### Java第一阶段—DAY15-JAVA案例

1. 定义一个List集合,添加元素“关羽”、“张飞”、“刘备”、"赵云"、"诸葛亮"，分别进行如下操作：
2. 获取索引为2的元素。
3. 打印集合的长度
4. 、判断赵云是否在集合中
5. 删除诸葛亮元素
6. 把集合转换成数组并且遍历
7. 清空集合
8. 打印集合是否为空
9. **public class** Demo01ArrayList {  
     
    **public static void** main(String[] args) {  
   *// 创建集合容器* List<String> list = **new** ArrayList<>();  
    list.add(**"关羽"**);*// 添加元素关羽，索引为0* list.add(**"张飞"**);*// 添加元素张飞，索引为1* list.add(**"刘备"**);*// 添加元素刘备，索引为2* list.add(**"赵云"**);*// 添加元素赵云，索引为3* list.add(**"诸葛亮"**);*// 添加元素诸葛亮，索引为4* System.out.println(list);*// 打印集合信息* System.out.println(list.get(2));*// 根据索引查找对应的元素* System.out.println(list.size());*// 查看集合中一共有多少个元素* System.out.println(list.contains(**"赵云"**)); *// 判断赵云是否存在集合中* list.remove(**"诸葛亮"**);*// 删除诸葛亮  
     
    //Object[] toArray()转换成一个Object数组* Object[] objects = list.toArray();  
    *// 遍历数组* **for** (**int** i = 0; i < objects.length; i++) {  
    System.out.println(objects[i]);  
    }  
     
    list.clear();*// 清空集合* System.out.println(list.isEmpty());*// boolean isEmpty() 判断是否为空* }  
   }
10. 定义一个List集合,添加元素“关羽”、“张飞”、“刘备”、"赵云"、"诸葛亮"，至少用2种方式遍历。

public class Demo02ArrayList {  
  
 public static void main(String[] args) {  
*// 创建集合容器* List<String> list = new ArrayList<>();  
 list.add("关羽");*// 添加元素关羽，索引为0* list.add("张飞");*// 添加元素张飞，索引为1* list.add("刘备");*// 添加元素刘备，索引为2* list.add("赵云");*// 添加元素赵云，索引为3* list.add("诸葛亮");*// 添加元素诸葛亮，索引为4  
  
// 遍历方式1* for (int i=0;i<list.size();i++){  
 System.out.println(list.get(i));  
 }  
  
*// 遍历方式2* for (String name : list){  
 System.out.println(name);  
 }  
  
 }  
}

1. 创建LinkedList对象，并测试其相关API。

public class Demo03LinkedList {  
  
 public static void main(String[] args) {  
 LinkedList<String> link = new LinkedList<String>();  
 *//添加元素* link.addFirst("关羽");  
 link.addFirst("张飞");  
 link.addFirst("刘备");  
  
 for (String s : link) {  
 System.out.println(s);  
 }  
  
  
  
 System.out.println("======================");  
  
  
*// System.out.println(link);  
//  
// System.out.println(link.getFirst()); // 获取第1个元素  
// System.out.println(link.getLast()); // 获取最后1个元素  
//  
// System.out.println(link.removeFirst());// 删除第1个元素  
// System.out.println(link.removeLast());// 删除最后1个元素  
//  
// while (!link.isEmpty()) { //判断集合是否为空  
// System.out.println(link.pop()); //弹出集合中的栈顶元素  
// }  
//  
// System.out.println(link);* }  
}

1. 定义一个HashSet集合,添加元素“关羽”、“张飞”、“刘备”、"赵云"、"诸葛亮"，测试其常用API。

public class Demo04HashSet {  
  
 public static void main(String[] args) {  
*// 创建集合容器* Set<String> set = new HashSet<>();  
 set.add("关羽");  
 set.add("张飞");  
 set.add("刘备");  
 set.add("赵云");  
 set.add("诸葛亮");  
  
 System.out.println(set);*// 打印集合信息* System.out.println(set.size());*// 查看集合中一共有多少个元素* System.out.println(set.contains("赵云")); *// 判断赵云是否存在集合中* set.remove("诸葛亮");*// 删除诸葛亮* set.clear();*// 清空集合* System.out.println(set.isEmpty());*// boolean isEmpty() 判断是否为空* }  
}

1. 定义Person类，有name跟age属性，初始化3个Person对象，装进TreeSet对象中，测试常用API.

public class Demo05TreeSet {  
  
 public static void main(String[] args) {  
 TreeSet<Person> treeSet = new TreeSet<>();  
 treeSet.add(new Person("刘备",33));  
 treeSet.add(new Person("关羽",23));  
 treeSet.add(new Person("张飞",27));  
 }  
  
}  
  
class Person implements Comparable<Person>{  
  
 private String name ;  
 private int age ;  
  
 public Person(String name, int age) {  
 this.name = name;  
 this.age = age;  
 }  
  
 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;  
 }  
  
 @Override  
 public String toString() {  
 return "Person{" +  
 "name='" + name + '\'' +  
 ", age=" + age +  
 '}';  
 }  
  
 @Override  
 public int compareTo(Person person) {  
 return this.age - person.age;  
 }  
}

1. 定义一个List集合,添加元素“关羽”、“张飞”、“刘备”、"赵云"、"诸葛亮"，通过使用迭代器遍历此集合。

public class Demo06Iterator {  
  
 public static void main(String[] args) {  
   
*// 准备集合跟元素* List<String> list = new ArrayList<>();  
 list.add("关羽");  
 list.add("张飞");  
 list.add("刘备");  
 list.add("赵云");  
 list.add("诸葛亮");  
  
*// 获取集合对应的迭代器* Iterator<String> iterator = list.iterator();  
*// 获取迭代器之后修改了集合结构，会异常  
// list.add("黄忠");  
  
// 每次判断是否有下一个元素，如果有返回true，否则返回false* while (iterator.hasNext()){  
 String hero = iterator.next();  
 System.out.println(hero);  
*// 遍历过程中修改了集合的结构，会异常  
// list.remove(hero);* }  
  
 }  
}