**中软国际考题-笔试**

1. 单选题

1、以下哪个集合接口支持元素排序？（C）

A. Collection B. Set C. List D. Map

2、以下代码的执行结果是？（B）

public class Example {

public static void main(String[] args) {

String s1 = "abc";

String s2 = "def";

Stack<String> stack = new Stack<String>();

stack.push(s1);

stack.push(s2);

try {

String s3 = stack.pop() + stack.peek();

System.out.println(s3);

} catch (Exception e) {

// TODO: handle exception

}

}

}

A、abcdef B、defabc C、abcabc D、defdef

3、以下代码执行结果是？（D）

public class Example {

public static void main(String[] args) {

HashSet<String> set = new HashSet<String>();

String s1 = "abc";

String s2 = "def";

String s3 = "";

set.add(s1);

set.add(s2);

set.add(s1);

set.add(s2);

Iterator<String> i = set.iterator();

while (i.hasNext()) {

s3 += i.next();

}

System.out.println(s3);

}

}

A、abcdefabcdef B、defabcdefabc C、fedcbafedcba D、abcdef

4、下列描述错误的是（B）

A.list下标排列顺序与数组一致B.Set可以通过for循环遍历访问子元素

C.Set一般用来存储不轻易改变的具体现实数据D.list可以有多种清空方式

5、下列描述正确的是（A）

A. Set清空不需要迭代器B. List排列顺序也是按照HashCode

C. List与数组底层实现类是一样的。D. Set与List调用的工具类连接是相同的

6、java中Set描述错误的是（D）

A.迭代器非常适合Set取值 B.Set内部数据存储并不是随机性

C.Set无法存储重复数据 D.Set也可以通过迭代器转换为有序状态存储数据

7、.ArrayList、Vector、LinkedList 说法正确的是(B)

A. ArrayList 是线程安全的

B. Vector 是线程安全的

C.这三个类底层都是由数组实现的

D.LinkedList 是线程安全的，底层是由链表实现的

8.下例哪个说法是正确的（ B）

A.List 不是有序的 Collection

B.次序是 List 最重要的特点：它保证维护元素特定的顺序。

C.使用此接口能够精确的控制每个元素。通过地址来访问 List 中的元素

D List 允许不能有相同的元素。

9、下列语句序列执行后的正确答案是（C）

import java.util.\*;

public class TestListSet{

public static void main(String args[]){

List list = new ArrayList();

list.add(“Hello”);

list.add(“Learn”);

list.add(“Hello”);

list.add(“Welcome”);

Set set = new HashSet();

set.addAll(list);

System.out.println(set.size());

}

}

A． 编译不通过 B． 编译通过，运行时异常 C． 编译运行都正常，输出3

D． 编译运行都正常，输出4

10、有下列一段代码，下面( C )语句可以确定"cat"包含在列表list中

ArrayList<String> list = new ArrayList<>();

list.add("dog");

list.add("cat");

list.add("horse");

-------------------------------------

A.list.contain("cat");

B.list.hasObject("cat");

C.list.indexOf("cat");

D.list.indexOf(1);

11、 选择正确答案。 （C）

import java.util.\*;

public class TestListSet{

public static void main(String args[]){

List list = new ArrayList();

list.add(“Hello”);

list.add(“Learn”);

list.add(“Hello”);

list.add(“Welcome”);

Set set = new HashSet();

set.addAll(list);

System.out.println(set.size());

}

}

A． 编译不通过

B． 编译通过，运行时异常

C． 编译运行都正常，输出3

D． 编译运行都正常，输出4

12、下面属于现成安全的类是( D )

A ArrayList B LinkedList C HashSet D Vector

13、欲构造ArrayList类的一个实例，此类继承了List接口，下列哪个方法是正确的？（ B ）

1. ArrayList myList=new Object（）； B、List myList=new ArrayList（）

C、ArrayList myList=new List（）； D、List myList=new List（）；

14.list是一个ArrayList的对象，哪个选项的代码填写到//todo delete处，可以在Iterator遍历的过程中正确并安全的删除一个list中保存的对象？(  D  )

Iterator it = list.iterator();

  int index = 0;

while (it.hasNext()){

              Object obj = it.next();

if (needDelete(obj)) { //needDelete返回boolean，决定是否要删除            //todo delete

}

              index ++;

