

# Day12-集合习题答案

## 应用知识点：

- Collection集合框架
- List集合
- Set集合
- Map集合框架

## 基础编程题答案：

1. 参考答案如下：

```
1 package day12;
2 import java.util.Scanner;
3
4 public class Test_01 {
5     public static void main(String[] args) {
6         //创建多个worker对象添加到集合中
7         ArrayList<Worker> list = new ArrayList<Worker>();
8         list.add(new Worker("胡八一", 23, 10000));
9         list.add(new Worker("刘胡兰", 24, 20000));
10        list.add(new Worker("张三", 26, 12000));
11        list.add(new Worker("李四", 20, 9000));
12
13        // //打印所有工人信息
14        for(Worker w:list){
15            System.out.println(w);
16        }
17
18        // 获取平均工资
19        double sum = 0.0;
20        for(Worker w:list){
21            sum += w.getSalary();
22        }
23        System.out.println("平均薪资："+sum/list.size());
24
25        System.out.println("-----姓名中包含胡的工人信息-----");
26        for(Worker w:list){
27            String name = w.getName();
28            if(name.contains("胡")){
```

```

29         System.out.println(w);
30     }
31 }
32
33 System.out.println("----打印输出姓胡的工人信息----");
34 for(Worker w:list){
35     String name = w.getName();
36     if(name.startsWith("胡")){
37         System.out.println(w);
38     }
39 }
40
41 // 键盘输入一个姓名，查找是否存在，存在打印该员工信息，不存在打印“查无
此人“
42 Scanner sc = new Scanner(System.in);
43 System.out.println("请输入您要查询的员工：");
44 String name = sc.next();
45
46 boolean flag = true;//是否打印的标志
47 for(Worker w:list){
48     if(name.equals(w.getName())){
49         flag = false;
50         System.out.println(w);
51     }
52 }
53 if (flag) {
54     System.out.println("查无此人");
55 }
56 System.out.println("-----");
57
58 // 输入一个工资输出大于输入工资员工的信息
59 System.out.println("请输入一个工资：");
60 double salary = sc.nextDouble();
61 for(Worker w : list){
62     if(w.getSalary() > salary){
63         System.out.println(w);
64     }
65 }
66
67 }
68 }
69 //工人类
70 class Worker {
71     private String name;//姓名
72     private int age;//年龄
73     private double salary;//工资

```

```

74
75     public Worker() {
76         super();
77     }
78
79     public Worker(String name, int age, double salary) {
80         super();
81         this.name = name;
82         this.age = age;
83         this.salary = salary;
84     }
85
86     public String getName() {
87         return name;
88     }
89
90     public void setName(String name) {
91         this.name = name;
92     }
93
94     public int getAge() {
95         return age;
96     }
97
98     public void setAge(int age) {
99         this.age = age;
100    }
101
102    public double getSalary() {
103        return salary;
104    }
105
106    public void setSalary(double salary) {
107        this.salary = salary;
108    }
109
110    public String toString() {
111        return "name=" + name + ", age=" + age + ", salary=" +
salary;
112    }
113 }

```

2. 参考答案如下：

```

1 package day12;

```

```

2  import java.util.Scanner;
3
4  public class Test_02 {
5      public static void main(String[] args) {
6          List<Worker> list = new ArrayList<Worker>();
7          list.add(new worker("zhang3", 18, 3000));
8          list.add(new worker("li4", 25, 3500));
9          list.add(new worker("wang5", 22, 3200));
10
11         //在li4之前插入一个工人
12         list.add(1, new worker("zhao6", 23, 3300));
13
14         //删除wang5的信息
15         list.remove(3);
16
17         //利用下标遍历打印输出所有工人信息
18         System.out.println("-----下标遍历-----");
19         for(int i=0;i<list.size();i++){
20             System.out.println(list.get(i));
21         }
22
23         //利用foreach遍历，打印输出所有年龄大于20的工人信息
24         System.out.println("-----forEach遍历-----");
25         for (worker w : list) {
26             if(w.getAge() > 20){
27                 System.out.println(w);
28             }
29         }
30
31         // 利用forEach遍历，打印输出所有年龄大于20的工人信息
32         list.forEach((w)->{
33             if(w.getName().length()>5){
34
35                 System.out.println(w.getName()+"\t"+w.getAge()+"\t"+w.getSalary())
36                 ;
37             }
38         });
39
40         class worker {
41             private String name;
42             private int age;
43             private double salary;
44
45             public worker() {

