笔 试 题

1. 描述一下ArrayList和LinkedList的比较。

1.ArrayList是实现了基于动态数组的数据结构，LinkedList是基于链表结构。

2.对于随机访问的get和set方法，ArrayList要优于LinkedList，因为LinkedList要移动指针。

3.对于新增和删除操作add和remove，LinkedList比较占优势，因为ArrayList要移动数据。

1. 什么是cookie？session和cookie有什么区别？

1、cookie数据存放在客户的浏览器上，session数据放在服务器上。

2、cookie不是很安全，别人可以分析存放在本地的COOKIE并进行COOKIE欺骗  
   考虑到安全应当使用session。

3、session会在一定时间内保存在服务器上。当访问增多，会比较占用你服务器的性能  
   考虑到减轻服务器性能方面，应当使用COOKIE。

4、单个cookie保存的数据不能超过4K，很多浏览器都限制一个站点最多保存20个cookie。

5、所以个人建议：  
   将登陆信息等重要信息存放为SESSION

   其他信息如果需要保留，可以放在COOKIE中

1. 什么是依赖注入？什么是控制反转(IOC)？ 在 Spring 中，有几种依赖注入方式？

其实是当一个bean实例引用到了另外一个bean实例时spring容器帮助我们创建依赖bean实例并注入（传递）到另一个bean中;

**就是由spring来负责控制对象的生命周期和对象间的关系;**

**setter;构造函数；注解注入**

1. 如何解决POST请求中文乱码问题，GET的又如何处理呢？

CharacterEncodingFilter

get的话重新编码：

String userName = new String(request.getParamter("userName").getBytes("ISO8859-1"),"utf-8")；

1. 编程题：有一数列，[1,1,2,3,5,8,13,21…….]。求第n项的值。请用java写出求第n项值的方法。

int sum（int a）｛

if(a==1)return 1;

else if(a==2)return 1 ;

else return sum(a-2)+sum(a-1);

｝

1. JAVA实现单例模式。

public class Single{

private Single{}

private static class Inner{

private static Single s=new Single();

}

public static Single getInstance(){

return Inner.s;

}

}

1. Java中CyclicBarrier 和 CountDownLatch有什么不同

CountDownLatch减计数，不可重用；

CyclicBarrier增计数，可重用；

CountDownLatch，当计数为0的时候，下一步的动作实施者是main函数；对于CyclicBarrier，下一步动作实施者是“其他线程”

CountDownLatch和CyclicBarrier都有让多个线程等待同步然后再开始下一步动作的意思，但是CountDownLatch的下一步的动作实施者是主线程，具有不可重复性；而CyclicBarrier的下一步动作实施者还是“其他线程”本身，具有往复多次实施动作的特点。

1. ConcurrentHashMap如何实现线程安全？

**采用的Segment ，16个分段锁功能，每一个Segment 都相当于小的hash table并且都有自己锁，只要修改不在同一个段上就不会引起并发问题。**

9.

class SingleTon {

private static SingleTon singleTon = new SingleTon();

public static int count1;

public static int count2 = 0;

private SingleTon() {

count1++;

count2++;

}

public static SingleTon getInstance() {

return singleTon;

}

}

public class Test {

public static void main(String[] args) {

SingleTon singleTon = SingleTon.getInstance();

System.out.println("count1=" + singleTon.count1);

System.out.println("count2=" + singleTon.count2);

}

}

请输出程序运行结果。

//count本来都是默认0；先初始化使用构造函数new一个对象后，都变成1，然后static关键字的顺序往下走，又重新赋值了，所以  
//count1==1；  
//count2==0；

10.分布式的序号生成器实现方案。

**public static** String **getUUID**(){  
 **return** UUID.**randomUUID**().**toString**().**replace**("-", "");  
 }

snowflake

11.根据下表获得**总分数大于150**的学生名单，请写出sql语句。

|  |  |  |
| --- | --- | --- |
| **学生表(T\_STUDENT)** | | |
| **id** | **stuName** | **stuCode** |
| **3** | **张三** | **zs01** |
| **4** | **李四** | **ls01** |
| **5** | **王五** | **ww01** |

|  |  |  |
| --- | --- | --- |
| **课程表(T\_COURSE)** | | |
| **id** | **courseName** | **courseCode** |
| **5** | **java** | **0004** |
| **12** | **python** | **0005** |

|  |  |  |  |
| --- | --- | --- | --- |
| **分数表（T\_SCORE）** | | | |
| **id** | **stuCode** | **courseCode** | **score** |
| **11** | **zs01** | **0004** | **95** |
| **12** | **zs01** | **0005** | **87** |
| **14** | **ls01** | **0004** | **78** |
| **15** | **ls01** | **0005** | **90** |
| **16** | **ww01** | **0004** | **75** |
| **17** | **ww01** | **0005** | **75** |

**SELECT sname FROM tb\_score GROUP BY sname HAVING SUM(score)>150 ORDER BY SUM(score) DESC;**

**姓名：**