       }

1. list.remove(obj); B）list.remove(index); C）list.remove(it.next()); D）it.remove();

15.ArrayList list = new ArrayList(20);中的list扩充几次（A）

A 0 B 1 C 2 D 3

16、欲构造ArrayList类的一个实例，此类继承了List接口，下列方法中，正确的是（B）

A. ArrayList myList = new Object（） B. List myList = new ArrayList()

C. ArrayList myList = new List（） D. List myList = new List（）

17、list是一个ArrayList的对象，当将选项（A）的代码填到//todo delelte处时，可以Iterator遍历的过程中正确并安全地删除一个list中保存的对象。

Iterator it = list.iterator();

Int index = 0;

While(it.hasNext()){

Object obj = it.next();

If(needDelete(obj)) { //needDelete返回boolean，决定是否要删除

//todo delete

}

Index++;

}

A.it.remove(); B.list.remove(index); C.list.remove(obj) D.list.delete(index)

1. 多选题
2. 以下哪些是Collection接口的子接口？（B、D）

A、Dictionary B、 List C、Map D、Set

2、下面哪些项描述不正确？（A、B）

A. List与Set一样不允许存储重复数据 B. Set无法存储实体类

C. List可以存储List子集 D. List允许冒泡排序

3、collection/set/list的联系和区别？（A、B、C、D）

A. collection是Java集合顶级接口，存储一组不唯一，无序的对象；

B. list接口和set接口是collections接口的两个子接口；

C. list存储一组不唯一，有序（插入顺序）的对象；

D. Set接口存储一组唯一，无序的对象；

4. set和list的特点与区别（ABC）

A. list 是有序的，使用此接口能够精确的控制每个元素插入的位置。

B. list能够使用索引来访问list中的元素，这类似于Java的数组。

C. set接口存储一组唯一、无序的对象

D. 操作数据的方法与list不同

5、List集合的遍历方式有如下哪几种（A、B、C）

A． Iterator迭代器实现 B． 增强for循环实现 C． get()和size()方法结合实现

D． get()和length()方法结合实现

6、 现有：list是一个合法的集合引用getCollection（A,B）返回一个合法集合的引用，以下语句哪些是合法的？

A. for(Object o : list)

B. for(Object o : getCollection())

C. for(Object o : list.iterator())

D. for(Iterator I;list.iterator(); i.hasNext())

7、(集合框架)下面不属于线程安全的类是( ACD )。

A.ArrayList  B.Vector  C.HashMap  D.Hashtable

8、下面哪些接口直接继承自Collection接口？（A、B）

A List B Map C Set D Iterator

9、下列说法正确的是（AC）

A LinkedList继承自List B AbstractSet继承自Set

C HashSet继承自AbstractSet D WeakMap继承自HashMap

10、下列接口中，直接继承自Collection接口的是？（A、C）

A. List B. Map C..list D .TreeSet

11. 下列Collection类结构的描述中，正确的是（A、C）

A． HashSet继承自Abstract B. AbstractSet继承自Set

C. LinkedList继承自List D. WeakMap继承自HashMap;

1. 填空题（每空1分，共10分）
2. 在Set中判定添加的元素是否和现有元素重复主要依赖两个方法： \_\_\_\_\_\_\_\_\_\_ 、\_\_\_\_\_\_\_\_\_。(equals、hashCode)

2、List与Set分为四种接口类：\_\_\_\_\_\_\_\_\_、 \_\_\_\_\_\_\_\_\_ 、 \_\_\_\_\_\_\_\_\_、ArrayList。(LinkedList，HashSet，TreeSet)

3、为快速查找设计的 Set 。存入 HashSet 的对象必须定义 hashCode() 。

4、 ArrayList 是由数组实现的 List

5、Collection、List、Set、Map都是\_\_\_\_\_\_\_\_\_\_ ，不能实例化。（接口）

6、Set 接口的特点是元素 无序 （有|无）顺序， 不可以 （可以|不可以）重复；

7、List 接口的特点是元素\_\_\_\_ \_（有|无）顺序，\_\_\_ \_\_\_（可以|不可以）重复 （有，可以）

8、List接口的特点是元素\_\_\_\_序,\_\_\_\_\_重复(有,可)

9、Linkedlist获取表头的方法有:linkedlist.element(),\_\_\_\_\_\_\_,\_\_\_\_\_\_,\_\_\_\_\_\_,linkedlist.pop().

