

注意：只整理考察点和一些开放性的问题，没有提到的都是可以 google 到的。

Java 基础

1. 定义 Integer x=20 Integer y=200 在内存里是个什么过程？

主要考察 int 和 Integer 的区别

```
Integer i = 20;

Integer j = 20;

System.out.print(i == j); //true
```

```
Integer i = 200;

Integer j = 200;

System.out.print(i == j); //false
```

java 在编译 Integer i = 200 ;时，会翻译成为 Integer i = Integer.valueOf(200)。而 java API 中对 Integer 类型的 valueOf 方法，对于-128 到 127 之间的数，会进行缓存。

2. volatile 关键字的原理？它能保证原子性吗？AtomicInteger 底层怎么实现的？

Volatile 不能保证原子性，可结合 AtomicInteger 源码通过 CAS 操作保证原子性来看。

3. threadLocal 关键字有用过吗？如果没有重写 initialValue 方法就直接 get 会怎样？

不初始化直接 get 会报 NullPointerException，很简单看源码 initialValue 方法。

```

public T get() {
    Thread t = Thread.currentThread();
    ThreadLocalMap map = getMap(t);
    if (map != null) {
        ThreadLocalMap.Entry e = map.getEntry(key: this);
        if (e != null) {
            /unchecked/
            T result = (T)e.value;
            return result;
        }
    }
    return setInitialValue();
}

```

```

protected T initialValue() {
    return null;
}

```

4. hashMap 与 concurrentHashMap 原理和区别？

hashMap 什么情况下会出现循环链表？ concurrentHashMap 写的时候用什么锁？

ReentrantLock 底层是怎么保证线程安全的？

大家都知道 hashMap 线程不安全，为什么不安全？就是因为在 put 操作的时候没有加锁，可能会导致多个线程进行扩容操作，扩容的时候要复制旧的链表，就可能形成循环链表。

5. 反射能获取到父类的私有方法吗？怎么防止反射破坏单例模式？

可以获取到并且可以执行。怎么破坏单例模式？

继承单例类通过反射获取父类私有构造方法，从而实例化新的对象。

怎么防止？

两种方法：1) 单例类中增加一个标志位，只要初始化第二次就抛异常。

2) 直接用枚举类

6. 描述下 JVM 内存模型。每个区的作用是什么？堆内存的工作原理，为什么需要两个幸存

区？只有一个行不行？老生代是用什么垃圾回收算法？

7. 描述下多线程原理。怎么开启一个线程？start 和 run 方法有什么区别？

怎么创建一个线程池，传入的参数分别什么含义？线程池是怎么实现维持核心线程数的？怎么实现一个自定义的拒绝策略？

线程池是怎么实现维持核心线程数的这个可以看下线程池源码 Worker，知道线程何时结束也就知道怎么维持的了。

情景模拟&其他

1. 这些问题当时答的一般，最近正在结合项目研究，欢迎私聊我讨论。。
2. 用代码实现 `cat 1.log |grep a |sort |uniq -c |sort -rn` 的功能。

```
import java.io.BufferedReader;
```

```
import java.io.File;
```

```
import java.io.FileInputStream;
```

```
import java.io.InputStreamReader;
```

```
import java.util.*;
```

```
/**
```

```
 * 用java实现 cat 1.log |grep a |sort |uniq -c |sort -rn 功能。
```

```
 *      假设：该 log 文件不大，可以直接一次读取到内存。并且该任务没有时效性要求，可以单线程处理。
```

```
 *      思路：按行读取文件，将包含'a'字符的每一行作为 key，出现次数作为 value 存入 map，最后按出现次数排序。
```

```

*/

public class CatLogTest {

    public static void main(String args[]) {

        Map<String,Integer> tmpMap = new HashMap<>();

        try {

            String pathname = "\\data\\log\\1.log";

            File filename = new File(pathname);

            InputStreamReader inputStreamReader = new
InputStreamReader(new FileInputStream(filename));

            BufferedReader bufferedReader = new
BufferedReader(inputStreamReader);

            String line = "";

            //按行读取文件

            while((line = bufferedReader.readLine()) != null){

                if (line.contains("a")){

                    putAndInc(tmpMap,line);

                }

            }

            //按次数排序

            List<Map.Entry<String,Integer>> entryList = new
ArrayList<Map.Entry<String,Integer>>(tmpMap.entrySet());

            Collections.sort(entryList,new

```

```

Comparator<Map.Entry<String,Integer>>() {
    public int compare(Map.Entry<String, Integer>
map1,Map.Entry<String, Integer> map2) {
        //降序
        return map2.getValue().compareTo(map1.getValue());
    }
});

//按要求输出
for(Map.Entry<String,Integer> mapping:entryList){
    System.out.println(mapping.getValue()+" "+
+mapping.getKey());
}
} catch (Exception e) {
    //logger.error("cat log err",e);
}
}

private static void pustAndInc(Map<String,Integer> map,String key){
    if (map.containsKey(key)){
        map.put(key, map.get(key)+1);
    } else{
        map.put(key,1);
    }
}

```

```
        }  
    }  
}
```

3. 如果现在有一台服务器响应突然变得很慢，怎么去定位问题？

- 1) 查看服务器负载、看有没有占用资源很严重的进程。
- 2) 查看 TPS 是否过高。
- 3) 查看网络稳不稳定，有没有频繁丢包。
- 4) 查看数据库有没有慢 SQL，数据库连接数是否异常。
- 5) 查看堆栈是否占用过大。

这是我当时答的几点，最后一点算是答到点子上了。其实面试官希望听到的是查 gc 日志，看是否频繁的垃圾回收。