生Hash冲突之前，通过某种运算减小它的发生概率

- 扰动函数，只是在一定的程度上能减少概率，但并不一定会减少
- 数组的长度必须是2的幂次方

10、在newHashMap对象的时候并没有创建数组，这是懒加载思想

11、懒加载又是什么意思

就是在没用到的时候并不去创建，在真正写的时候才会去创建

从设计的底层，就帮你们规避了浪费空间的可能性

12、HashMap中的key和value可以为null吗?

可以

13、HashMap中可以存在多少个key为null的值

1个

14、HashMap中数组的默认长度是16

15、HashMap中的扩容因子是0.75

16、链表转红黑树的条件

- 链表的长度一定要大于等于8
- 数组的长度必须大于等于64

17、HashMap在使用过程中会存在什么极端问题

线程安全问题

18、为什么会出现线程安全问题呢?

在jdk1.7的时候和他之前，由于put会导致数据丢失，同时扩容的时候是头插法，会导致死循环

在jdk1.8之后（包括），只会存在put丢失问题，不存在死循环

3、在实现核心功能的过程中，他源码是怎么玩的

1、创建对象

只干了一件事

```
this.loadFactor = 0.75f;
```

但是没有创建数组，没有创建数组的原因是懒加载思想

2、写的源码

跟把一头大象装进冰箱的步骤类似

- 有数组（冰箱）
- 计算下标
- 放进去

真正的流程:

1. 求girl的hash值

```
 hash ^ (hash >>> 16)//这个方法叫做扰动函数，那为什么要用扰动函数呢--能一定程度上减少Hash冲突的发生概率
```

```
//girl:    01001010 11000010 10100110 11011001     hash
//         00000000 00000000 01001010 11000010   hash>>>16
//         01001010 11000010 11101100 00011011


//boy:     01011110 11011010 10101110 11001001
//         00000000 00000000 01011110 11011010
//         01011110 11011010 11110000 00010011

//取下标的过程中就是取boy和girl身上的某个点来比较，如果这个点相同，则他们两个冲突
//假设现在数组的长度是10，10-1=9    9的2进制是1001
//girl :01001010 11000010 11101100 00011001
```

```
//                                      1001
//                                      1001

//boy:  01011110 11011010 11110000 00011011
//                                      1001
//                                      1001
//取下标的过程中就是取boy和girl身上的某个点来比较，如果这个点，被第三方影响了，就会增加相同的概率
```

2. 创建数组

   Node[] table = new Node[];

```
final Node<K,V>[] resize() {
    Node<K,V>[] oldTab = table;
    int oldCap = (oldTab == null) ? 0 : oldTab.length;
    int oldThr = threshold;
    int newCap, newThr = 0;



        newCap = 16;
        newThr = 12;


    threshold = newThr;

    Node<K,V>[] newTab = (Node<K,V>[])new Node[newCap];
    table = newTab;

    return newTab;
}
```

3. 计算下标

   假设girl的下标计算出来等于3

4. 放数据

   - 如果3号桶里面是空的，我直接将girl，18外面封装一层node，然后放到3号桶里面
   - 如果冲突里面本身就是存的girl，直接覆盖
   - 如果冲突里面是boy，则加床，

```
final V putVal(int hash, K key, V value, boolean onlyIfAbsent,
               boolean evict) {


    //创建长度为16的数组，并且，把这个数组给到tab，把数组的长度给到n=16
    //把3号桶赋值给了p
    //走冲突的逻辑
    else {
        Node<K,V> e; K k;
```

```
        else {
            for (int binCount = 0; ; ++binCount) {
                if ((e = p.next) == null) {
                    p.next = newNode(hash, key, value, null);
                    if (binCount >= TREEIFY_THRESHOLD - 1) // -1 for 1st
                        treeifyBin(tab, hash);
                    break;
                }
                if (e.hash == hash &&
                    ((k = e.key) == key || (key != null && key.equals(k))))
                    break;
                p = e;
            }
        }
        if (e != null) { // existing mapping for key
            V oldValue = e.value;
            if (!onlyIfAbsent || oldValue == null)
                e.value = value;
            afterNodeAccess(e);
            return oldValue;
        }
    }
    ++modCount;
    if (++size > threshold)
        resize();
    afterNodeInsertion(evict);
    return null;
}
```

1、互联网很多大厂都在裁员，或者已经裁完了，他们被裁去中厂，中厂被裁的去哪呢（2015年，湖南大学，2014：260:130多个进了BAT、美图、字节、华为） 120

2015年就苦逼了，50多个，283 233

2、行情不好，没资本，公司都不好过，

大专，3年，只会CRUD，20k

3、好不容易，运气一好，拿到了面试机会。

以前面试对于3年以上，5年以下的基本上只问八股文

现在被八股文，三个面试题就能被吊打

1. spring

   Bean的生命周期、、spring的di、ioc、aop、事务、循环依赖、三级缓存、

   spring的扩展
2. SpringBoot：自动装配、启动流程
3. SpringCloud

- Gate
- 限流
- Nacos
- Dubbo或者openFeign
- zipkin+sleth
- ELK做日志　e：存 logstash：发送