答案:linkedlist.getFirst();linkedlist.peek();linkedlist.pool();

10、\_\_\_\_\_\_\_\_\_包包含了Collection的接口和类的API。(java.util)

11、 在Java语言中，位于集合框架的顶层的接口是 。（Collection）

12、 ArrayList al = new ArrayList(20)中的list扩充了 次。（0）

1. 判断题
2. Set接口的特点是不允许重复元素，元素无顺序。（ T ）
3. 当equals方法被重写时，通常有必要重写 hashCode 方法，以维护 hashCode 方法的常规协定，该协定声明相等对象必须具有相等的哈希码。（T）
4. **Set只能由迭代器访问其组成子元素。（ Y ）**
5. LinkedList是线程序安全的。（ F ）
6. empty 方法测试堆栈是否为空(T)
7. ArrayList是线程安全的类。（ F ）
8. LinkedList类可用于创建链表数据结构的对象。（ T ）
9. Collections通过add添加元素。（ F ）、
10. Set中可以存储多个null。（F）
11. ArrayList比LindedList的操作效率高。（F）
12. LinkedList通过调用poll实现FIFO。（T）
13. Set集合通过get(int index) 获取某个元素。（F）
14. List集合通过set方法修改元素信息。（T）
15. 泛型只能是引用数据类型。（T）
16. Iterator迭代器通过next()获取当前元素。（T）
17. Collection 是最基本的集合接口 （T）
18. List接口是一个有序的 Collection（T）
19. Set 具有与 Collection 完全一样的接口，只是行为上不同，Set 不保存重复的元素（T）
20. List与set接口都继承自Collection接口(正确)
21. Set接口的特点是元素无序不可重复(正确)
22. set集合是线程不安全的集合,里面存储的数据是有序且不重复的(F)
23. compareTo(Object obj)是Collection接口定义的。（ F ）

五、简答题

1、下列代码执行结果是？说明原因。

class RectObject {

public int x;

public int y;

public RectObject(int x, int y) {

this.x = x;

this.y = y;

}

@Override

public int hashCode() {

final int prime = 31;

int result = 1;

result = prime \* result + x;

result = prime \* result + y;

return result;

}

@Override

public boolean equals(Object obj) {

if (this == obj)

return true;

if (obj == null)

return false;

if (getClass() != obj.getClass())

return false;

final RectObject other = (RectObject) obj;

if (x != other.x) {

return false;

}

if (y != other.y) {

return false;

}

return true;

}

}

public class Example {

public static void main(String[] args) {

HashSet<RectObject> set = new HashSet<RectObject>();

RectObject r1 = new RectObject(3, 3);

RectObject r2 = new RectObject(5, 5);

RectObject r3 = new RectObject(3, 6);

set.add(r1);

set.add(r2);

set.add(r3);

r3.y = 7;

System.out.println("删除前的大小size:" + set.size());

set.remove(r3);

System.out.println("删除后的大小size:" + set.size());

}

}

答案：

删除前的大小size:3

删除后的大小size:3

hashCode和equals导致的集合内存泄漏。

在调用remove方法的时候，会先使用对象的hashCode值去找到这个对象，然后进行删除，这种问题就是因为我们在修改了r3对象的y属性的值，又因为RectObject对象的hashCode方法中有y值参与运算,所以r3对象的hashCode就发生改变了，所以remove方法中并没有找到r3了，所以删除失败。即r3的hashCode变了，但是他存储的位置没有更新，仍然在原来的位置上，所以当我们用他的新的hashCode去找肯定是找不到了。

2、Collection和Collections的区别，Array和Arrays的区别

答案：java.util.Collection 是一个集合接口。它提供了对集合对象进行基本操作的通用接口方法。Collection接口在Java 类库中有很多具体的实现。Collection接口的意义是为各种具体的集合提供了最大化的统一操作方式。Collections是针对集合类的一个帮助类，他提供一系列静态方法实现对各种集合的搜索、排序、线程安全化等操作。

