**每日作业卷答案**

**JavaSE第4天**

传智播客.黑马程序员

# 基础题

### 练习一：Map接口的特点

1. 请简述Map 的特点。

* Map每个元素由键与值两部分组成
* Map键不能重复,每个键对应一个值
* 键和值可以为null

### 练习二：Entry键值对对象

1. 说出Entry键值对对象遍历Map集合的原理。

Map中存放的是两种对象，一种称为key(键)，一种称为value(值)，它们在在Map中是一一对应关系，这一对对象又称做Map 中的一个Entry(项)。Entry将键值对的对应关系封装成了对象。即键值对对象，这样我们在遍历Map集合时，就可以从每一个键值对（Entry）对象中获取对应的键与对应的值。

### 练习三：Map接口中的常用方法

1. 请使用Map集合的方法完成添加元素，根据键删除，以及根据键获取值操作。

**public class MapTest01**{  
**public static void** main(String[] args) {  
// 1.创建HashMap  
HashMap<String, String> hm = **new** HashMap<String, String>();  
  
// 2.使用put添加元素  
hm.put(**"黄晓明"**, **"Baby"**);  
 hm.put(**"邓超"**, **"孙俪"**);  
 hm.put(**"李晨"**, **"范冰冰"**);  
 hm.put(**"大黑牛"**, **"范冰冰"**);  
  
// 3.使用put修改元素  
String v1 = hm.put(**"李晨"**, **"白百合"**);  
  
// 4.使用get获取元素  
String string = hm.get(**"大黑牛"**);  
  
// 5.使用remove删除元素  
String v2 = hm.remove(**"大黑牛"**);  
 System.**out**.println(v2);  
  
// 6.打印集合中的元素  
System.**out**.println(hm);  
 }  
}

### 练习四：Map接口中的方法

1. 往一个Map集合中添加若干元素。获取Map中的所有value，并使用增强for和迭代器遍历输出每个value。

**public class** MapTest02 {  
**public static void** main(String[] args) {  
// 1.创建HashMap  
HashMap<String, String> hm = **new** HashMap<String, String>();  
  
// 2.使用put添加元素  
hm.put(**"黄晓明"**, **"Baby"**);  
 hm.put(**"邓超"**, **"孙俪"**);  
 hm.put(**"李晨"**, **"范冰冰"**);  
 hm.put(**"大黑牛"**, **"范冰冰"**);  
  
// 3.使用Map的values方法获取到所有的value  
Collection<String> values = hm.values();  
  
// 4.使用增强for获取每个value  
**for** (String value : values) {  
 System.**out**.println(value);  
 }  
  
 System.**out**.println(**"----------------"**);  
// 5.使用迭代器获取每个value  
Iterator<String> itr = values.iterator();  
**while** (itr.hasNext()) {  
 System.**out**.println(itr.next());  
 }  
 }  
}

### 练习五：HashMap存储键是自定义对象值是String

1. 请使用Map集合存储自定义数据类型Car做键，对应的价格做值。并使用keySet和entrySet两种方式遍历Map集合。

汽车类:

// 1.定义汽车类.包含名称和价格属性,重写hashCode和equals方法  
**public class** Car {  
  
**private** String **name**;  
  
**private** String **color**;  
  
**public** Car() {  
 }  
  
**public** Car(String name, String color) {  
**this**.**name** = name;  
**this**.**color** = color;  
 }  
  
**public** String getName() {  
**return name**;  
 }  
  
**public void** setName(String name) {  
**this**.**name** = name;  
 }  
  
**public** String getColor() {  
**return color**;  
 }  
  
**public void** setColor(String color) {  
**this**.**color** = color;  
 }  
  
@Override  
**public boolean** equals(Object o) {  
**if** (**this** == o) **return true**;  
**if** (!(o **instanceof** Car)) **return false**;  
  
 Car car = (Car) o;  
  
**if** (**name** != **null** ? !**name**.equals(car.**name**) : car.**name** != **null**) **return false**;  
**return color** != **null** ? **color**.equals(car.**color**) : car.**color** == **null**;  
 }  
  
@Override  
**public int** hashCode() {  
**int** result = **name** != **null** ? **name**.hashCode() : 0;  
 result = 31 \* result + (**color** != **null** ? **color**.hashCode() : 0);  
**return** result;  
 }  
}

测试类:

**public class** MapTest03 {  
**public static void** main(String[] args) {  
// 2.创建HashMapkey保存汽车对象,value是汽车价格  
HashMap<Car, Integer> hm = **new** HashMap<>();  
  
// 3.添加汽车到HashMap中  
Car c1 = **new** Car(**"长安奔奔"**, **"黄色"**);  
 Car c3 = **new** Car(**"奇瑞QQ"**, **"黑色"**);  
 Car c2 = **new** Car(**"铃木奥拓"**, **"白色"**);  
  
 hm.put(c1, 10000);  
 hm.put(c2, 20000);  
 hm.put(c3, 30000);  
  
