

# Chp10 集合框架

## 参考答案

### 1. 参考答案

Collection 接口的特点是元素是对象；

List 接口的特点是元素有（有|无）顺序，可以（可以|不可以）重复；

Set 接口的特点是元素无（有|无）顺序，不可以（可以|不可以）重复；

Map 接口的特点是元素是键值对，其中值可以重复，键不可以重复。

### 2. 参考答案

1) 略

2) Hello Java Learn World

3) 要换实现类，只需要修改 `List l = new XXXX();` 这一句即可。

ArrayList 和 LinkedList 的比较如下

ArrayList      数组实现      查询快，增删慢

LinkedList    链表实现      查询慢，增删快

4) 要换实现类，与上一问相同。

ArrayList 和 Vector 两者都是用数组实现，比较如下：

ArrayList      轻量级，速度快，线程不安全

Vector          重量级，速度慢，线程安全

### 3. 输出结果为

Hello

Learn

在调用第一个 `remove` 方法时，会删掉遇到的第一个“Hello”对象，List 中剩下的元素为： World Hello Learn

在调用第二个 `remove` 方法时，会删掉下标为 0 的元素。List 中剩下的元素为：

Hello Learn

### 4. C

该程序主要是调用 List 的 `addAll` 方法。调用该方法时，会把 List 中的所有元素都加入 Set 中。要注意的是，重复元素不会被加入。

### 5. 略

### 6. B

hashCode 正确性要求：相同对象返回相同的 hashCode。这方面，1)，2) 都满足要求。

hashCode 效率要求：不同对象返回的 hashCode 尽量不同。这方面 2) 做的更好，用到了 Worker 类的两个属性，因此效率更高。

### 7. 参考答案

```

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;
    }
    //hashCode 必须是 public 的
    //由于 Object 类中的 hashCode() 方法为 public 的
    //因此子类的覆盖方法，其修饰符必须也写成 public 的
    int hashCode(){
        //返回值类型不对。应当返回 int 类型，此处返回了 double 类型
        //需要进行强转
        return name.hashCode() + age + salary;
    }

    //equals 方法参数和实现都不对。
    //1.要覆盖 Object 类中的 equals 方法，参数必须一致
    //equals 方法签名位 public boolean equals(Object obj)
    //2.equals 方法实现有错。具体请参考上一章相关练习
    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 set = new HashSet();
        set.add(new Worker("tom", 18, 2000));
        set.add(new Worker("tom", 18, 2000));
        //Set 接口中没有定义带下标的 add 方法。
        set.add(0, new Worker("jerry", 18, 2000));
        System.out.println(set.size());
    }
}

```

8. 参考 Worker.java 和 TestWorkerHashSet.java
9. 参考 Worker.java 和 TestWorkerTreeSet.java

10. 参考答案

put 方法表示放入一个键值对，如果键已存在则新值替换旧值，如果键不存在则增加一个键值对。

remove 方法接受 1 个参数，表示 删除一个键值对。

get 方法表示 通过键查找值，get 方法的参数表示 键，返回值表示 值。

要想获得 Map 中所有的键，应该使用方法 keySet，该方法返回值类型为 Set。

要想获得 Map 中所有的值，应该使用方法 values，该方法返回值类型为 Collection。

要想获得 Map 中所有的键值对的集合，应该使用方法 entrySet，该方法返回一个 Map.EntrySet 类型所组成的 Set。

11. 参考 TestWorldCup.java

12. 参考 TestCourse.java

13. 略

14. 参考 TestCourse2.java

15. 输出结果为

10

50

30

在 List 中保存的是对象的引用，因此 mc4/mc1 与 List 中下标为 1 的三个引用都指向同一个对象。

16. B

当把一个 Student 对象放到 HashSet 中时，会调用该对象的 hashCode 方法。Student 类覆盖了 hashCode 方法，会调用 name 属性的 hashCode 方法。

对于 stu1 对象而言，由于调用了默认构造函数，其 name 属性为 null。当调用 name 属性的 hashCode 时，会产生 NullPointerException。

17. 见 ex17/目录下的 Worker.java、Address.java、TestWorker.java

18. 参考 TestWorldCup2.java

有多种解法，其中比较简单的为：对 Map 进行键值遍历，如果值（球队的名字）和用户输入的球队名字相等，则输出键。

19. 参考 ex19 目录下的 Account.java，TestAccount.java

20. 参考 ReverseList.java 文件

思路：与颠倒数组的练习一样。交换下标为 i 与 list.size() - 1 - i 的元素的值。

21. 输出结果

abc

2

注意，Map 要求键不能重复，值可以重复。HashMap 在判断键是否重复的时候，采用的是和 HashSet 类似的 Hash 算法。

在本题中，HashMap 的键由于是自定义的 MyKey 类型，没有覆盖 equals 和 hashCode 方法。因此在本题的 HashMap 中会有两个键值对。

## 22. 参考答案

1), 2) 略

3) 错误。hashCode()方法要求相同对象返回的 hashCode 值相同。而相同对象的判断依赖于 equals。在本题中，equals 方法仅判断 id，因此要求 id 相同的对象 hashCode 方法返回值必须相同。

假设有这样两个对象：

Id	Name	Salary
100001	Tom	1000
100001	Jim	2000

在程序中，equals 方法返回 true，因此要求 hashCode 方法返回值相同。如题目中所写的 hashCode 方法的实现，返回值必定不同，因此题目中给出的 hashCode 方法的实现是错误的。

注：该题目的实际意义在于，如果 equals 仅判断 id 属性的话，就能保证在某个 set 中，所有员工的 id 都是唯一的。即：不会出现两个使用同一 id 的员工。

## 23. 参考答案

1) 略

2) 参考 TestAverage.java。基本思路：创建一个 Map，该 Map 的键为班号，该 Map 的值为一个 List，List 中放的是相应班级中所有的学生。

代码中有更为详细的注释。

## 24. 参考 TestGroup.java

思路：创建一个 List：其元素为四个小 List，其中每个小 List 表示一个分组。

## 25. 参考 TestStack.java

注意：应当使用 LinkedList 来实现。因为 Stack 数据结构，查询主要查栈顶元素，因此查询的次数较少。由于入栈和出栈操作较多，即增删操作较多。因此，应当选择增删快而查询稍慢的 LinkedList 来实现。

又，为了能增加查询的效率，栈顶元素应当是下标为 0 的元素。