**每日作业卷**

**JavaSE第09天**

# 关卡1

## 训练案例1

### 训练知识点

1.Map特点

### 训练描述

请简述Map 的特点

### 操作步骤描述

Map的特点:

1.Map每个元素由键与值两部分组成

2.Map键不能重复,每个键对应一个值

3.键和值可以为null

## 训练案例2

### 训练知识点

1.Map接口中的常用方法

### 训练描述

练习Map集合的常用方法: put, get, remove

### 操作步骤描述

1. 创建HashMap
2. 使用put添加元素
3. 使用put修改元素
4. 使用get获取元素
5. 使用remove删除元素
6. 打印集合中的元素

**package** com.day09.exam01;

**import** java.util.HashMap;

**public** **class** Test02 {

**public** **static** **void** main(String[] args) {

HashMap<String, String> hashMap = **new** HashMap<>();

hashMap.put("姓名", "张三");

System.***out***.println(hashMap);

hashMap.put("姓名", "李四");

System.***out***.println(hashMap);

String name = hashMap.get("姓名");

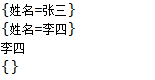
System.***out***.println(name);

hashMap.remove("姓名");

System.***out***.println(hashMap);

}

}



## 训练案例3

### 训练知识点

1.Map接口中的方法

### 训练描述

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

### 操作步骤描述

1. 创建HashMap
2. 使用put添加元素
3. 使用Map的keySet方法获取到所有的key
4. 使用增强for获取每个key
5. 获取keySet的迭代器
6. 循环判断迭代器中是否有下一个元素
7. 使用迭代器获取每个key

**package** com.day09.exam01;

**import** java.util.HashMap;

**import** java.util.Iterator;

**import** java.util.Set;

**public** **class** Test03 {

**public** **static** **void** main(String[] args) {

HashMap<String, String> map = **new** HashMap<String, String>();

map.put("及时雨", "宋江");

map.put("玉麒麟", "卢俊义");

map.put("智多星", "高俅");

map.put("智多星", "吴用");

Set<String> keys = map.keySet();

**for** (String s : keys) {

System.***out***.print(s + " ");

}

System.***out***.println("");

Iterator<String> iterator = keys.iterator();

**while** (iterator.hasNext()) {

String s = (String) iterator.next();

System.***out***.print(s + " ");

}

}

}



## 训练案例4

### 训练知识点

1.Map接口中的方法

### 训练描述

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

### 操作步骤描述

1. 创建HashMap
2. 使用put添加元素
3. 使用Map的values方法获取到所有的value
4. 使用增强for获取每个value
5. 使用迭代器获取每个value

**package** com.day09.exam01;

**import** java.util.Collection;

**import** java.util.HashMap;

**import** java.util.Iterator;

**public** **class** Test04 {

**public** **static** **void** main(String[] args) {

HashMap<String, String> map = **new** HashMap<String, String>();

map.put("及时雨", "宋江");

map.put("玉麒麟", "卢俊义");

map.put("智多星", "高俅");

map.put("智多星", "吴用");

Collection<String> values = map.values();

**for** (String s : values) {

System.***out***.print(s + " ");

}

System.***out***.println("");

Iterator<String> iterator = values.iterator();

**while** (iterator.hasNext()) {

String s = (String) iterator.next();

System.***out***.print(s + " ");

}

}

}



## 训练案例5

### 训练知识点

1.Map遍历

### 训练描述

Map集合中包含5对元素: "邓超"->"孙俪", "李晨"->"范冰冰", "刘德华"->"柳岩", “黄晓明”->” Baby”, “谢霆锋”->”张柏芝”.使用keySet方式迭代出Map集合中的元素

### 操作步骤描述

1. 创建HashMap
2. 使用put方法添加元素
3. 使用keySet方法获取所有的键
4. 获取到keySet的迭代器
5. 循环判断迭代器是否有下一个元素
6. 使用迭代器next方法获取到一个键
7. 通过一个键找到一个值
8. 输出键和值

**package** com.day09.exam01;

**import** java.util.HashMap;

**import** java.util.Iterator;

**import** java.util.Set;

**public** **class** Test05 {

**public** **static** **void** main(String[] args) {

HashMap<String, String> hashMap = **new** HashMap<>();

hashMap.put("邓超", "孙俪");

hashMap.put("李晨", "范冰冰");

hashMap.put("刘德华", "柳岩");

hashMap.put("黄晓明", "Baby");

hashMap.put("谢霆锋", "张柏芝");

Set<String> keys = hashMap.keySet();

Iterator<String> iterator = keys.iterator();

**while** (iterator.hasNext()) {

String s = (String) iterator.next();

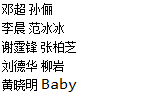
String name = hashMap.get(s);

System.***out***.println(s + " " + name);

}

}

}



## 训练案例6

### 训练知识点

1.Map遍历

### 训练描述

Map集合中包含5对元素: "邓超"->"孙俪", "李晨"->"范冰冰", "刘德华"->"柳岩", “黄晓明”->” Baby”, “谢霆锋”->”张柏芝”.使用Entry方式迭代出Map集合中的元素

### 操作步骤描述

1. 创建HashMap
2. 使用put方法添加元素
3. 使用entrySet方法获取到所有的键值对关系
4. 获取entrySet的迭代器
5. 使用迭代器查看是否有下一个元素
6. 使用迭代器拿到下一个元素(Entry类型)
7. 通过一个Entry<K,V>获取到key和value

**package** com.day09.exam01;

**import** java.util.HashMap;

**import** java.util.Iterator;

**import** java.util.Map;

**import** java.util.Map.Entry;

**import** java.util.Set;

**public** **class** Test06 {

**public** **static** **void** main(String[] args) {

HashMap<String, String> hashMap = **new** HashMap<>();

hashMap.put("邓超", "孙俪");

hashMap.put("李晨", "范冰冰");

hashMap.put("刘德华", "柳岩");

hashMap.put("黄晓明", "Baby");

hashMap.put("谢霆锋", "张柏芝");

Set<Entry<String, String>> entrySet = hashMap.entrySet();

Iterator<Entry<String, String>> iterator = entrySet.iterator();

**while** (iterator.hasNext()) {

Map.Entry<String, String> entry = (Map.Entry<String, String>) iterator.next();