Array类提供了动态创建和访问 Java 数组的方法。

Arrays包含用来操作数组（比如排序和搜索）的各种方法。此类还包含一个允许将数组作为列表来查看的静态工厂。

3、Vector、ArrayList、LinkedList的区别

答案：ArrayList 和Vector是采用数组方式存储数据，此数组元素数大于实际存储的数据以便增加和插入元素，都允许直接序号索引元素，但是插入数据要设计到数组元素移动等内存操作，所以索引数据快插入数据慢，Vector由于使用了synchronized方法（线程安全）所以性能上比ArrayList要差，LinkedList使用双向链表实现存储，按序号索引数据需要进行向前或向后遍历，但是插入数据时只需要记录本项的前后项即可，所以插入数度较快！

4、请简述List与Set的主要特点（5分）

答案：1.List接口存储顺序有序，并可以保存重复元素

2.Set存储顺序无序，且不可以保存重复元素

5、Java中的集合，总结list，set之间的区别。 (分别写出2条即可)（5分）

答案：

一. List：有序集合，序列，可以重复

ArrayList:

1.允许null值，不同步，可调整大小。

2.在内存中分配连续的空间，在遍历元素和随机访问元素时效率高。

3.如果多个线程同时访问ArrayList 实例，并且至少有一个线程在结构上修改列表，那么它必须在外部进行同步。

LinkedList：

1.允许null值.不同步，List和Decode接口的双链表实现，与ArrayList类似。

2.链表存储方式，在插入，删除元素时效率高，提供额外的addFist(),addlast(),removeFirst(),removeLast()，索引到列表中的操作将从开头或结尾遍历列表，以较接近指定索引为准。

3.如果多个线程同时访问LinkedList实例，并且至少有一个线程在结构上修改列表，那么它必须在外部进行同步。

Vector：

1.实现可扩展的对象数组。（像数组一样，它包含可以使用整数索引访问的组件。但是，Vector矢量的大小可根据需要增大或缩小，以便在创建“矢量”后添加和移除项目。）

2.Vector中最后一个元素后面的任何数组元素都为null。

3.存储矢量分量的数组缓冲区, 矢量的容量是这个数组缓冲区的长度，并且至少足够大以包含所有矢量的元素。

Stack：类表示后进先出（LIFO）对象堆栈。

二. Set：一个不包含重复元素的集合，且最多有一个null

HashSet：允许null，不同步

6、List、Set两个接口，存取元素时，各有什么特点？

答案：List与Set都是单列元素的集合，它们有一个功共同的父接口Collection。

Set里面不允许有重复的元素，

存元素：add方法有一个boolean的返回值，当集合中没有某个元素，此时add方法可成功加入该元素时，则返回true；当集合含有与某个元素equals相等的元素时，此时add方法无法加入该元素，返回结果为false。

取元素：没法说取第几个，只能以Iterator接口取得所有的元素，再逐一遍历各个元素。

List表示有先后顺序的集合，

存元素：多次调用add(Object)方法时，每次加入的对象按先来后到的顺序排序，也可以插队，即调用add(int index,Object)方法，就可以指定当前对象在集合中的存放位置。

取元素：方法1：Iterator接口取得所有，逐一遍历各个元素

        方法2：调用get(index i)来明确说明取第几个。

7、什么类实现了List接口？什么类实现了Set接口？

实现List接口的类：ArrayList、LinkedList、Vector

实现Set接口的类：HashSet、LinkedHashSet、TreeSet

8、Collection和Collections的区别（5分）

答案：Collection:是单列集合的顶层接口，有子接口List和Set。

Collections:是针对集合操作的工具类，有对集合进行排序和二分查找的方法

9、说出ArrayList,Vector, LinkedList的存储性能和特性

答案：ArrayList 和Vector都是使用数组方式存储数据，此数组元素数大于实际存储的数据以便增加和插入元素，它们都允许直接按序号索引元素，但是插入元素要涉及数组元素移动等内存操作，所以索引数据快而插入数据慢，Vector由于使用了synchronized方法（线程安全），通常性能上较ArrayList差，而LinkedList使用双向链表实现存储，按序号索引数据需要进行前向或后向遍历，但是插入数据时只需要记录本项的前后项即可，所以插入速度较快。