```

```
46         super();
47     }
48     public Worker(String name, int age, double salary) {
49         super();
50         this.name = name;
51         this.age = age;
52         this.salary = salary;
53     }
54     public String getName() {
55         return name;
56     }
57
58     public void setName(String name) {
59         this.name = name;
60     }
61
62     public int getAge() {
63         return age;
64     }
65
66     public void setAge(int age) {
67         this.age = age;
68     }
69
70     public double getSalary() {
71         return salary;
72     }
73
74     public void setSalary(double salary) {
75         this.salary = salary;
76     }
77
78     public boolean equals(Object obj) {
79         if (this == obj)
80             return true;
81         if (obj == null)
82             return false;
83         if (getClass() != obj.getClass())
84             return false;
85         Worker other = (Worker) obj;
86         if (this.name.equals(other.name) && this.age == other.age
87             && this.salary == other.salary){
88             return true;
89         }else{
90             return false;
91         }
```

```

92     }
93
94     public String toString() {
95         return "name=" + name + ", age=" + age + ", salary=" +
salary;
96     }
97 }

```

### 3. 参考答案如下：

```

1  package day12;
2
3  public class Test_03 {
4      public static void main(String[] args) {
5          // 创建多个商品存储在set集合中，集合中不能出现重复商品所以选择
hashset集合
6          Set<Product> set = new HashSet<Product>();
7          set.add(new Product("卫龙", 2.5, 100, "北京市沙河"));
8          set.add(new Product("加多宝", 4.0, 25, "北京市海淀区"));
9          set.add(new Product("加多宝", 4.0, 20, "天津武清区"));
10         set.add(new Product("龙井", 1500.0, 2, "北京市朝阳区"));
11
12         // 显示所有商品信息
13         for (Product p : set) {
14             System.out.println(p);
15         }
16         System.out.println("-----");
17
18         // 输出价格1000以上的
19         for (Product p : set) {
20             if (p.getPrice() > 1000) {
21                 System.out.println(p);
22             }
23         }
24
25         System.out.println("售空的商品信息如下：");
26         // 打印售空的商品信息
27         for (Product p : set) {
28             if (p.getNum() == 0) {
29                 System.out.println(p);
30             }
31         }
32
33         System.out.println("产地在北京的商品信息：");
34

```

```

35         // 打印产地北京的商品
36         for (Product p : set) {
37             if (p.getAddress().contains("北京")) {
38                 System.out.println(p);
39             }
40         }
41
42         // 输入一个商品名，查询出此类商品的信息，如果不存在，则打印"商场无此产
    品"
43         Scanner sc = new Scanner(System.in);
44         System.out.println("请输入商品名称:");
45         String brand = sc.next();
46         int count = 0; //打印标志
47         for (Product p : set) {
48             if (p.getBrand().equals(brand)) {
49                 count++;
50                 System.out.println(p);
51             }
52         }
53         if (count==0) {
54             System.out.println("商场无此产品");
55         }
56
57         System.out.println("-----");
58
59         // 输入一个价格段，展示所有区间的商品
60         System.out.println("接下来输入按照价格搜索商品");
61         System.out.println("请输入最低价格:");
62         double start = sc.nextDouble();
63         System.out.println("请输入最高价格:");
64         double end = sc.nextDouble();
65         System.out.println("价格在：" + start + "到" + end + "之间的商品信
    息：");
66         for (Product p : set) {
67             if (p.getPrice() >= start && p.getPrice() <= end) {
68                 System.out.println(p);
69             }
70         }
71     }
72 }
73
74 // 商品
75 class Product {
76     private String brand; // 商品名
77     private double price; // 价格
78     private int num; // 商品数量

```

```
79     private String address;// 产地
80     public Product(){}
81     public Product(String brand,double price,int num,String
address){
82         super();
83         this.brand = brand;
84         this.price = price;
85         this.num = num;
86         this.address = address;
87     }
88
89     public String getBrand() {
90         return brand;
91     }
92     public void setBrand(String brand) {
93         this.brand = brand;
94     }
95     public double getPrice() {
96         return price;
97     }
98     public void setPrice(double price) {
99         this.price = price;
100    }
101    public int getNum() {
102        return num;
103    }
104    public void setNum(int num) {
105        this.num = num;
106    }
107    public String getAddress() {
108        return address;
109    }
110    public void setAddress(String address) {
111        this.address = address;
112    }
113
114    public int hashCode(){
115        return this.brand.hashCode();
116    }
117    public boolean equals(Object obj) {
118        if (this == obj)
119            return true;
120        if (obj == null)
121            return false;
122        if (getClass() != obj.getClass())
123            return false;
```



```

124         Product other = (Product) obj;
125         if (this.brand.equals(other.brand)) {
126             return true;
127         } else {
128             return false;
129         }
130     }
131
132     public String toString() {
133         return "brand=" + brand + ", price=" + price + ", num=" +
num
134             + ", address=" + address;
135     }
136 }