String name1 = entry.getKey();

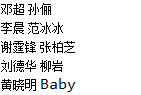
String name2 = entry.getValue();

System.***out***.println(name1 + " " + name2);

}

}

}



## 训练案例7

### 训练知识点

1.HashMap存储键是String值是自定义对象

### 训练描述

每位学生（姓名，年龄）都有自己的家庭住址。那么，既然有对应关系，则将学生对象和家庭住址存储到Map集合中。**家庭住址**作为**键**, **学生**作为**值**。并使用keySet和entrySet方式遍历Map集合

### 操作步骤描述

1. 定义学生类.包含姓名和年龄属性
2. 创建HashMapkey保存学生家庭住址,value是学生对象
3. 添加一批学生到HashMap中
4. 使用keySet方式遍历Map
5. 使用entrySet方式遍历Map

**package** com.day09.exam01;

**import** java.util.HashMap;

**import** java.util.Map.Entry;

**import** java.util.Set;

**public** **class** Test07 {

**public** **static** **void** main(String[] args) {

HashMap<String, Student> hashMap = **new** HashMap<>();

hashMap.put("江苏", **new** Student("张三", 18));

hashMap.put("山东", **new** Student("李四", 19));

Set<String> keys = hashMap.keySet();

**for** (String s : keys) {

Student student = hashMap.get(s);

System.***out***.println(s + " " + student.toString());

}

System.***out***.println("=======================");

Set<Entry<String, Student>> entries = hashMap.entrySet();

**for** (Entry<String, Student> entry : entries) {

String address = entry.getKey();

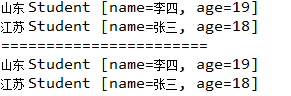
Student student = entry.getValue();

System.***out***.println(address + " " + student.toString());

}

}

}



## 训练案例8

### 训练知识点

1.HashMap存储键是自定义对象值是String

### 训练描述

每位学生（姓名，年龄）都有自己的家庭住址。那么，既然有对应关系，则将学生对象和家庭住址存储到map集合中。**学生**作为**键**, **家庭住址**作为**值,** 并使用keySet和entrySet方式遍历Map集合.注意，学生**姓名**相同并且**年龄**相同视为同一名学生

### 操作步骤描述

1. 定义学生类.包含姓名和年龄属性, 重写hashCode和equals方法
2. 创建HashMapkey保存学生对象,value是学生家庭住址
3. 添加一批学生到HashMap中
4. 使用keySet方式遍历Map
5. 使用entrySet方式遍历Map

**package** com.day09.exam01;

**import** java.util.HashMap;

**import** java.util.Map.Entry;

**import** java.util.Set;

**public** **class** Test08 {

**public** **static** **void** main(String[] args) {

HashMap<Student, String> hashMap = **new** HashMap<>();

hashMap.put(**new** Student("张三", 18), "江苏");

hashMap.put(**new** Student("李四", 18), "山东");

hashMap.put(**new** Student("张三", 18), "海南");

Set<Student> keys = hashMap.keySet();

**for** (Student student : keys) {

String address = hashMap.get(student);

System.***out***.println(student.toString() + " " + address);

}

System.***out***.println("===================");

Set<Entry<Student, String>> entries = hashMap.entrySet();

**for** (Entry<Student, String> entry : entries) {

Student student = entry.getKey();

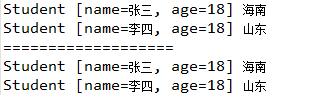
String address = entry.getValue();

System.***out***.println(student.toString() + " " + address);

}

}

}



## 训练案例9

### 训练知识点

1.可变参数

### 训练描述

定义一个方法add接收0个及以上的int类型数据,这个方法的作用是将传入的int类型的值进行相加,并返回相加后的结果

### 操作步骤描述

1. 定义add方法.参数是可变参数(int... arr)
2. 在add方法定义求和变量sum
3. 使用增强for获取每个参数
4. 将每个参数都加到求和变量sum中
5. 返回sum
6. 调用add方法传入任意个int类型的参数

**package** com.day09.exam01;

**public** **class** Test09 {

**public** **static** **void** main(String[] args) {

System.***out***.println(*add*(1, 2, 3, 4));

}

**public** **static** **int** add(**int**... arr) {

**int** sum = 0;

**for** (**int** i : arr) {

sum += i;

}

**return** sum;

}

}



## 训练案例10

### 训练知识点

1.Collections工具类

### 训练描述

ArrayList集合中有如下内容: {33,11,77,55},使用Collections.sort()对ArrayList集合中的数据进行排序,并打印出排序后的结果

### 操作步骤描述

1. 创建ArrayList
2. 使用add方法添加{33,11,77,55}四个元素
3. 调用Collections的sort方法,对集合排序
4. 使用增强for遍历ArrayList集合

**package** com.day09.exam01;

**import** java.util.ArrayList;

**import** java.util.Collections;

**public** **class** Test10 {

**public** **static** **void** main(String[] args) {

ArrayList<Integer> arrayList = **new** ArrayList<>();

arrayList.add(33);

arrayList.add(11);

arrayList.add(77);

arrayList.add(55);

Collections.*sort*(arrayList);

**for** (**int** i : arrayList) {

System.***out***.println(i);

}

}

}



## 训练案例11

### 训练知识点

1.Collections工具类

### 训练描述

ArrayList集合中有如下内容: {11,33,55,77},使用Collections.shuffle ()对ArrayList集合中的数据进行随机打乱顺序,并打印出排序后的结果

### 操作步骤描述

1. 创建ArrayList
2. 使用add方法添加{11,33,55,77}四个元素
3. 调用Collections的shuffle方法,对集合中的元素打乱顺序
4. 使用增强for遍历ArrayList集合

**package** com.day09.exam01;

**import** java.util.ArrayList;

**import** java.util.Collections;

**public** **class** Test11 {

**public** **static** **void** main(String[] args) {

ArrayList<Integer> arrayList = **new** ArrayList<>();

arrayList.add(11);

arrayList.add(33);

arrayList.add(55);

arrayList.add(77);

Collections.*shuffle*(arrayList);

**for** (**int** i : arrayList) {

System.***out***.println(i);