10、Set和List的区别

a. Set 接口实例存储的是无序的，不重复的数据。List 接口实例存储的是有序的，可以重复的元素

b. Set检索效率低下，删除和插入效率高，插入和删除不会引起元素位置改变 <实现类有HashSet,TreeSet>

c. List和数组类似，可以动态增长，根据实际存储的数据的长度自动增长List的长度。查找元素效率高，插入删除效率低，因为会引起其他元素位置改变 <实现类有ArrayList,LinkedList,Vector>

11 介绍Collection框架的结构

集合是Java中的一个非常重要的一个知识点，主要分为List、Set、Map、Queue三大数据结构。它们在Java中的结构关系如下：

Collection接口是List、Set、Queue的父级接口。

Set接口有两个常用的实现类：HashSet和TreeSet。List接口的常用接口有ArrayList和Vector接口。

Map接口有两个常用的实现类：Hashtable和HashMap。

12. ArrayList和Vector的区别（是否有序、是否重复、数据结构、底层实现）

ArrayList和Vector都实现了List接口，他们都是有序集合，并且存放的元素是允许重复的。它们的底层都是通过数组来实现的，因此列表这种数据结构检索数据速度快，但增删改速度慢。

而ArrayList和Vector的区别主要在两个方面：

第一，线程安全。Vector是线程安全的，而ArrayList是线程不安全的。因此在如果集合数据只有单线程访问，那么使用ArrayList可以提高效率。而如果有多线程访问你的集合数据，那么就必须要用Vector，因为要保证数据安全。

第二，数据增长。ArrayList和Vector都有一个初始的容量大小，当存储进它们里面的元素超过了容量时，就需要增加它们的存储容量。ArrayList每次增长原来的0.5倍，而Vector增长原来的一倍。ArrayList和Vector都可以设置初始空间的大小，Vector还可以设置增长的空间大小，而ArrayList没有提供设置增长空间的方法。

13. List 和 Map 区别？（数据结构，存储特点）

这个要从两个方面来回答，一方面是List和Map的数据结构，另一方面是存储数据的特点。在数据结构方面，List存储的是单列数据的集合，而Map存储的是key、value类型的数据集合。在数据存储方面，List存储的数据是有序且可以重复的，而Map中存储的数据是无序且key值不能重复（value值可以重复）

14. 如果要把实现类由ArrayList 换为LinkedList，应该改哪里？ArrayList 和LinkedList 使用上有什么区别？实现上有什么区别？

只需要中 List list =new ArrayList(); 改变为 List list =new LinkedList(); 即可

arrayList 是数组列表， 其特点是查询速度快 但是删除和添加的速度很慢。

LinkedList 表是链表的列表 其特点是查询速度慢 但是删除和添加数据速度快

15. 如果要把实现类由ArrayList 换为Vector，应该改哪里？ArrayList 和Vector 使用上有什么区别？实现上有什么区别？

与数组一样，它包含可以使用整数索引进行访问的组件。但是，Vector 的大小可以根据需要增大或缩小

16. Array与ArrayList有什么区别

①Array是Java中的数组，声明数组有三种方式

在定义一个数组的时候，必须指定这个数组的数据类型及数组的大小，也就是说数组中存放的元素个数固定并且类型一样

②ArrayList是动态数组,也就是数组的复杂版本，它可以动态的添加和删除元素，被称为”集合“，在不使用泛型的情况下，这个list是可以添加进不同类型的元素的，而且arraylist是可以不用指定长度的。在使用泛型时，我们就只能添加一种类型的数据了

17、请详细说明List与Set接口中常用的实现类

答案：ArryList线性表 适合用于查询操作 不适合做插入与删除操作

LinkdedList双向循环链表合适头尾节点的添加删除操作

TreeSet二叉数据结构在判断元素是否重复会调用对象的compareTo方法进行比较

HashSet散列结构在判断元素是否重复时会调用对象的equalse方法进行比较

18、Collection 和 Collections的区别

答案: Collection是集合类的上级接口，继承与他的接口主要有Set 和List.

