

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

答案：

1. 填空

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. for (Object o : list) {  
  
    System.out.println(o);  
  
}
```

II. Hello Java Learn World

III. 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

IV. 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：Hello、Learn

I. 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

II. 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

I. `hashCode` 正确性要求：相同对象返回相同哈希码。这方面 I、II 都满足要求。

II. `hashCode` 效率要求：不同对象返回的哈希码尽量不同。这方面 II 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 需要强转

        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：TestWorldCup.java

11. 参考：TestCourse.java

12. B

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

13. 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。

14. 参考：TestCharCount.java