}

}

}



## 训练案例12

### 训练知识点

1.Collections工具类

### 训练描述

ArrayList集合中有如下内容: {11,33,55,77},使用Collections.reverse()对ArrayList集合中的数据进行反转,使用增强for遍历ArrayList集合

### 操作步骤描述

1. 创建ArrayList
2. 使用add方法添加{11,33,55,77}四个元素
3. 调用Collections的reverse方法,对ArrayList集合中的数据进行反转
4. 使用增强for遍历ArrayList集合

**package** com.day09.exam01;

**import** java.util.ArrayList;

**import** java.util.Collections;

**public** **class** Test12 {

**public** **static** **void** main(String[] args) {

ArrayList<Integer> arrayList = **new** ArrayList<>();

arrayList.add(11);

arrayList.add(33);

arrayList.add(55);

arrayList.add(77);

Collections.*reverse*(arrayList);

**for** (**int** i : arrayList) {

System.***out***.println(i);

}

}

}



## 训练案例13

### 训练知识点

1.Collections工具类

### 训练描述

ArrayList集合中有如下内容: {11,33,55,77},使用Collections.binarySearch ()查询33元素在ArrayList集合中的索引是多少

### 操作步骤描述

1. 创建ArrayList
2. 使用add方法添加{11,33,55,77}四个元素
3. 调用Collections的binarySearch方法,对ArrayList集合中的数据进行反转
4. 输出查找到的索引

**package** com.day09.exam01;

**import** java.util.ArrayList;

**import** java.util.Collections;

**public** **class** Test13 {

**public** **static** **void** main(String[] args) {

ArrayList<Integer> arrayList = **new** ArrayList<>();

arrayList.add(11);

arrayList.add(33);

arrayList.add(55);

arrayList.add(77);

System.***out***.println(Collections.*binarySearch*(arrayList, 33));

}

}



# 关卡2

## 训练案例1

### 训练知识点

1.Map集合

### 训练描述

使用键盘录入5个学生信息,录入格式为:”姓名,年龄”.将这5个学生信息存入Map集合中,key是姓名,value是年龄

### 操作步骤描述

1. 创建HashMap
2. 创建键盘录入对象
3. 循环判断只要HashMap数量是否小于5
4. HashMap数量小于5个就让用户接着输入
5. 使用,分割用户输入的字符串
6. 将分割后的索引为0的字符串作为key,索引为1的字符串作为value,放入HashMap中
7. 输出HashMap中的数据

**package** com.day09.exam02;

**import** java.util.HashMap;

**import** java.util.Scanner;

**public** **class** Test01 {

**public** **static** **void** main(String[] args) {

HashMap<String, String> hashMap = **new** HashMap<>();

Scanner scanner = **new** Scanner(System.***in***);

**while** (hashMap.size() < 5) {

String str = scanner.nextLine();

String arr[] = str.split(",");

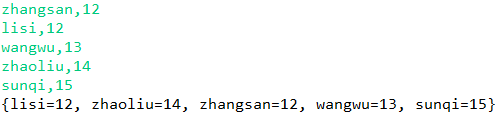
hashMap.put(arr[0], arr[1]);

}

System.***out***.println(hashMap);

}

}



## 训练案例2

### 训练知识点

1.可变参数

### 训练描述

定义一个方法add接收0个及以上的String类型数据,这个方法的作用是将传入的String类型的参数放入一个ArrayList中,并返回这个ArrayList集合

### 操作步骤描述

1. 定义add方法,参数是(String ... str),返回值是ArrayList<String>
2. 在add方法中创建ArrayList用于存放传入的字符串
3. 使用增强for循环获取每个参数
4. 将每个参数都添加到ArrayList中
5. 返回array
6. 调用add方法传入任意个字符串
7. 使用增强for遍历返回的ArrayList,打印每个元素

**package** com.day09.exam02;

**import** java.util.ArrayList;

**public** **class** Test02 {

**public** **static** **void** main(String[] args) {

ArrayList<String> strArr = **new** ArrayList<>();

strArr = *add*("aa", "bb", "cc");

**for** (String s : strArr) {

System.***out***.println(s);

}

}

**public** **static** ArrayList<String> add(String... str) {

ArrayList<String> arrayList = **new** ArrayList<>();

**for** (String s : str) {

arrayList.add(s);

}

**return** arrayList;

}

}



## 训练案例3

### 训练知识点

1.集合转数组

### 训练描述

ArrayList集合中有如下内容: {11,33,55,77}.使用2种方式将集合中的数据转成数组

### 操作步骤描述

1. 定义ArrayList
2. 使用add方法添加元素{11,33,55,77}
3. 使用方式1转数组.toArray()
4. 使用方式2转数组.准备空数组
5. 打印数组

**package** com.day09.exam02;

**import** java.util.ArrayList;

**public** **class** Test03 {

**public** **static** **void** main(String[] args) {

ArrayList<Integer> arrayList = **new** ArrayList<>();

arrayList.add(11);

arrayList.add(33);

arrayList.add(55);

arrayList.add(77);

Object[] arr1 = arrayList.toArray();

**for** (**int** i = 0; i < arr1.length; i++) {

System.***out***.println(arr1[i]);

}

System.***out***.println("------------------");

**int**[] arr2 = **new** **int**[arrayList.size()];

**for** (**int** i = 0; i < arr2.length; i++) {

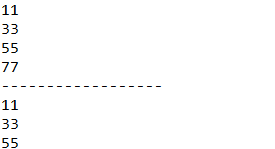
arr2[i] = arrayList.get(i);

System.***out***.println(arr2[i]);

}

}

}



## 训练案例4

### 训练知识点

1.Map集合

### 训练描述

研发部门有5个人，信息如下：（姓名-工资）【柳岩=2100, 张亮=1700, 诸葛亮=1800, 灭绝师太=2600, 东方不败=3800】,将以上员工的相关信息存放在适合的集合中,给柳岩涨工资300,迭代出每个元素的内容输出到控制台

### 操作步骤描述

1. 定义HashMap,姓名作为key,工资作为value
2. 使用put方法添加需要的元素
3. 获取到柳岩的工资
4. 修改柳岩的工资为当前工资加上300
5. 使用增强for+keySet迭代出每个员工的工资

