Chp10 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

练习

1. 填空

个数可变，类型可以不同，只能是引用数据类型

Collection 接口的特点是元素是_____；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. （List）有如下代码

```
import java.util.*;
public class TestList{
    public static void main(String args[]){
        List list = new ArrayList();
        list.add(“Hello”);
        list.add(“World”);
        list.add(1, “Learn”);
        list.add(1, “Java”);
        printList(list);
    }
    public static void printList(List list){
        //1
    }
}
```

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

三：
1.改动：把new后面的ArrayList改为LinkedList
2. 使用上的区别： 操作ArrayList很慢，因为它在内部中使用数组，只要删除一个元素，在内存中的所有元素都会移动，而操作LikedList就很快，因为它在内部中使用双链表，不需要移动内存中的元素
3. 实现上的区别：ArrayList内部使用动态数组来存储元素，LikedList在内部中使用双链表来存储元素

要求：

1）把//1 处的代码补充完整，要求输出 list 中所有元素的内容

2）写出程序执行的结果

Hello Java Learn World

3）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？

4）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？

四：
1.改new后面的ArrayList为Vector
2.使用上的区别：
相同：都是动态数组的实现
不同：1.同步 Vector线程同步，ArrayList线程不同步
2.当元素个数超过初始容量时，扩容不同

Vector 容量翻倍 ArrayList增加50%

3. 版本不同

ArrarList 1.2
Vector 1.0

4. 速度不同

ArrayList 快，因为不同步

Vector 慢，因为同步

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 list = new ArrayList();
        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));
        }
    }
}
```

Hello
Learn

4. （Set，List）

```
import java.util.*;
public class TestListSet{
    public static void main(String args[]){
        List list = new ArrayList();
        list.add(“Hello”);
        list.add(“Learn”);
        list.add(“Hello”);
        list.add(“Welcome”);
        Set set = new HashSet();
        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

1) 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

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

3) 删除 wang5 的信息

4) 利用 for 循环遍历，打印 List 中所有工人的信息

5) 利用迭代遍历，对 List 中所有的工人调用 work 方法。

6) 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：

1) return 0;

2)

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

3) 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. 1)，2)写法正确，2)效率更高

C. 2)写法正确，1),3)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 set = new HashSet();
        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());
    }
}
```

public int hashCode(){ 重写方法，范围修饰符只能变大，不能变小，从 public 变为 default，范围变小了

return (int) (name.hashCode() + age + salary); // double 转为int可能会精度丢失

set.add(new Worker("jerry", 18, 2000)); hashSet没有下标，是无序的

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. （Set，Comparable 接口）在前面的 Worker 类基础上，为 Worker 类添加相应的代码， 使得 Worker 对象能正确放入 TreeSet 中。并编写相应的测试代码。
注：比较时，先比较工人年龄大小，年龄小的排在前面。如果两个工人年龄相同，则再比较其收入，收入少的排前面。如果年龄和收入都相同，则根据字典顺序比较工人姓名。例
如：有三个工人，基本信息如下：

姓名	年龄	工资
zhang3	18	1500
li4	18	1500
wang5	18	1600
zhao6	17	2000

放入 TreeSet 排序后结果为：
zhao6 li4 zhang3 wang5

10. （Map）关于下列 Map 接口中常见的方法
put 方法表示放入一个键值对，如果键已存在则__覆盖，返回旧值__，如果键不存在则__返回null，并存入__。 remove 方法接受_1_个参数，表示_根据键删除映射，返回相应的值
get 方法表示_返回指定键所映射的值；如果此映射不包含该键的映射关系，则返回 null。_，get 方法的参数表示__键__，返回值表示键对应的值 要想获得 Map 中所有的键，应该使用方法_Set<K> 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_Collection<V> 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.Entry_类型所组成的 Set。

11. （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：世界杯冠军以及对应的夺冠年份，请参考本章附录。

12. （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：
1） 使用一个 Map，以老师的名字作为键，以老师教授的课程名作为值，表示上述课程安排。

2） 增加了一位新老师 Allen 教 JDBC
3） Lucy 改为教 CoreJava
4） 遍历 Map，输出所有的老师及老师教授的课程
5） *利用 Map，输出所有教 JSP 的老师。

13. （泛型）使用泛型，改写第 5 题

14. （泛型）使用泛型和 Map.Entry 接口，改写第 12 题的前 4 问

15. *（List）写出下面程序的输出结果