// 4.使用keySet方式遍历Map  
Set<Car> keySet = hm.keySet();  
**for** (Car c : keySet) {  
// 根据key获取value  
Integer value = hm.get(c);  
 System.**out**.println(c.getName() + **","**+ c.getPrice() + **" - "**+ value);  
 }  
  
 System.**out**.println(**"-------------"**);  
  
// 5.使用entrySet方式遍历Map  
Set<Map.Entry<Car, Integer>> entrySet = hm.entrySet();  
**for** (Map.Entry<Car, Integer> entry : entrySet) {  
 Car key = entry.getKey();  
 Integer value = entry.getValue();  
 System.**out**.println(key.getName() + **","**+ key.getPrice() + **" - "**+ value);  
 }  
 }  
}

### 练习六：Map集合的使用（一）

1. 现在有一个map集合如下：

Map<Integer,String> map = new HashMap<Integer, String>();  
        map.put(1, "张三丰");  
        map.put(2, "周芷若");  
        map.put(3, "汪峰");  
        map.put(4, "灭绝师太");

要求：

1.遍历集合，并将序号与对应人名打印。

2.向该map集合中插入一个编码为5姓名为李晓红的信息

3.移除该map中的编号为1的信息

4.将map集合中编号为2的姓名信息修改为"周林"

**public class** MapTest04 {  
**public static void** main(String[] args) {  
// 1.定义HashMap,编号作为key,姓名作为value  
Map<Integer,String> map = **new** HashMap<Integer, String>();  
  
// 2.使用put方法添加元素  
map.put(1, **"张三丰"**);  
 map.put(2, **"周芷若"**);  
 map.put(3, **"汪峰"**);  
 map.put(4, **"灭绝师太"**);  
  
// 3.使用keySet+增强for迭代map中的元素,并打印  
Set<Integer> keySet = map.keySet();  
**for** (Integer key : keySet) {  
 String value = map.get(key);  
 System.**out**.println(key + **" -- "**+ value);  
 }  
  
// 4.使用put向该map集合中插入一个编码为5姓名为李晓红的信息  
map.put(5, **"李晓红"**);  
  
// 5.使用remove移除该map中的编号为1的信息  
map.remove(1);  
  
// 6.使用put将map集合中编号为2的姓名信息修改为"周林"  
map.put(2, **"周林"**);  
 System.**out**.println(map);  
 }  
}

### 练习七：Map集合的使用（二）

1. 有2个数组，第一个数组内容为：[黑龙江省,浙江省,江西省,广东省,福建省]，第二个数组为：[哈尔滨,杭州,南昌,广州,福州]，将第一个数组元素作为key，第二个数组元素作为value存储到Map集合中。如{黑龙江省=哈尔滨, 浙江省=杭州, …}。

**public class** MapTest05 {  
**public static void** main(String[] args) {  
// 1.定义第一个数组arr1  
String[] arr1 = {**"黑龙江省"**, **"浙江省"**, **"江西省"**, **"广东省"**, **"福建省"**};  
// 2.定义第二个数组arr2  
String[] arr2 = {**"哈尔滨"**, **"杭州"**, **"南昌"**, **"广州"**, **"福州"**};  
  
// 3.创建HashMap,key存放省,value存放市  
HashMap<String, String> hm = **new** HashMap<>();  
  
// 4.使用普通for循环遍历arr1  
**for** (**int** i = 0; i < arr1.**length**; i++) {  
// 5.根据索引到arr1中获取到省  
String key = arr1[i];  
// 6.根据索引到arr2中获取到省会城市  
String value = arr2[i];  
  
// 7.将省和省会城市添加到HashMap中  
hm.put(key, value);  
 }  
// 8.输出HashMap中的内容  
System.**out**.println(hm);  
 }  
}

## 扩展题

### 练习八：Map集合的使用（三）

1. 定义一个泛型为String类型的List集合，统计该集合中每个字符（注意，不是字符串）出现的次数。例如：集合中有”abc”、”bcd”两个元素，程序最终输出结果为：“a = 1,b = 2,c = 2,d = 1”。

**public class** MapTest06 {  
**public static void** main(String[] args) {  
// 1.定义ArrayList存放元素  
ArrayList<String> arr = **new** ArrayList<String>();  
  
// 2.使用add方法添加需要的元素  
arr.add(**"abc"**);  
 arr.add(**"bcd"**);  
  
// 3.定义HashMap,key是字符,value是字符对应的次数  
HashMap<Character, Integer> hm = **new** HashMap<>();  
  
// 4.使用增强for获取ArrayList中的每个字符串  
**for** (String str : arr) {  
  
// 5.将每个字符串转成字符数组  
**char**[] charArray = str.toCharArray();  
  
