

# 学习任务一技能节报名小程序

## 子任务1.2 信息存储



## 任务准备——了解Java中的集合





## 任务准备——了解Java中的集合

❑ Collection和Map，是集合框架的根接口。

❑ Collection的子接口

1、Set接口 --- 实现类

HashSet、LinkedHashSet

2、Set的子接口SortedSet接口 --- 实现类

TreeSet

3、List:接口 --- 实现类

LinkedList, Vector, **ArrayList**

4、Map接口 --- 实现类

HashMap、Hashtable

```
Demo_List.java
8
9     public static void main(String[] args) {
10         List list1 = new ArrayList();
11         List list2 = new ArrayList();
12         List list3 = new ArrayList();
13         list1.add("demo1");
14         list1.add("demo2");
15         list1.add("demo3");
16         list2.add("test1");
17         list2.add("test2");
18         System.out.println("没有操作前list1中所有的元素: " + list1);
19         System.out.println("没有操作前list2中所有的元素: " + list2);
20         System.out.println("没有操作前list3中所有的元素: " + list3);
21         Boolean b = list1.addAll(2, list2);
22         System.out.println("list1插入list2后, list1所有的元素: " + list1);
23         System.out.println("list1插入list2的返回值是: " + b);
24         b = list1.addAll(2, list3);
25         System.out.println("list1插入list3后, list1所有的元素: " + list1);
26         System.out.println("list1插入list3的返回值是: " + b);
27     }
```

Console

<terminated> Demo\_List [Java Application] D:\Java64\java1.8\bin\javaw.exe (2018年8月19日 下午10:31:45)

没有操作前list1中所有的元素: [demo1, demo2, demo3]

没有操作前list2中所有的元素: [test1, test2]

没有操作前list3中所有的元素: []

list1插入list2后, list1所有的元素: [demo1, demo2, test1, test2, demo3]

list1插入list2的返回值是: true

list1插入list3后, list1所有的元素: [demo1, demo2, test1, test2, demo3]

list1插入list3的返回值是: false

## 任务准备——了解Java中的集合

❑ Collection和Map，是集合框架的根接口。

❑ Collection的子接口

1、Set:接口 --- 实现类

**HashSet**、LinkedHashSet

2、Set的子接口SortedSet接口 --- 实现类

TreeSet

3、List:接口 --- 实现类

LinkedList, Vector, ArrayList

4、Map接口 --- 实现类

HashMap、Hashtable

Demo\_Set.java

```
1 package com.xiaostudy;
2
3 import java.util.HashSet;
4 import java.util.Set;
5
6 public class Demo_Set {
7
8     public static void main(String[] args) {
9         Set set = new HashSet();
10        set.add("demo1");
11        set.add("demo2");
12        System.out.println("没有添加重复元素前set中所有的元素: " + set);
13        boolean b = set.add("demo3");
14        System.out.println("添加demo3元素的返回值是: " + b);
15        b = set.add("demo1");
16        System.out.println("添加demo1元素的返回值是: " + b);
17        System.out.println("添加重复元素后set中所有的元素: " + set);
18    }
19
20 }
21
```

Console

Progress

Search

Deliver Monitor

<terminated> Demo\_Set [Java Application] D:\Java(x86)\jre1.8.0\_02\bin\javaw.exe (2018-8-21 上午11:26:28)

没有添加重复元素前set中所有的元素: [demo1, demo2]

添加demo3元素的返回值是: true

添加demo1元素的返回值是: false

添加重复元素后set中所有的元素: [demo3, demo1, demo2]

## 任务准备——了解Java中的集合

▣ Collection和Map，是集合框架的根接口。

▣ Collection的子接口

1、Set:接口 --- 实现类

HashSet、LinkedHashSet

2、Set的子接口SortedSet接口 --- 实现类

TreeSet

3、List:接口 --- 实现类

LinkedList, Vector, ArrayList

4、Map接口 --- 实现类

HashMap、Hashtable

```
1 public class MapDemo {
2     public static void main(String[] args) {
3         //创建Map对象
4         Map<String, String> map = new HashMap<String, String>(); //数据采集
5         //给map中添加元素
6         map.put("星期一", "Monday");
7         map.put("星期日", "Sunday");
8         System.out.println(map); // {星期日=Sunday, 星期一=Monday}
9
10        //当给Map中添加元素，会返回key对应的原来的value值，若key没有对应的值，返回null
11        System.out.println(map.put("星期一", "Mon")); // Monday
12        System.out.println(map); // {星期日=Sunday, 星期一=Mon}
13
14        //根据指定的key获取对应的value
15        String en = map.get("星期日");
16        System.out.println(en); // Sunday
17
18        //根据key删除元素，会返回key对应的value值
19        String value = map.remove("星期日");
20        System.out.println(value); // Sunday
21        System.out.println(map); // {星期一=Mon}
22    }
23 }
```

## 课前准备——课前观看视频

观看视频使用集合存储数据，回答如下问题

要点↵	提炼↵
集合如何创建↵	↵
什么是泛型↵	↵
如何往集合里添加对象↵	↵
如何遍历集合↵	↵



## 课堂练习：使用集合存储和遍历选手信息

请输入学生的学号：12

请输入学生的姓名：123

请输入学生的成绩：80

\*\*\*\*\*

已经报名的学生信息如下：

学号	姓名	成绩
----	----	----

12	123	80
----	-----	----

13	aa	78
----	----	----

14	ee	85
----	----	----

\*\*\*\*\*

1、利用循环重复三次输入三个学生的信息

2、用一个集合存储这三个学生信息

3、将集合中的信息打印出来

## 课堂练习：类的结构分析