```

4. 参考答案如下：

```

1  package day12;
2
3  public class Test_04 {
4      public static void main(String[] args) {
5          Set<Integer> set = new HashSet<Integer>();
6
7          Random r = new Random();
8          /* 每次循环获取一个随机数存储到集合中，如果集合中
9             * 已经存在则不存储，不存在则存储
10             */
11         while (set.size() != 10) {
12             int random = r.nextInt(51);
13             set.add(random);
14         }
15
16         for(Integer i:set){
17             System.out.println(i);
18         }
19     }
20 }

```

5. 参考答案如下：

```

1  package day12;
2  import java.util.*;
3
4  public class Test_05 {
5      public static void main(String[] args) {

```

```

6      Scanner sc = new Scanner(System.in);
7      System.out.println("请输入一个字符串:");
8      String s = sc.next();
9
10     Set<Character> set = new HashSet<Character>();
11
12     for (int i = 0; i < s.length(); i++) {
13         char c = s.charAt(i);
14         set.add(c);
15     }
16     //打印去除重复后的字符
17     for(Character c:set){
18         System.out.println(c);
19     }
20
21     System.out.println("-----");
22     // 如果希望字符 按照字符串中的顺序进行打印输出
23     Set<Character> set2 = new LinkedHashSet<Character>();
24     for (int i = 0; i < s.length(); i++) {
25         char c = s.charAt(i);
26         set2.add(c);
27     }
28     //打印去除重复后的字符
29     for(Character c:set2){
30         System.out.println(c);
31     }
32 }
33 }

```

6. 参考答案如下：

```

1  package day12;
2  import java.util.HashMap;
3  import java.util.Map;
4  import java.util.Scanner;
5  import java.util.Set;
6
7  public class Test_06 {
8      public static void main(String[] args) {
9          // 创建多个用户存储到hashMap集合中
10         Map<Integer, BankUser> map = new HashMap<Integer, BankUser>
11         ();
12         BankUser bu1 = new BankUser(1, "小明", 200000.0);
13         BankUser bu2 = new BankUser(2, "小花", 400000.0);
14         BankUser bu3 = new BankUser(3, "小刚", 600000.0);

```

```
14      BankUser bu4 = new BankUser(4, "小红", 80000.0);
15
16      // id作为键,User对象作为值
17      map.put(bu1.getId(), bu1);
18      map.put(bu2.getId(), bu2);
19      map.put(bu3.getId(), bu3);
20      map.put(bu4.getId(), bu4);
21
22      // 展示所有的name 和对应的余额
23      Set<Integer> keys = map.keySet();
24      for (Integer key : keys) {
25          BankUser user = map.get(key);
26          System.out.println(user.getUserName()+" "
27                              +user.getBalance());
28      }
29
30      // 键盘录入一个name查看是否存在
31      Scanner sc = new Scanner(System.in);
32      System.out.println("请输入一个用户名:");
33      String name = sc.next();
34
35      boolean flag = true;//是否打印的标识
36      for (Integer key : keys) {
37          BankUser u = map.get(key);
38          if (u.getUserName().equals(name)) {
39              System.out.println(u);
40              flag = false;
41          }
42      }
43      if (flag) {
44          System.out.println("查无此人");
45      }
46
47      System.out.println("--余额大于200000的用户信息:");
48      for (Integer key : keys) {
49          BankUser u = map.get(key);
50          if (u.getBalance() > 200000){
51              System.out.println(u);
52          }
53      }
54  }
55 }
56
57 // 银行的用户类
58 class BankUser {
59     private Integer id;
```

```

60     private String userName;
61     private double balance;
62     public BankUser() {}
63     public BankUser(Integer id, String userName, double balance) {
64         this.id = id;
65         this.userName = userName;
66         this.balance = balance;
67     }
68     public Integer getId() {
69         return id;
70     }
71     public void setId(Integer id) {
72         this.id = id;
73     }
74     public String getUserName() {
75         return userName;
76     }
77     public void setUserName(String userName) {
78         this.userName = userName;
79     }
80     public double getBalance() {
81         return balance;
82     }
83     public void setBalance(double balance) {
84         this.balance = balance;
85     }
86     public String toString() {
87         return "id=" + id + ", userName=" + userName + ", balance="
88             + balance;
89     }
90 }