// 6.使用增强for遍历字符数组  
**for** (**char** ch : charArray) {  
// 7.获取拿到每个字符,使用字符去HashMap中查找次数  
Integer num = hm.get(ch);  
// 8.如果为空,说明该字符第一次出现  
**if** (num == **null**) {  
// 9.放入字符作为key,把次数设置为1  
hm.put(ch, 1);  
 } **else** {  
// 10.如果之前,已经有字符了,就把字符的次数+1  
hm.put(ch, num + 1);  
 }  
 }  
 }  
// 11.输出map中的内容  
System.**out**.println(hm);  
 }  
}

### 练习九：Map集合的使用（四）

1. 利用Map，完成下面的功能：

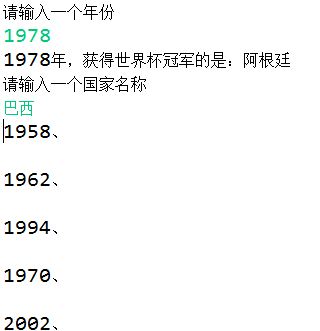
从命令行读入一个字符串，表示一个年份，输出该年的世界杯冠军是哪支球队。如果该 年没有举办世界杯，则输出：没有举办世界杯。

//tips:参阅Map接口containsKey(Object key)方法

1. 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

//tips:参阅Map接口containsValue(Object value)方法

示例：



附：历届世界杯冠军

|  |  |  |  |
| --- | --- | --- | --- |
| 届数 | 举办年份 | 举办地点 | 冠军 |
| [第一届](http://baike.baidu.com/view/11095686.htm) | 1930年 | 乌拉圭 | 乌拉圭 |
| 第二届 | 1934年 | 意大利 | 意大利 |
| 第三届 | 1938年 | 法国 | 意大利 |
| 第四届 | 1950年 | 巴西 | 乌拉圭 |
| 第五届 | 1954年 | 瑞士 | 西德 |
| 第六届 | 1958年 | 瑞典 | 巴西 |
| 第七届 | 1962年 | 智利 | 巴西 |
| 第八届 | 1966年 | 英格兰 | 英格兰 |
| 第九届 | 1970年 | 墨西哥 | 巴西 |
| 第十届 | 1974年 | 前西德 | 西德 |
| 第十一届 | 1978年 | 阿根廷 | 阿根廷 |
| 第十二届 | 1982年 | 西班牙 | 意大利 |
| 第十三届 | 1986年 | 墨西哥 | 阿根廷 |
| 第十四届 | 1990年 | 意大利 | 西德 |
| 第十五届 | 1994年 | 美国 | 巴西 |
| 第十六届 | 1998年 | 法国 | 法国 |
| 第十七届 | 2002年 | 韩日 | 巴西 |
| 第十八届 | 2006年 | 德国 | 意大利 |
| 第十九届 | 2010年 | 南非 | 西班牙 |
| 第二十届 | 2014年 | 巴西 | 德国 |

**import java.util.HashMap;  
import java.util.Map;  
import java.util.Scanner;  
  
public class MapTest09 {  
  
 public static void main(String[] args) {  
  
 Map<Integer, String> m = new HashMap<>();  
  
 m.put(1930, "乌拉圭");  
 m.put(1934, "意大利");  
 m.put(1938, "意大利");  
 m.put(1950, "乌拉圭");  
 m.put(1954, "西德");  
 m.put(1958, "巴西");  
 m.put(1962, "巴西");  
 m.put(1966, "英格兰");  
 m.put(1970, "巴西");  
 m.put(1974, "西德");  
 m.put(1978, "阿根廷");  
 m.put(1982, "意大利");  
 m.put(1986, "阿根廷");  
 m.put(1990, "西德");  
 m.put(1994, "巴西");  
 m.put(1998, "法国");  
 m.put(2002, "巴西");  
 m.put(2006, "意大利");  
 m.put(2010, "西班牙");  
 m.put(2014, "德国");  
  
 System.out.println("请输入一个年份");  
 Scanner s = new Scanner(System.in);  
 int key = s.nextInt();  
  
 if (m.containsKey(key)) {  
 System.out.println(key + "年，获得世界杯冠军的是：" + m.get(key));  
 } else {  
 System.out.println("该年没有举办世界杯！");  
 }  
  
 System.out.println("请输入一个国家名称");  
 Scanner g = new Scanner(System.in);  
 String val = g.nextLine();  
  
 if (m.containsValue(val)) {  
 for (Integer year : m.keySet()) {  
 if (m.get(year).equals(val)) {  
 System.out.println(year + "、\n");  
 }  
 }  
 } else {  
 System.out.println("该国家没有获得世界杯冠军");  
 }  
  
 }  
  
}**