Collections是针对集合类的一个帮助类，他提供一系列静态方法实现对各种集合的搜索、排序、线程安全化等操作。

19、Java 中 Set 与 List 有什么不同?

· Set是一个不允许重复元素存在的集合

· Set没有索引

· Set仅仅允许一个null值

· Set有类：HashSet、LinkedHashMap、TreeSet

· List有索引

· List允许N个null值

· List可以按插入顺序显示

· List有类：Vector、ArrayList、LinkedList

20、Collection和Collecitons的区别？

java.util.Collection 是一个集合接口。它提供了对集合对象进行基本操作的通用接口方法。Collection接口在Java 类库中有很多具体的实现。Collection接口的意义是为各种具体的集合提供了最大化的统一操作方式。

java.util.Collections 是一个包装类。它包含有各种有关集合操作的静态多态方法。此类不能实例化，就像一个工具类，服务于Java的Collection框架。

21、ArrayList、Vector、LinkedList有什么特点？

答案：ArrayList、Vector、LinkedList类均在Java.util包中，都是可伸缩的数组，即可以动态改变长度的数组。

ArrayList和Vector都是基于存储元素的Object[] array来实现的，它们会在内存中开辟一块连续的空间来存储。由于数据存储实连续的，因此，它们支持用序号（下表、索引）来访问元素，同时，索引数据的速度比较快。但是在插入元素的时候需要移动容器中的元素，所以，对数据的插入操作执行速度比较慢。ArrayList和Vector都有一个初始化的容量的大小，当里面存储的元素超过这个大小的时候，就需要动态的扩充它们的存储空间。为了提高程序的效率，每次扩充容量的时候，不是简单地扩充一个存储单元，而是一次就会增加多个存储单元，Vector默认扩充为原来的两倍（每次扩充空间的大小是可以设置的），而ArrayList默认扩充为原来的1.5倍（没有提供方法来设置空间扩种的方法）。

ArrayList与Vector最大的区别就是Synchronization（同步）的使用，没有一个ArrayList的方法是同步的，而Vector的绝大多数方法（例如add、insert、remove、set、equals、hashCode等）都是直接或者间接同步的，所以，Vector是线程安全的。ArrayList不是线程安全的。正是由于Vector提供了线程安全的机制，使其性能上也要略逊于ArrayList。

L0inkedList是采用双向列表来实现的，对数据的索引需要从列表头开始遍历，因此，随机访问的效率比较低，但是插入元素的时候不需要对数据进行移动，插入的效率较高。同时，LinkedList不是线程安全的。

在实际使用中，当对数据的主要操作为索引或只是在集合的末端增加、删除元素，使用ArrayList或Vector效率比较高。当对数据的操作主要为指定位置的插入和删除操作，使用LinkedList效率比较高。当在多线程中使用容器时（即多个线程同时访问该容器），选用Vector较为安全。

六、程序实现题

1、读取文本文件中的用户列表数据，保存为Java中的List集合元素，根据用户的年龄排序后依次输出

|  |
| --- |
| public class Test {  public static void main(String[] args) {  try {  BufferedReader br = new BufferedReader(new FileReader("Users.txt"));  String user = br.readLine();  ArrayList<User> ulist = new ArrayList<User>();  while ((user = br.readLine()) != null) {  String[] userParamer = user.split(",");  User userObj = new User(userParamer[0],  Integer.parseInt(userParamer[1]),  Integer.parseInt(userParamer[2]));  ulist.add(userObj);  }  Collections.sort(ulist, new UserComparator());  for (User u : ulist) {  System.out.println(u);  }  } catch (Exception ex) {  ex.printStackTrace();  }  }  }  class User {  private String name;  private int age;  private int height;  public String getName() {  return name;  }  public void setName(String name) {  this.name = name;  }  public int getAge() {  return age;  }  public void setAge(int age) {  this.age = age;  }  public int getHeight() {  return height;  }  public void setHeight(int height) {  this.height = height;  }  public User(String name, int age, int height) {  this.name = name;  this.age = age;  this.height = height;  }  @Override  public String toString() {  // TODO Auto-generated method stub  return name + "\t" + age + "\t" + height;  }  }  class UserComparator implements Comparator<User> {  @Override  public int compare(User o1, User o2) {  // TODO Auto-generated method stub  return o1.getAge() - o2.getAge();  }  } |