```
import java.util.*;
class MyClass{
    int value;
    public MyClass(){}
    public MyClass(int value){ this.value = value; }
    public String toString(){
        return “”+value;
    }
}
public class TestList{
    public static void main(String args[]){
        MyClass mc1 = new MyClass(10);
        MyClass mc2 = new MyClass(20);
        MyClass mc3 = new MyClass(30);
        List list = new ArrayList();
        list.add(mc1);
        list.add(mc2);
        list.add(mc3);
        MyClass mc4 = (MyClass) list.get(1);
        mc4.value = 50;
        for(int i = 0; i<list.size(); i++){
            System.out.println(list.get(i));
        }
    }
}
```

10
50
30

更改下标为1的数据 20 为 50

16. *（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;
```

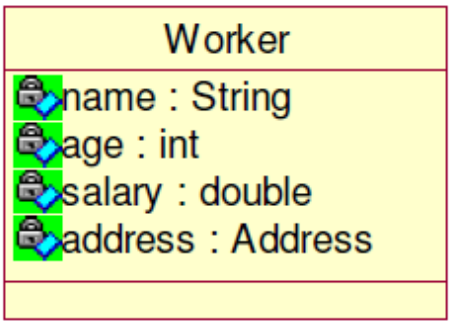
```
        this.age = age;
    }
    public int hashCode(){
        return name.hashCode() + age;
    }
    public boolean equals(Object o){
        if (o == null) return false;
        if (o == this) return true;
        if (o.getClass() != this.getClass()) return false;
        Student stu = (Student) o;
        if (stu.name.equals(name) && stu.age == age) return true;
        else return false;
    }
}
public class TestHashSet{
    public static void main(String args[]){
        Set set = new HashSet();
        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());
    }
}
```

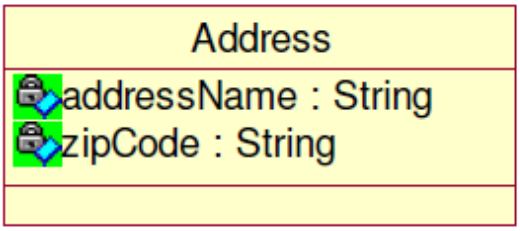
// null没有hashCode方法
// 解决方法：try catch

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

17. *（Set）有如下两个类（只写了类的属性，请自行添加相应的构造方法和 get/set 方法）

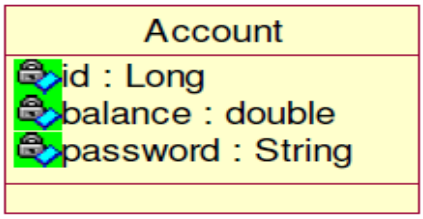




要求，完善 Worker 和 Address 类，使得 Worker 对象能够正确放入 HashSet 中：即将 Worker 放入 HashSet 中时不会出现重复元素。并编写相应测试代码。

18. *（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，读入“巴西”，应当输出
1958 1962 1970 1994 2002
读入“荷兰”，应当输出
没有获得过世界杯

19. *（Map）设计 Account 对象如下：



要求完善设计，使得该 Account 对象能够自动分配 id。
给定一个 List 如下：

```
List list = new ArrayList();
list.add(new Account(10.00, “1234”));
list.add(new Account(15.00, “5678”));
list.add(new Account(0, “1010”));
```

要求把 List 中的内容放到一个 Map 中，该 Map 的键为 id，值为相应的 Account 对象。
最后遍历这个 Map，打印所有 Account 对象的 id 和余额。

20. *（List）写一个函数 reverseList，该函数能够接受一个 List，然后把该 List 倒序排列。
例如：

```
List list = new ArrayList();
list.add(“Hello”);
list.add(“World”);
list.add(“Learn”); //此时 list 为 Hello World Learn
reverseList(list);
//调用 reverseList 方法之后，list 为 Learn World Hello
```

