# 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);  
 }  
}