2、迭代遍历：创建一个Set容器，存放若干值并利用迭代器遍历访问

（10分）

|  |
| --- |
| public class Test {    public static void main(String[] args) {  Set set=new HashSet();  set.add("111");  set.add("222");  set.add("333");  Iterator it=set.iterator();  while(it.hasNext()){  System.out.println(it.next());  }      }  } |

3、有两个集合A、B，在A集合去除B集合中People对象的数据

public class People {

private String name;

private int age

}

public static ArrayList<People> removeDuplicteUsers(ArrayList<People> userList) {

Set<People> s = new TreeSet<People>(new Comparator<People>() {

@Override

public int compare(People o1, People o2) {

return o1.getName().compareTo(o2.getName());

}

});

s.addAll(userList);

return new ArrayList<People>(s);

}

4、Java中数组、List、Set使用代码实现互相转换

|  |
| --- |
| 数组转Set  String[] staffs = new String[]{"Tom", "Bob", "Jane"};  Set<String> staffsSet = new HashSet<>(Arrays.asList(staffs));  staffsSet.add("Mary"); // ok  staffsSet.remove("Tom"); // ok  List转数组  String[] staffs = new String[]{"Tom", "Bob", "Jane"};  List staffsList = Arrays.asList(staffs);  Object[] result = staffsList.toArray();  List转Set  String[] staffs = new String[]{"Tom", "Bob", "Jane"};  List staffsList = Arrays.asList(staffs);  Set result = new HashSet(staffsList);  Set转数组  String[] staffs = new String[]{"Tom", "Bob", "Jane"};  Set<String> staffsSet = new HashSet<>(Arrays.asList(staffs));  Object[] result = staffsSet.toArray();  Set转List  String[] staffs = new String[]{"Tom", "Bob", "Jane"};  Set<String> staffsSet = new HashSet<>(Arrays.asList(staffs));  List<String> result = new ArrayList<>(staffsSet); |

5、请完成下面编程

（10分）

|  |
| --- |
| import java.util.\*;  public class TestList{  public static void main(String args[]){  List list = new ArrayList();  list.add(“Hello”);  list.add(“World”);  list.add(1, “Learn”);  list.add(1, “Java”);  printList(list);  }  public static void printList(List list){  //1  }  }  把//1 处的代码补充完整，要求用迭代器输出list 中所有元素的内容  Iterator it=list.iterator();  while(it.hasNext())  System.out.println(it.next()); |

6、如何判断两个集合是否有交集，并打印出他们的交集

|  |
| --- |
| public static void main(String[] args) {  HashSet<String> hs = new HashSet<String>();  hs.add("George");  hs.add("Jim");  hs.add("Blake");  hs.add("Kevin");  hs.add("Mecheal");  hs.add("John");  HashSet<String> hs2 = new HashSet<String>();  hs2.add("George");  hs2.add("Kate");  hs2.add("Kevin");  hs2.add("Mecheal");  hs2.add("Ryan");  hs.retainAll(hs2);  if (hs.size() == 0) {  System.out.println("没有交集");  } else {  System.out.println("有交集");  }  for (String s : hs) {  System.out.println(s);  }  } |

7、已知有十六支男子足球队参加2008 北京奥运会。写一个程序，把这16 支球队随机分为4 个组。采用List集合和随机数

 2008 北京奥运会男足参赛国家：

科特迪瓦，阿根廷，澳大利亚，塞尔维亚，荷兰，尼日利亚、日本，美国，中国，新西 兰，巴西，比利时，韩国，喀麦隆，洪都拉斯，意大利

package com.jihelianxi;

import java.util.ArrayList;

import java.util.List;

import java.util.Random;

public class Zuqiufenlei {

public static void main(String[] args) {

Random ra=new Random();

List<String> li=new ArrayList<>();

li.add("科特迪瓦");

li.add("阿根廷");

li.add("澳大利亚");

li.add("塞尔维亚");

li.add("荷兰");

li.add("尼日利亚");

li.add("日本");

li.add("美国");

li.add("中国");

li.add("新西 兰");

li.add("巴西");

li.add("比利时");

li.add("韩国");

li.add("喀麦隆");

li.add("洪都拉斯");

li.add("意大利");

for(int i=1;i<=4;i++)