**package** com.day09.exam02;

**import** java.util.HashMap;

**public** **class** Test04 {

**public** **static** **void** main(String[] args) {

HashMap<String, Integer> hashMap = **new** HashMap<>();

hashMap.put("柳岩", 2100);

hashMap.put("张亮", 1700);

hashMap.put("诸葛亮", 1800);

hashMap.put("灭绝师太", 2600);

hashMap.put("东方不败", 3800);

**int** money = hashMap.get("柳岩");

System.***out***.println(money);

hashMap.put("柳岩", money + 300);

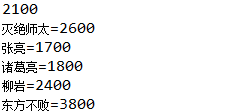
**for** (String s : hashMap.keySet()) {

System.***out***.println(s + "=" + hashMap.get(s));

}

}

}



## 训练案例5

### 训练知识点

1.Map集合

### 训练描述

现在有一个map集合如下:{1="张三丰",2="周芷若",3="汪峰",4="灭绝师太"}

要求:

1.将map集合中的所有信息显示到控制台上

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"

### 操作步骤描述

1. 定义HashMap,编号作为key,姓名作为value
2. 使用put方法添加元素
3. 使用keySet+增强for迭代map中的元素,并打印
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
5. 使用remove移除该map中的编号为1的信息
6. 使用put将map集合中编号为2的姓名信息修改为"周林"

**package** com.day09.exam02;

**import** java.util.HashMap;

**public** **class** Test05 {

**public** **static** **void** main(String[] args) {

HashMap<Integer, String> hashMap = **new** HashMap<>();

hashMap.put(1, "张三丰");

hashMap.put(2, "周芷若");

hashMap.put(3,"汪峰");

hashMap.put(4, "灭绝师太");

System.***out***.println(hashMap);

hashMap.put(5, "李晓红");

hashMap.remove(1);

hashMap.put(2, "周林");

System.***out***.println(hashMap);

}

}



## 训练案例6

### 训练知识点

1.Map集合

### 训练描述

利用键盘录入，输入一个字符串,统计该字符串中各个字符的数量

如用户输入字符串:"hello world java",程序输出结果：h(1)e(1)l(3)o(2) (2)w(1)r(1)d(1)j(1)a(2)v(1)

注：输出结果不要求顺序一致

### 操作步骤描述

1. 创建键盘输入对象
2. 使用nextLine从键盘中读取一行内容
3. 将字符串转换成字符数组
4. 创建map集合，用来存放字符：key是字符 value是字符出现的个数
5. 遍历字符数组
6. 拿到每个字符,去map中查看是否有保存对应的次数
7. 如果这个字符没有保存次数,就设置为1次
8. 如果这个字符有保存次数,就设置为原来的次数+1次
9. 创建字符串缓冲区对象
10. 按题目要求拼接字符串
11. 输出结果

**package** com.day09.exam02;

**import** java.util.HashMap;

**import** java.util.Scanner;

**public** **class** Test06 {

**public** **static** **void** main(String[] args) {

HashMap<Character, Integer> hashMap = **new** HashMap<>();

Scanner scanner = **new** Scanner(System.***in***);

**char** str[] = scanner.nextLine().toCharArray();

**for** (**int** i = 0; i < str.length; i++) {

**if** (!hashMap.containsKey(str[i])) {

hashMap.put(str[i], 1);

} **else** {

hashMap.put(str[i], hashMap.get(str[i]) + 1);

}

}

StringBuffer sb = **new** StringBuffer();

**for** (**char** i : hashMap.keySet()) {

sb.append(i).append("(").append(hashMap.get(i)).append(")");

}

System.***out***.println(sb.toString());

}

}



## 训练案例7

### 训练知识点

1.Map集合

### 训练描述

键盘录入一个字符串,分别统计出其中英文字母、空格、数字和其它字符的数量,输出结果:”其他=1, 空格=2, 字母=12, 数字=6”

### 操作步骤描述

1. 定义countingKey(HashMap<String, Integer> map, String key)方法.用于统计key出现的次数.调用一次,让key所对应的次数+1.
2. 在countingKey方法中.使用get方法从map中取出key的值
3. 如果为空,说明key第一次出现
4. 放入key,把次数设置为1
5. 如果之前,已经有key了,就把key的次数+1
6. 创建键盘录入对象
7. 使用nextLine获取用户录入的字符串
8. 定义一个Map用于统计,字母,空格,数字和其他字符的数量,key是字符value是字符出现的次数
9. 将输入的字符串转成字符
10. 遍历字符串数组中的每一个字符
11. 如果ch是一个字符,调用countingKey方法传入(map, "字母")作为参数
12. 如果字符是一个空格,调用countingKey方法传入(map, "空格")作为参数
13. 如果字符是一个数字,调用countingKey方法传入(map, "数字")作为参数
14. 如果字符是一个其他,调用countingKey方法传入(map, "其他")作为参数
15. 打印统计结果

**package** com.day09.exam02;

**import** java.util.HashMap;

**import** java.util.Scanner;

**public** **class** Test07 {

**public** **static** **void** main(String[] args) {

HashMap<String, Integer> hashMap = **new** HashMap<>();

Scanner scanner = **new** Scanner(System.***in***);

**char**[] chs = scanner.nextLine().toCharArray();

**for** (**int** i = 0; i < chs.length; i++) {

**if** ((chs[i] >= 'a' && chs[i] <= 'z') || (chs[i] >= 'A' && chs[i] <= 'Z')) {

*countingKey*(hashMap, "字母");

} **else** **if** (chs[i] >= '0' && chs[i] <= '9') {

*countingKey*(hashMap, "数字");

} **else** **if** (chs[i] == ' ') {

*countingKey*(hashMap, "空格");

} **else** {

*countingKey*(hashMap, "其他");

}

}

System.***out***.println(hashMap);

}

**public** **static** **void** countingKey(HashMap<String, Integer> map, String key) {

**if** (!map.containsKey(key)) {

map.put(key, 1);

} **else** {

map.put(key, map.get(key) + 1);

}

}

}



## 训练案例8

### 训练知识点

1.Map集合

### 训练描述

定义一个List集合,泛型为String类型。统计整个集合中所有字符（注意，不是字符串）的个数。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”