```

7. 参考答案如下：

```

1  package day12;
2  import java.util.*;
3
4  public class Test_07 {
5      public static void main(String[] args) {
6          Map<String,String> worldCupMap=new HashMap<String,String>
7          ();
8          worldCupMap.put("1930", "乌拉圭");
9          worldCupMap.put("1934", "意大利");
10         worldCupMap.put("1938", "意大利");
11         worldCupMap.put("1950", "乌拉圭");

```

```

11     worldCupMap.put("1954", "德国");
12     worldCupMap.put("1958", "巴西");
13     worldCupMap.put("1962", "巴西");
14     worldCupMap.put("1966", "英格兰");
15     worldCupMap.put("1970", "巴西");
16     worldCupMap.put("1974", "德国");
17     worldCupMap.put("1978", "阿根廷");
18     worldCupMap.put("1982", "意大利");
19     worldCupMap.put("1986", "阿根廷");
20     worldCupMap.put("1990", "德国");
21     worldCupMap.put("1994", "巴西");
22     worldCupMap.put("1998", "法国");
23     worldCupMap.put("2002", "巴西");
24     worldCupMap.put("2006", "意大利");
25
26     Scanner sc = new Scanner(System.in);
27     System.out.println("请输入年份: ");
28     String year = sc.next();
29     if(worldCupMap.containsKey(year)){
30         System.out.println(worldCupMap.get(year));
31     }else{
32         System.out.println("没有举办");
33     }
34 }
35 }

```

8. 参考答案如下：

```

1 package day12;
2 import java.util.*;
3
4 public class Test_08 {
5     public static void main(String[] args) {
6         Map<String, String> worldCupMap=new HashMap<String,String>
7         ();
8         worldCupMap.put("1930", "乌拉圭");
9         worldCupMap.put("1934", "意大利");
10        worldCupMap.put("1938", "意大利");
11        worldCupMap.put("1950", "乌拉圭");
12        worldCupMap.put("1954", "德国");
13        worldCupMap.put("1958", "巴西");
14        worldCupMap.put("1962", "巴西");
15        worldCupMap.put("1966", "英格兰");
16        worldCupMap.put("1970", "巴西");
17        worldCupMap.put("1974", "德国");

```

```

17     worldCupMap.put("1978", "阿根廷");
18     worldCupMap.put("1982", "意大利");
19     worldCupMap.put("1986", "阿根廷");
20     worldCupMap.put("1990", "德国");
21     worldCupMap.put("1994", "巴西");
22     worldCupMap.put("1998", "法国");
23     worldCupMap.put("2002", "巴西");
24     worldCupMap.put("2006", "意大利");
25
26     Scanner sc = new Scanner(System.in);
27     System.out.println("请输入国家: ");
28     String value = sc.next();
29
30     if(worldCupMap.containsValue(value)){
31         Set<String> keys = worldCupMap.keySet();
32         for (String key : keys) {
33             String value2 = worldCupMap.get(key);
34             if(value.equals(value2)){
35                 System.out.println(key);
36             }
37         }
38     }else{
39         System.out.println("没有获得过");
40     }
41 }
42 }

```

9. 参考答案如下：

```

1  package day12;
2  import java.util.*;
3
4  public class Test_09 {
5      public static void main(String[] args) {
6          Map<String, String> map = new HashMap<String, String>();
7          map.put("Tom", "CoreJava");
8          map.put("John", "Oracle");
9          map.put("Susan", "Oracle");
10         map.put("Jerry", "JDBC");
11         map.put("Jim", "Unix");
12         map.put("Kevin", "JSP");
13         map.put("Lucy", "JSP");
14
15         //增加了一个Allen教JDBC
16         map.put("Allen", "JDBC");

```

```

17
18      //Lucy改教CoreJava
19      map.put("Lucy", "CoreJava");
20
21      //遍历Map
22      Set<String> set = map.keySet();
23      for(String key:set){
24          String value = map.get(key);
25          System.out.println(key+" "+value);
26      }
27
28      //输出所有教JSP的老师
29      System.out.println("-----JSP老师-----");
30      for(String key:set){
31          String value = map.get(key);
32          if("JSP".equals(value)){
33              System.out.println(key+" "+value);
34          }
35      }
36
37      System.out.print("教授CoreJava和JDBC的老师人数为：");
38      int count = 0;
39      for (String key : set) {
40          String value = map.get(key);
41          if("CoreJava".equals(value) || "JDBC".equals(value)){
42              count++;
43          }
44      }
45
46      System.out.println(count);
47  }
48  }

