|  |
| --- |
| 专业班级 信息173班 学号 201711010228 姓名 苏栋 |

**实验代码：**

package com.keda;

import java.util.\*;

public class ListApp

{

public static void main(String[] args)

{

// List集合 有序的 可添加重复元素 可以放null值

ArrayList a = new ArrayList();

//向集合中添加元素0

a.add(0);

//向集合中添加元素90

a.add(90);

//向集合中添加元素60

a.add(60);

//向集合中添加元素80

a.add(80);

//向集合中添加元素hello字符串

a.add("hello");

//向集合中添加元素90

a.add(90);

//向集合中添加元素null

a.add(null);

Vector v = new Vector();

//创建一个array数组

List b = new ArrayList();

//想数组中添加元素100

b.add(100);

//想数组中添加元素150

b.add(150);

//想数组中添加元素200

b.add(200);

// 将一个集合添加到另外一个集合中

a.addAll(b);

// 将一个集合添加到另外一个集合的指定的位置

// a.addAll(5, b);

// 清除集合中的元素

// a.clear();

// 判断集合中是否包含hello这个元素

// System.out.println(a.contains("hello"));

}

// 判断两个集合中的元素是否相等

// System.out.println(a.equals(b));

// 根据索引获得元素

// System.out.println(a.get(0));

// 返回集合的哈希吗值

// System.out.println(a.hashCode());

// 根据元素获得索引

// System.out.println(a.indexOf("hello"));

// 判断集合是否为空

// System.out.println(a.isEmpty());

// 根据索引删除元素

// a.remove(0);

// 删除对象

// a.remove("hello");

// a.remove(new Integer(60));

// 从a集合中删除b集合

// a.removeAll(b);

// 获得迭代器的接口对象

// Iterator iter=a.iterator();

// //如果仍有元素可以迭代，则返回 true。

// while(iter.hasNext()){

// //返回迭代的下一个元素。

// System.out.println(iter.next());

// }

// 获得集合的长度

// System.out.println(a.size());

// 截取集合中的元素组成新集合

// List c=a.subList(2, 4);

// for(Object obj:c){

// System.out.println(obj);

// }

// 将集合转换为数组

// Object[] obj=a.toArray();

// for (int i = 0; i < obj.length; i++)

{

// System.out.println(obj[i]);

// }

int[] i = new int[25];

for (int h = 0; h < i.length; h++)

{

i[h] = h + 1;

System.out.println(i[h]);

}

int[] j = new int[25];

// 将指定的 int 值分配给指定 int 型数组的每个元素。

// Arrays.fill(i, 47);

Arrays.fill(j, 99);

// 从指定源数组中复制一个数组，复制从指定的位置开始，到目标数组的指定位置结束。

// 源数组名称 源数组下标开始复制 目标数组 目标数组填充的下标 指定索引的位置

System.arraycopy(i, 5, j, 1, 4);

for (int h : j) {

System.out.println(h);

}

//创建一个int数组

int[] k = new int[10];

Arrays.fill(k, 103);

//将i数组从第一个元素开始及之后的元素拷贝到k数组

System.arraycopy(i, 0, k, 0, k.length);

//将k数组中的元素填充为103

Arrays.fill(k, 103);

//将k数组从第一个元素开始及之后的元素拷贝到i数组

System.arraycopy(k, 0, i, 0, k.length);

Integer[] u = new Integer[10];

Integer[] v2 = new Integer[5];

//将数组中元素全部填充为47

Arrays.fill(u, new Integer(47));

//将数组中元素全部填充为99

Arrays.fill(v2, new Integer(99));

System.arraycopy(v2, 0, u, u.length / 2, v2.length);

}

}