### 操作步骤描述

1. 定义ArrayList存放元素
2. 使用add方法添加需要的元素
3. 定义HashMap,key是字符,value是字符对应的次数
4. 使用增强for获取ArrayList中的每个字符串
5. 将每个字符串转成字符数组
6. 使用增强for遍历字符数组
7. 获取拿到每个字符,使用字符去HashMap中查找次数
8. 如果为空,说明该字符第一次出现
9. 放入字符作为key,把次数设置为1
10. 如果之前,已经有字符了,就把字符的次数+1
11. 输出map中的内容

**package** com.day09.exam02;

**import** java.util.ArrayList;

**import** java.util.HashMap;

**public** **class** Test08 {

**public** **static** **void** main(String[] args) {

HashMap<Character, Integer> hashMap = **new** HashMap<>();

ArrayList<String> strArr = **new** ArrayList<>();

strArr.add("abc");

strArr.add("bcd");

**for** (String s : strArr) {

**char**[] chs = s.toCharArray();

**for** (**int** i = 0; i < chs.length; i++) {

**if** (!hashMap.containsKey(chs[i])) {

hashMap.put(chs[i], 1);

} **else** {

hashMap.put(chs[i], hashMap.get(chs[i]) + 1);

}

}

}

System.***out***.println(hashMap);

}

}



## 训练案例9

### 训练知识点

1.Map集合

### 训练描述

键盘录入语句英语,统计每个单词出现的次数,有如下字符串"If you want to change your fate I think you must come to the dark horse to learn java"(用空格间隔)

打印格式：

to=3

think=1

you=2

…

### 操作步骤描述

1. 创建键盘输入对象
2. 使用nextLine从键盘中读取一行内容
3. 定义HashMap,key是单词,value是字符对应的次数
4. 使用空格分割字符串
5. 遍历分割好的每个字符串
6. 获取拿到每个字符串,使用字符去HashMap中查找次数
7. 如果为空,说明该字符串第一次出现
8. 放入字符串作为key,把次数设置为1
9. 如果之前,已经有字符串了,就把字符的次数+1

**package** com.day09.exam02;

**import** java.util.HashMap;

**import** java.util.Scanner;

**public** **class** Test09 {

**public** **static** **void** main(String[] args) {

HashMap<String, Integer> hashMap = **new** HashMap<>();

Scanner scanner = **new** Scanner(System.***in***);

String[] strArr = scanner.nextLine().split(" ");

**for** (String s : strArr) {

**if** (!hashMap.containsKey(s)) {

hashMap.put(s, 1);

} **else** {

hashMap.put(s, hashMap.get(s) + 1);

}

}

System.***out***.println(hashMap);

}

}



## 训练案例10

### 训练知识点

1.Map集合

### 训练描述

通过键盘录入五个人的姓名,每当录入一个人的姓名,随机给出一个成绩[0, 100](包含0分,也包含100分).将姓名和成绩存储到Map集合中,姓名作为key,成绩作为value.统计五个学生的总分,最高分,最低分,平均分,并将总分,最高分,最低分,平均分,写入到score.txt文件中,格式为: 总分=xxx,最高分=xxx,最低分=xxx,平均分=xxx

### 操作步骤描述

1. 创建键盘输入对象
2. 创建随机数对象
3. 创建HashMap集合key存放姓名,value存放成绩
4. 循环判断HashMap中数量是否小于5
5. 如果HashMap中数量小于5,让用户再输入一个姓名
6. 使用随机数生成一个成绩
7. 将姓名和分数添加到HashMap中
8. 使用迭代器获取一个键
9. 通过键找到对应的分数,作为默认分数
10. 定义总分分变量
11. 定义最高分变量=默认分数
12. 定义最低分变量=默认分数
13. 使用keySet获取到所有的key
14. 使用增强for遍历所有的key
15. 根据key找到对应的value
16. 将这个分数添加到总分
17. 如果最高分小于这个分数,让最高分=这个分数
18. 如果最低分大于这个分数,让最低分=这个分数
19. 计算平均分
20. 定义StringBuilder,用于拼接,总分,最高分,最低分,平均分字符串
21. 创建FileWriter字符输出流
22. 使用字符输出流将拼接好的字符串写到文件中
23. 关闭流

**package** com.day09.exam02;

**import** java.io.BufferedWriter;

**import** java.io.FileWriter;

**import** java.io.IOException;

**import** java.util.HashMap;

**import** java.util.Random;

**import** java.util.Scanner;

**public** **class** Test10 {

**public** **static** **void** main(String[] args) **throws** IOException {

HashMap<String, Integer> hashMap = **new** HashMap<>();

Scanner scanner = **new** Scanner(System.***in***);

Random random = **new** Random();

**for** (**int** i = 0; i < 5; i++) {

String name = scanner.nextLine();

**int** score = random.nextInt(101);

hashMap.put(name, score);

}

**int** sum = 0;

**int** max = 0;

**int** min = 100;

**for** (String s : hashMap.keySet()) {

sum += hashMap.get(s);

**if** (hashMap.get(s) > max) {

max = hashMap.get(s);

}

**if** (hashMap.get(s) < min) {

max = hashMap.get(s);

}

}

**int** average = sum / hashMap.size();

HashMap<String, Integer> hashMap2 = **new** HashMap<>();

hashMap2.put("总分", sum);

hashMap2.put("最高分", max);

hashMap2.put("最低分", min);

hashMap2.put("平均分", average);

BufferedWriter bufferedWriter = **new** BufferedWriter(**new** FileWriter("score.txt"));

**for** (String s : hashMap2.keySet()) {

StringBuffer sb = **new** StringBuffer();

sb.append(s).append("=").append(hashMap2.get(s));

bufferedWriter.write(sb.toString());

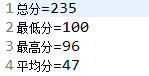
bufferedWriter.newLine();

}

bufferedWriter.close();

}

}



## 训练案例11

### 训练知识点

1.Map集合

### 训练描述

有一个Map集合里面存储了学生的名字,年龄,内容如下:{赵四=21, 王二=17, 张三=21},1将里面的元素使用两种遍历方式打印的控制台,2将赵四的年龄改为18,3将元素存入到student.txt中.一行保存格式:姓名,年龄