21. **（Map，Hash 算法）有如下代码：

```
import java.util.*;
class MyKey{
    int keyValue;
    public MyKey(){}
    public MyKey(int value){this.keyValue = value;}
}
```

```
class MyValue{
    String value;
    public MyValue(){}
    public MyValue(String value){this.value = value;}
    public String toString(){return value;}
}
public class TestMap{
    public static void main(String args[]){
        Map map = new HashMap();
        MyKey key1 = new MyKey(10);
        map.put(key1, new MyValue(“abc”));
        map.put(new MyKey(10), new MyValue(“cde”));
        System.out.println(map.get(key1));
        System.out.println(map.size());
    }
}
```

abc
2

写出该代码的输出结果。

22. **（Id，hashCode，equals）为 Worker 类增加 id 属性，用来唯一标识一个员工。即：如果员工的 id 不同，则不管其姓名、年龄、工资是否相同，都认为是不同的员工。部分代码如下：

```
class Worker{
    private final Long id;
    private String name;
    private double salary;
    private int age;
    //构造方法…
    //get/set 方法…
    public boolean equals(Object obj){
        //1 此处仅判断 id 是否相同
    }
    public int hashCode(){
        //2 此处返回 hashCode
    }
}
```

要求：

1） 完善构造方法和 get/set 方法。要求自动分配 Worker 的 id

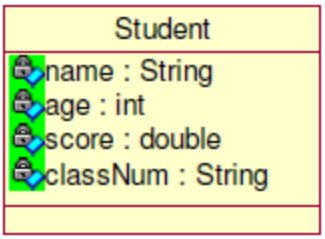
2） 完善 equals 方法。要求仅判断 id 是否相同

3） //2 处，如果写成

```
return (int)(name.hashCode() + id.hashCode() + age + salary);
```

是否正确？为什么？ 不能，final修饰的id没有hashCode方法

23. **（综合）有如下 Student 对象



其中，classNum 表示学生的班号，例如“class05”。
有如下 List

```
List list = new ArrayList();
list.add(new Student(“Tom”, 18, 100, “class05”));
list.add(new Student(“Jerry”, 22, 70, “class04”));
list.add(new Student(“Owen”, 25, 90, “class05”));
list.add(new Student(“Jim”, 30,80 , “class05”));
list.add(new Student(“Steve”, 28, 66, “class06”));
list.add(new Student(“Kevin”, 24, 100, “class04”));
```

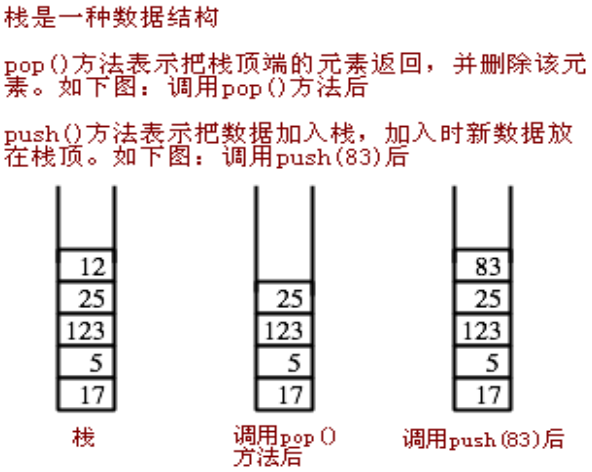
在这个 list 的基础上，完成下列要求：

1） 计算所有学生的平均年龄
2） 计算各个班级的平均分

24. **（综合）已知有十六支男子足球队参加 2008 北京奥运会。写一个程序，把这 16 支球队随机分为 4 个小组。

注：参赛球队列表见附录
注 2：使用 Math.random 来产生随机数。

25. **（综合）写一个 MyStack 类，表示“栈”这种数据结构。
栈在表示上，就如同一个单向开口的盒子，每当有新数据进入时，都是进入栈顶。其基本操作为 push 和 pop。push 表示把一个元素加入栈顶，pop 表示把栈顶元素弹出。
示意图如下：



栈的特点：先进后出。

栈的基本操作：

1） push(Object o)：表示把元素放入栈

2） Object pop()：返回栈顶元素，并把该元素从栈中删除。如果栈为空，则返回 null 值

3） Object peek()：返回栈顶元素，但不把该元素删除。如果栈为空，则返回 null 值。

4） boolean isEmpty()：判断该栈是否为空

5） int size()：返回该栈中元素的数量

要求：

1） 利用 List，实现栈。

2） 讨论：应当用 ArrayList 作为实现类还是用 LinkedList？为什么？

附录

1. 截止到 2009 年为止，历届世界杯冠军

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

2. 2008 北京奥运会男足参赛国家：

科特迪瓦，阿根廷，澳大利亚，塞尔维亚，荷兰，尼日利亚、日本，美国，中国，新西兰，巴西，比利时，韩国，喀麦隆，洪都拉斯，意大利