

## Chapter 10 集合框架

---

### Key Point :

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

### 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. （List）有如下代码

```
import java.util.*;

public class TestList{

    public static void main(String args[]){
```

```

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

        list.add( "Hello" );

        list.add( "World" );

        list.add(1, "Learn" );

        list.add(1, "Java" );

        printList(list);

    }

    public static void printList(List list){

        //1

    }

}

```

要求：

- I. 把//1 处的代码补充完整，要求输出 list 中所有元素的内容
- II. 写出程序执行的结果
- III. 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？
- IV. 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？

### 3. （List）写出下面程序的运行结果

```

import java.util.*;

public class TestList{

    public static void main(String args[]){

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

```

```

list.add( "Hello" );

list.add( "World" );

list.add( "Hello" );

list.add( "Learn" );

list.remove( "Hello" );

list.remove(0);

for(int i = 0; i<list.size(); i++){

    System.out.println(list.get(i));

}

}

```

#### 4. ( 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

5. ( List ) 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 List，在 List 中增加三个工人，基本信息如下

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

II. 在 li4 之前插入一个工人，信息为：姓名：zhao6，年龄：24，工资 3300

III. 删除 wang5 的信息

IV. 利用 for 循环遍历，打印 List 中所有工人的信息

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

VI. 为 Worker 类添加 equals 方法

6. (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

7. (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());

    }

}
```

8. ( Set , Hash 算法 ) 在前面的 Worker 类基础上 , 为 Worker 类增加相应的方法 , 使得 Worker 放入 HashSet 中时 , Set 中没有重复元素。并编写相应的测试代码。

9. ( 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\_\_\_\_\_。

10. ( 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

11. ( Map ) 已知某学校的教学课程内容安排如下：

老师	课程
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。

12. ( 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?