### 操作步骤描述

1. 创建HashMap集合key存放姓名,value存放年龄
2. 使用put添加对应的数据
3. 使用keySet遍历Map
4. 使用entrySet遍历Map
5. 使用put修改赵四的年龄为18岁
6. 创建BufferedWriter字符输出流
7. 使用keySet遍历Map
8. 根据key拿到对应的value
9. 使用字符输出流将拼接好的字符串写到文件中
10. 关闭流

**package** com.day09.exam02;

**import** java.io.BufferedWriter;

**import** java.io.FileWriter;

**import** java.io.IOException;

**import** java.util.HashMap;

**import** java.util.Iterator;

**import** java.util.Map;

**public** **class** Test11 {

**public** **static** **void** main(String[] args) **throws** IOException {

HashMap<String, Integer> hashMap = **new** HashMap<>();

hashMap.put("赵四", 21);

hashMap.put("王二", 17);

hashMap.put("张三", 21);

**for** (String s : hashMap.keySet()) {

System.***out***.println(s + "=" + hashMap.get(s));

}

System.***out***.println("----------------");

Iterator<java.util.Map.Entry<String, Integer>> iterator = hashMap.entrySet().iterator();

**while** (iterator.hasNext()) {

Map.Entry<String, Integer> entry = (Map.Entry<String, Integer>) iterator.next();

System.***out***.println(entry.getKey() + "=" + entry.getValue());

}

hashMap.put("赵四", 18);

BufferedWriter bufferedWriter = **new** BufferedWriter(**new** FileWriter("student.txt"));

**for** (String s : hashMap.keySet()) {

StringBuffer sb = **new** StringBuffer();

sb.append(s).append(",").append(hashMap.get(s));

bufferedWriter.write(sb.toString());

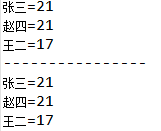
bufferedWriter.newLine();

}

bufferedWriter.close();

}

}





## 训练案例12

### 训练知识点

1.Map集合

### 训练描述

使用键盘循环录入:书名,价格.创建一个Map集合,key存储书名,value存储书的价格. 录入以下5本书数据:{Java=32.5,C++=25,PHP=28.6,UI=21.3,Python=10.2} 要求;

1.删除集合中的C++这本书

2.Java的这本书涨价为38.5元

3.使用keySet和entrySet两种不同方式对集合遍历输出

### 操作步骤描述

1. 创建HashMap集合key存放姓名,value存放年龄
2. 创建键盘录入对象Scanner
3. 循环判断HashSet的数量是否小于5
4. 如果HashSet的数量小于5,让用户输入一本书的信息
5. 使用,分割书名和价格
6. 将书名和价格添加到HashMap中
7. 使用remove方法删除集合中的C++这本书
8. 使用put方法将Java的这本书涨价为38.5元
9. 使用keySet遍历Map
10. 使用entrySet遍历Map

**package** com.day09.exam02;

**import** java.io.IOException;

**import** java.util.HashMap;

**import** java.util.Iterator;

**import** java.util.Map;

**public** **class** Test12 {

**public** **static** **void** main(String[] args) **throws** IOException {

HashMap<String, Double> hashMap = **new** HashMap<>();

hashMap.put("Java", 32.5);

hashMap.put("C++", 25.0);

hashMap.put("PHP", 28.6);

hashMap.put("UI", 21.3);

hashMap.put("Python", 10.2);

System.***out***.println(hashMap);

hashMap.remove("C++");

hashMap.put("Java", 38.5);

**for** (String s : hashMap.keySet()) {

System.***out***.println(s + "=" + hashMap.get(s));

}

System.***out***.println("------------------");

Iterator<java.util.Map.Entry<String, Double>> iterator = hashMap.entrySet().iterator();

**while** (iterator.hasNext()) {

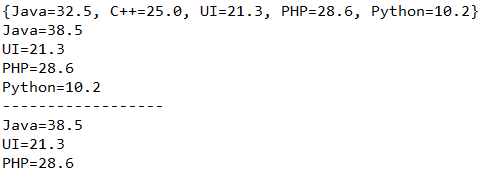
Map.Entry<String, Double> entry = (Map.Entry<String, Double>) iterator.next();

System.***out***.println(entry.getKey() + "=" + entry.getValue());

}

}

}



## 训练案例13

### 训练知识点

1.Map集合

### 训练描述

Map中有{张三=23,李四=24,王五=15,赵六=36,孙七=17,田七=57,周八=35}这些元素key为姓名value为年龄.将周八的年龄修改为22,并将年龄小于30岁人的信息写入到perosn.txt中,person.txt每行数据格式为:姓名=年龄

### 操作步骤描述

1. 创建HashMap集合key存放姓名,value存放年龄
2. 使用put方法添加对应的元素
3. 使用put将周八的年龄修改为22
4. 创建BufferedWriter字符输出流
5. 使用keySet遍历HashMap集合
6. 根据key获取到对应的value
7. 将key和value按格式拼接好写入到person.txt文件中
8. 关闭流

**package** com.day09.exam02;

**import** java.io.BufferedWriter;

**import** java.io.FileWriter;

**import** java.io.IOException;

**import** java.util.HashMap;

**public** **class** Test13 {

**public** **static** **void** main(String[] args) **throws** IOException {

HashMap<String, Integer> hashMap = **new** HashMap<>();

hashMap.put("张三", 23);

hashMap.put("李四", 24);

hashMap.put("王五", 15);

hashMap.put("赵六", 36);

hashMap.put("孙七", 17);

hashMap.put("田七", 57);

hashMap.put("周八", 35);

hashMap.put("周八", 22);

BufferedWriter bufferedWriter = **new** BufferedWriter(**new** FileWriter("perosn.txt"));

**for** (String s : hashMap.keySet()) {

**if** (hashMap.get(s) < 30) {

StringBuffer sb = **new** StringBuffer();

sb.append(s).append(",").append(hashMap.get(s));

bufferedWriter.write(sb.toString());

bufferedWriter.newLine();

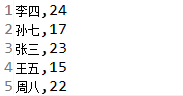
}

}

bufferedWriter.close();

