

# Day 17 集合框架

Key Point :

- Collection 接口、Set 接口基本操作
- Set 接口及其实现类
- Hash 算法与hashCode 方法
- Comparable 接口
- Map 接口及其实现类
- 遍历Map

问题：

1.填空

Collection 接口的特点是元素是\_；  
List 接口的特点是元素（有|无）顺序，\_（可以|不可以）重复；  
Set接口的特点是元素（有|无）顺序，\_（可以|不可以）重复；  
Map接口的特点是元素是\_，其中可以重复，\_不可以重复。

2.填空

Map中定义的添加方法声明是\_  
删除方法的声明\_  
获取元素个数的方法声明\_  
根据key获取value方法声明\_\_\_\_  
判断包含指定key的方法声明\_\_\_\_  
判断包含指定value的方法声明\_\_\_\_  
将map中所有key封装一个Set返回的方法声明\_\_\_\_  
将map中所有value封装程一个Collection返回的方法声明\_\_\_\_  
将map中所有键值对封装程一个Set返回的方法声明\_\_\_\_

3. ( Set , List )

```
import java.util.*;
public class TestListSet{
    public static void main(String args[]){
        List<String> list = new ArrayList<String>();
        list.add("Hello");
        list.add("Learn");
        list.add("Hello");
        list.add("Welcome");
        Set<String> set = new HashSet<String>();
        set.addAll(list);
        System.out.println(set.size());
    }
}
```

选择正确答案

- 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

4. ( Set ) 已知有一个Worker 类如下：

```

public class Worker {
    private int age;
    private String name;
    private double salary;
    public Worker (){}
    public Worker (String name, int age, double salary){
        this.name = name;
        this.age = age;
        this.salary = salary;
    }
    public int getAge() {
        return age;
    }
    public void setAge(int age) {
        this.age = age;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public double getSalary(){
        return salary;
    }
    public void setSalary(double salary){
        this.salary = salary;
    }
    public void work(){
        System.out.println(name + " work");
    }
}

```

完成下面的要求：

I. 创建一个Set，在Set 中增加三个Worker对象，基本信息如下

姓名	年龄	工资
zhang3	18	3000
li4	25	3500
wang5	22	3200

II. 删除wang5 的信息

III. 利用for-each遍历，对List 中所有的工人调用work 方法。

IV. 为Worker 类添加equals 方法 和 hashCode方法，保证Worker对象添加到HashSet中元素内容不重复，并尽可能提高效率

5. ( Set，Hash算法 ) 为上一题的Worker类，在添加完equals方法的基础上，添加一个hashCode方法。

```
public int hashCode(){  
    //1  
}
```

有几种写法：

I. return 0;

II. int result = 0;  
if (name != null) result = name.hashCode();  
return result + age;

III. return super.hashCode();

现在要把Worker类放入HashSet中，并希望在HashSet中没有重复元素，则下面说法正确的是：

- A. 三种写法都正确
- B. I、II写法正确，II的效率更高
- C. II写法正确，I、III写法都不正确

6. ( Set，Hash 算法，方法覆盖 ) 代码改错

```

import java.util.*;
class Worker{
    String name;
    int age;
    double salary;
    public Worker(){
    public Worker(String name, int age, double salary){
        this.name = name;
        this.age = age;
        this.salary = salary;
    }
    int hashCode(){
        return name.hashCode() + age + salary;
    }
    public boolean equals(Worker w){
        if (w.name == name && w.salary == salary && w.age == age){
            return true;
        }else{
            return false;
        }
    }
}
public class TestWorker{
    public static void main(String args[]){
        Set<Worker> set = new HashSet<Worker>();
        set.add(new Worker("tom", 18, 2000));
        set.add(new Worker("tom", 18, 2000));
        set.add(0, new Worker("jerry", 18, 2000));
        System.out.println(set.size());
    }
}

```

7. ( Map ) 关于下列Map 接口中常见的方法：

- I. put方法表示放入一个键值对，如果键已存在则（ ），如果键不存在则（ ）。
- II. remove方法接受（ ）个参数，表示（ ）。
- III. get方法表示（ ），get 方法的参数表示（ ），返回值表示（ ）。
- IV. 要想获得Map中所有的键，应该使用方法（ ），该方法返回值类型为（ ）。
- V. 要想获得Map中所有的值，应该使用方法（ ），该方法返回值类型为（ ）。

8. ( Map ) 利用Map，完成下面的功能：

从命令行读入一个字符串，表示一个年份，输出该年的世界杯冠军是哪支球队。如果该年没有举办世界杯，则输出：没有举办世界杯。

附录：截止2009 年，历届世界杯冠军、世界杯冠军以及对应的夺冠年份：

届数	年份	冠军
18	2006	意大利
17	2002	巴西
16	1998	法国
15	1994	巴西
14	1990	德国
13	1986	阿根廷
12	1982	意大利
11	1978	阿根廷
10	1974	德国
9	1970	巴西
8	1966	英格兰
7	1962	巴西
6	1958	巴西
5	1954	德国
4	1950	乌拉圭
3	1938	意大利
2	1934	意大利
1	1930	乌拉圭

9. ( Map ) 已知某学校的教学课程内容安排如下：

老师	课程
field1	field2
Tom	CoreJava
John	Oracle
Susan	Oracle
Jerry	JDBC
Jim	Unix
Kevin	JSP
Lucy	JSP

完成下列要求：

- I. 使用Map，以老师的名字作为键，教授的课程名作为值，表示上述课程安排。
- II. 增加了一位新老师Allen 教JDBC
- III. Lucy 改为教CoreJava
- IV. 遍历Map，输出所有的老师及老师教授的课程
- V. 利用Map，输出所有教JSP 的老师。

10. ( Set , HashSet , 空指针 ) 有下面代码

```
import java.util.*;
class Student {
    int age;
    String name;
    public Student() {}
    public Student(String name, int age){
        this.name = name;
        this.age = age;
    }
    public int hashCode(){
        return name.hashCode() + age;
    }
    public boolean equals(Object o){
        if (o == null) return false;
        if (o == this) return true;
        if (o.getClass() != this.getClass()) return false;
        Student stu = (Student) o;
        if (stu.name.equals(name) && stu.age == age) return true;
        else return false;
    }
}
public class TestHashSet{
    public static void main(String args[]){
        Set<Student> set = new HashSet<Student>();
        Student stu1 = new Student();
        Student stu2 = new Student("Tom", 18);
        Student stu3 = new Student("Tom", 18);
        set.add(stu1);
        set.add(stu2);
        set.add(stu3);
        System.out.println(set.size());
    }
}
```

下列说法正确的是：

- A. 编译错误
- B. 编译正确，运行时异常
- C. 编译运行都正确，输出结果为3
- D. 编译运行都正确，输出结果为2

13. ( Map ) 在原有世界杯Map 的基础上，增加如下功能：

读入一支球队的名字，输出该球队夺冠的年份列表。

例如：

- I. 读入 “巴西” ，应当输出1958、1962、1970、1994、2002
- II. 读入 “荷兰” ，应当输出“ 没有获得过世界杯” 。

14.(Map) 给定一个字符串,请输出该字符串由哪些字符组成,每个字符出现几次?