{

System.out.println("第"+i+"组:");

for(int j=0;j<4;j++)

{

String str=li.get(ra.nextInt(li.size()));

System.out.print(str+" ");

li.remove(str);

}

System.out.println();

}

}

}

8.  （List）写一个函数reverseList，该函数能够接受一个List，然后把该List 倒序排列。 例如：

List list = new ArrayList();

list.add(“Hello”);

list.add(“World”);

list.add(“Learn”); //此时list 为Hello World Learn

reverseList(list); //调用reverseList 方法之后，list 为Learn World Hello

package com.jihelianxi;

import java.util.ArrayList;

import java.util.List;

public class Lianxi4 {

public void reverseList(List<String> list)

{

List<String> li=new ArrayList<>();

for(int i=list.size()-1;i>=0;i--){

li.add(list.get(i));

}

System.out.println("list="+li);

}

public static void main(String[] args) {

List<String> list = new ArrayList<>();

list.add("Hello");

list.add("World");

list.add("Learn");

System.out.println("list="+list.toString());

Lianxi4 lx=new Lianxi4();

lx.reverseList(list);

}

}

9、重新定义List<String>集合中现有以下元素Tom Jerry Andy 请按字符串长度 对其中元素进行排序

（10分）

|  |
| --- |
| public class Test {  public static void main(String[] args) {  List <String>lists=new ArrayList<String>();  Lists.add(“jerry”);  Lists.add(“tom”)  Lists.add(“andy”)  Collections.sort(lists,new CompareTor(){  Public int compare(String o1,Stringo2){  Return o1.length-o2.length  }  })  } |

1. 10、使用ArrayList集合存储数据,获取10个1-20之间的随机数，要求不能重复

public static void main(String[] args) {

  Random r=new Random();//创建随机数生成器对象

    ArrayList<Integer> al=newArrayList<Integer>();//创建ArrayList集合对象,该集合泛型为Integer类型

   int count=0;//定义统计变量

   while(count<10){

     intresult=r.nextInt(20)+1;//产生1-20之间的随机数

    if(!al.contains(result)){

     al.add(result);

        count++;

        }

    }

  System.out.println("使用ArrayList集合产生10个1-20之间不重复的随机数");

   for(Integer i:al){

    System.out.print(i + " ");

   }

}

11、ListIterator是List集合特有的迭代器，是Iterator的子接口，在迭代时，不可以通过集合对象的方法操作集合中的元素。因为会发生ConcurrentModificationException异常。所以，在迭代器时，只能用迭代器的方法操作元素，可是Iterator方法是有限的，只能对元素进行判断，取出，删除的操作，如果想要其他的操作如添加，修改等，就需要使用其子接口ListIterator。请使用ListIterator编写添加和修改操作。

|  |
| --- |
| import java.util.\*;  class ListDemo  {  public static void sop(Object obj)  {  System.out.println(obj);  }  public static void method()  {    ArrayList al = new ArrayList();    //添加元素  al.add("java01");  al.add("java02");  al.add("java03");    sop("原集合是："+al);  //在指定位置添加元素。  al.add(1,"java09");    //删除指定位置的元素。  //al.remove(2);    //修改元素。  //al.set(2,"java007");    //通过角标获取元素。  sop("get(1):"+al.get(1));    sop(al);    //获取所有元素。  for(int x=0; x<al.size(); x++)  {  System.out.println("al("+x+")="+al.get(x));  }    Iterator it = al.iterator();    while(it.hasNext())  {  sop("next:"+it.next());  }      //通过indexOf获取对象的位置。  sop("index="+al.indexOf("java02"));    List sub = al.subList(1,3);    sop("sub="+sub);  }      public static void main(String[] args)  {    //演示列表迭代器。  ArrayList al = new ArrayList();    //添加元素  al.add("java01");  al.add("java02");  al.add("java03");    sop(al);      ListIterator li = al.listIterator();    //修改元素  while(li.hasNext())  {  Object obj = li.next();    if(obj.equals("java02"))  li.set("java006");    }    while(li.hasPrevious())  {  sop("pre::"+li.previous());  }    sop(al);      }  } |