}

}



## 训练案例14

### 训练知识点

1. 字符串分割
2. Map集合

### 训练描述

现有一串字符串”常州极客营,无锡极客营,武汉极客营,镇江极客营,苏州极客营”,要求使用程序统计出”极客营”在该字符串中出现的次数。

### 操作步骤描述

1. 定义传极客营字符串
2. 创建HashMap保存极客营的次数
3. 使用put添加key"极客营"
4. 使用,分割字符串
5. 使用增强for遍历获取每个分割出来的字符串
6. 如果分割后的字符串串包含"极客营",让HashSet中"极客营"的value+1
7. 打印HashMap的内容

**package** com.day09.exam02;

**import** java.util.HashMap;

**public** **class** Test14 {

**public** **static** **void** main(String[] args) {

String str = "常州极客营,无锡极客营,武汉极客营,镇江极客营,苏州极客营";

HashMap<String, Integer> hashMap = **new** HashMap<>();

String strArr[] = str.split(",");

**for** (**int** i = 0; i < strArr.length; i++) {

**if** (strArr[i].contains("极客营")) {

**if** (!hashMap.containsKey("极客营")) {

hashMap.put("极客营", 1);

} **else** {

hashMap.put("极客营", hashMap.get("极客营") + 1);

}

}

}

System.***out***.println(hashMap);

}

}



## 训练案例15

### 训练知识点

1.Map集合

### 训练描述

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, …}

### 操作步骤描述

1. 定义第一个数组arr1
2. 定义第二个数组arr2
3. 创建HashMap,key存放省,value存放市
4. 使用普通for循环遍历arr1
5. 根据索引到arr1中获取到省
6. 根据索引到arr2中获取到省会城市
7. 将省和省会城市添加到HashMap中
8. 输出HashMap中的内容

**package** com.day09.exam02;

**import** java.util.HashMap;

**public** **class** Test15 {

**public** **static** **void** main(String[] args) {

String str1[] = {"黑龙江省","浙江省","江西省","广东省","福建省"};

String str2[] = {"哈尔滨","杭州","南昌","广州","福州"};

HashMap<String, String>hashMap = **new** HashMap<>();

**for** (**int** i = 0; i < str1.length; i++) {

**for** (**int** j = 0; j < str2.length; j++) {

hashMap.put(str1[i], str2[i]);

}

}

System.***out***.println(hashMap);

}

}



# 关卡3

## 训练案例1

### 训练知识点

1.Map集合

### 训练描述

Map中有{张三=3,李四=4,王五=5,赵六=6,孙七=7,田七=7,周八=5}这些元素.将Map中value相同的元素从Map中去除(value为7和5是相同的,都要删除).再将Map中剩余元素的key存放到一个ArrayLis中,并打印到控制台上

**package** com.day09.exam03;

**import** java.io.IOException;

**import** java.util.ArrayList;

**import** java.util.HashMap;

**import** java.util.Iterator;

**import** java.util.Map;

**public** **class** Test01 {

**public** **static** **void** main(String[] args) **throws** IOException {

HashMap<String, Integer> hashMap = **new** HashMap<>();

hashMap.put("张三", 3);

hashMap.put("李四", 4);

hashMap.put("王五", 5);

hashMap.put("赵六", 6);

hashMap.put("孙七", 7);

hashMap.put("田七", 7);

hashMap.put("周八", 5);

HashMap<Integer, Integer> valMap = **new** HashMap<>();

**for** (**int** i : hashMap.values()) {

**if** (!valMap.containsKey(i)) {

valMap.put(i, 1);

} **else** {

valMap.put(i, valMap.get(i) + 1);

}

}

ArrayList<Integer> intArr = **new** ArrayList<>();

**for** (**int** i : valMap.keySet()) {

**if** (valMap.get(i) > 1) {

intArr.add(i);

}

}

Iterator<java.util.Map.Entry<String, Integer>> iterator = hashMap.entrySet().iterator();

**while** (iterator.hasNext()) {

Map.Entry<String, Integer> entry = (Map.Entry<String, Integer>) iterator.next();

**for** (**int** i : intArr) {

**if** (entry.getValue() == i) {

iterator.remove();

}

}

}

System.***out***.println(hashMap);

}

}



## 训练案例2

### 训练知识点

1.HashMap嵌套HashMap

### 训练描述

极客营包含2个班级:Java基础班,Java就业班

Java基础班:

001 李晨

002 范冰冰

Java就业班:

001 马云

002 马化腾

使用HashMap嵌套HashMap方式完成,1使用keySet和增强for遍历出里面的元素,2使用entrySet和迭代器遍历出里面的元素

**package** com.day09.exam03;

**import** java.util.HashMap;

**import** java.util.Iterator;

**import** java.util.Map;

**import** java.util.Map.Entry;

**public** **class** Test02 {

**public** **static** **void** main(String[] args) {

HashMap<String, HashMap<String, String>> hashMap = **new** HashMap<>();

HashMap<String, String> class1 = **new** HashMap<>();

class1.put("001", "李晨");

class1.put("002", "范冰冰");

HashMap<String, String> class2 = **new** HashMap<>();

class2.put("001", "马云");

class2.put("002", "马化腾");

hashMap.put("Java基础班", class1);

hashMap.put("Java就业班", class2);

**for** (String i : hashMap.keySet()) {

**for** (String s : hashMap.get(i).keySet()) {

System.***out***.println(i + ":" + s + ":" + hashMap.get(i).get(s));

}

}

System.***out***.println("-------------------------");

Iterator<Entry<String, HashMap<String, String>>> iterator = hashMap.entrySet().iterator();

**while** (iterator.hasNext()) {

Map.Entry<String, HashMap<String, String>> entry = (Map.Entry<String, HashMap<String, String>>) iterator

.next();

Iterator<Entry<String, String>> iterator2 = entry.getValue().entrySet().iterator();

**while** (iterator2.hasNext()) {

Map.Entry<String, String> entry2 = (Map.Entry<String, String>) iterator2.next();

System.***out***.println(entry.getKey() + ":" + entry2.getKey() + ":" + entry2.getValue());

}

}

}

}



## 训练案例3

### 训练知识点

1.Map集合

### 训练描述

已知Map中保存如下信息:{“及时雨”=”宋江”, “玉麒麟”=”卢俊义”, “智多星”=”吴用”},其中键表示水浒中人物的外号,value表示人物的姓名.1往Map中添加 “入云龙”=”公孙胜”, ”豹子头”=”林冲”两位好汉.2删除“玉麒麟”=”卢俊义”,3将key为“智多星”的value修改为null,4.将“及时雨”=”宋江”,修改为”呼保义”=” 宋江”

**package** com.day09.exam03;

**import** java.util.HashMap;

**public** **class** Test03 {

**public** **static** **void** main(String[] args) {

HashMap<String, String> hashMap = **new** HashMap<>();

hashMap.put("及时雨", "宋江");

hashMap.put("玉麒麟", "卢俊义");

hashMap.put("智多星", "吴用");

hashMap.put("入云龙", "公孙胜");

hashMap.put("豹子头", "林冲");

hashMap.remove("玉麒麟");

hashMap.put("智多星", **null**);

hashMap.remove("及时雨");

hashMap.put("呼保义", "宋江");

System.***out***.println(hashMap);

}

}



## 训练案例4

### 训练知识点

1. HashSet
2. Map

### 训练描述

有四种水果(苹果，香蕉，西瓜，橘子)

1、给每种水果设定一个商品号，商品号是8个0-9的随机数，商品号码不能重复, 最小值 "00000001", 最大值 "99999999"

2、根据商品号查询对应的商品。

如果查不到输出：“查无此商品”

如果能查到打印：”根据商品号：12345678，查询到对应的商品为：西瓜”

**package** com.day09.exam03;

**import** java.util.ArrayList;

**import** java.util.HashMap;

**import** java.util.Random;

**import** java.util.Scanner;

**public** **class** Test04 {

**public** **static** **void** main(String[] args) {

HashMap<String, String> hashMap = **new** HashMap<>();

Random random = **new** Random();

ArrayList<String> arrayList = **new** ArrayList<>();

**for** (**int** i = 0; i < 4; i++) {

StringBuffer sb = **new** StringBuffer();

**for** (**int** j = 0; j < 8; j++) {

sb.append(random.nextInt(10));

}

arrayList.add(sb.toString());

}

hashMap.put(arrayList.get(0), "苹果");

hashMap.put(arrayList.get(1), "香蕉");

hashMap.put(arrayList.get(2), "西瓜");

hashMap.put(arrayList.get(3), "橘子");

System.***out***.println(hashMap);

Scanner scanner = **new** Scanner(System.***in***);

System.***out***.println("输入商品号");

String id = scanner.nextLine();

String realId = **null**;

String name = **null**;

Boolean flag = **false**;

**for**(String s:hashMap.keySet()){

**if**(id.equals(s)){

realId = s;

name = hashMap.get(s);

flag = **true**;

}

}

**if**(flag){

System.***out***.println("根据商品号："+realId+"，查询到对应的商品为"+name);

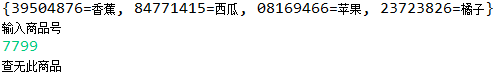
}**else** {

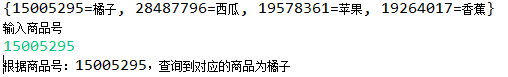
System.***out***.println("查无此商品");

}

}

}





## 训练案例5

### 训练知识点

1. 斗地主洗牌发牌

### 训练描述

完成课上斗地主洗牌发牌案例(牌需要排序)

**package** com.day09.exam03;

**import** java.util.ArrayList;

**import** java.util.Collections;

**import** java.util.HashMap;

**import** java.util.Map;

**public** **class** Test05 {

**public** **static** **void** main(String[] args) {

Map<Integer, String> map = **new** HashMap<Integer, String>();

ArrayList<String> colors = **new** ArrayList<String>();

colors.add("♣");

colors.add("♦");

colors.add("♠");

colors.add("♥");

ArrayList<String> numbers = **new** ArrayList<String>();

**for** (**int** i = 3; i <= 10; i++) {

numbers.add(i + "");

}

Collections.*addAll*(numbers, "J", "Q", "K", "A", "2");

**int** cardNumber = 0;

**for** (String thisNumber : numbers) {

**for** (String thisColor : colors) {

String thisCard = thisColor + thisNumber;

map.put(cardNumber, thisCard);

cardNumber++;

}

}

map.put(cardNumber++, "小☺");

map.put(cardNumber, "大☺");

ArrayList<Integer> poker = **new** ArrayList<Integer>();

**for** (**int** i = 0; i < 54; i++) {

poker.add(i);

}

Collections.*shuffle*(poker);

ArrayList<Integer> player01 = **new** ArrayList<Integer>();

ArrayList<Integer> player02 = **new** ArrayList<Integer>();

ArrayList<Integer> player03 = **new** ArrayList<Integer>();

ArrayList<Integer> dipai = **new** ArrayList<Integer>();

**for** (**int** index = 0; index < poker.size() - 3; index++) {

Integer integerCard = poker.get(index);

**if** (index % 3 == 0) {

player01.add(integerCard);

} **else** **if** (index % 3 == 1) {

player02.add(integerCard);

} **else** {

player03.add(integerCard);

}

}

**for** (**int** index = poker.size() - 3; index < poker.size(); index++) {

Integer integerCard = poker.get(index);

dipai.add(integerCard);

}

Collections.*sort*(player01);

Collections.*sort*(player02);

Collections.*sort*(player03);

Collections.*sort*(dipai);

System.***out***.print("玩家1:");

**for** (**int** i = player01.size() - 1; i >= 0; i--) {

Integer integer = player01.get(i);

String realCard = map.get(integer);

System.***out***.print(realCard + " ");

}

System.***out***.println();

System.***out***.print("玩家2:");

**for** (**int** i = player02.size() - 1; i >= 0; i--) {

Integer integer = player02.get(i);

String realCard = map.get(integer);

System.***out***.print(realCard + " ");

}

System.***out***.println();

System.***out***.print("玩家3:");

**for** (**int** i = player03.size() - 1; i >= 0; i--) {

Integer integer = player03.get(i);

String realCard = map.get(integer);

System.***out***.print(realCard + " ");

}

System.***out***.println();

System.***out***.print("底牌:");

**for** (Integer integer : dipai) {

String realCard = map.get(integer);

System.***out***.print(realCard + " ");

}

}

}