```

## 概念分析题答案：

10. Collection 接口的特点是**存储任意类型的多个对象**；

List 接口的特点：元素**有**（有|无）顺序，**可以**（可以|不可以）重复；

Set 接口的特点：元素**无**（有|无）顺序，**不可以**（可以|不可以）重复；

Map 接口的特点：元素是**键值对**，其中**值**可以重复，**键**不可以重复。

11. **B、C**

解析：A选项：List是接口，不能创建对象

D选项：List中定义了独有的方法

12. B

13. 程序运行输出结果为：

Hello

Learn

14. 简单介绍ArrayList、LinkedList、Vector的特性及区别。

(1) ArrayList:底层数组，查询快，增删慢，线程不安全，效率高。

(2) Vector:底层数组，查询快，增删慢，线程安全，效率低。

(3) LinkedList:底层链表，查询慢，增删快，线程不安全，效率高。

15. 两者的区别：

(1) List< String> list1= new ArrayList< String>(); 只能存储String类型数据。

(2) List list2 =new ArrayList(); 可以存储Object类型的数据。

< String>作用：泛型的应用，用于约定集合中存储的数据类型

16. B

解析：hashCode方法覆盖的原则：内容相同的对象返回相同的哈希码值；为了提高效率，尽可能做到内容不同的对象返回不同的哈希码值。所以通常会将体现对象内容的属性拼凑为一个int类型的结果作为返回值返回。

第(3)种方式：super.hashCode() 代表调用父类Object中的hashCode方法，父类中hashCode是由对象在内存中的地址转换的结果，和对象的内容无关。

17. C

解析：Set是Collection的子接口，特点是无序、无下标、元素内容不允许重复

18. 程序中不正确的地方：

(1) 覆盖hashCode方法时，访问修饰符必须为public

(2) 对于hashCode的返回值类型为int，而其方法实现中由于salary为double类型，所以return 返回的结果为double类型(自动类型提升)。

对于hashCode方法的修改如下：

```
1 public int hashCode(){
2     return name.hashCode()+age+(int)salary;
3 }
```

(3) equals方法的参数类型为Object，同时需要满足覆盖的5个步骤

(4) Set是没有下标，所以 set.add(0,new Worker("jerry",18,2000)); 使用不正确，



修改为普通的添加：`set.add(new Worker("jerry",18,2000))`。

19. C

解析：考查点为 `addAll(Collection c)`：接收 `Collection` 类型的数据，`List` 是 `Collection` 的子接口，所以可以作为参数进行传递(多态用在形式参数上)；同时 `Set` 是无序、无下标、元素内容不允许重复。

20. 关于下列 Map 接口中常见的方法：

- (1) `put` 方法表示放入一个键值对，如果键已存在则：**新的value数据覆盖原有的value数据，被覆盖的value作为返回值进行返回**；如果键不存在则**直接存储，返回值为null**；
- (2) `remove` 方法接受1个参数，表示**通过键删除这组键值对**；
- (3) `get` 方法表示**通过键获取对应的值**，`get` 方法的参数表示**一个键**，返回值表示**该键对应的值**；
- (4) 要想获得 Map 中所有的键，应该使用方法**`keySet()`**，该方法返回值类型为**`Set`**；
- (5) 要想获得 Map 中所有的值，应该使用方法**`values()`**，该方法返回值类型为**`Collection`**

21. B

解析：A选项：`HashMap`和`Hashtable`都是Map接口的实现类  
C选项：`HashMap`线程不安全，`Hashtable`线程安全

## 能力扩展题答案：

22. 参考答案如下：

```
1 package day12;
2 import java.util.*;
3
4 public class Test_09 {
5     public static void main(String[] args) {
6         // 目前的课程
7         Course c1 = new Course(1,"生物");
8         Course c2 = new Course(2,"化学");
9         Course c3 = new Course(3,"地理");
10        Course c4 = new Course(4,"历史");
11        Course c5 = new Course(5,"政治");
12        Course c6 = new Course(6,"物理");
13
14        // 目前的学生和所选择的课程
15        Student s1 = new Student("杨颖",30);
```

```

16      // 1. 该生选择的课程为：生物、化学、物理
17      List<Course> list1 = s1.getCourses();
18      list1.add(c1);
19      list1.add(c2);
20      list1.add(c6);
21
22      Student s2 = new Student("郑凯",35);
23      // 2. 该生选择的课程为：化学、历史、物理
24      List<Course> list2 = s2.getCourses();
25      list2.add(c1);
26      list2.add(c4);
27      list2.add(c6);
28
29      Student s3 = new Student("邓紫棋",28);
30      // 3. 该生选择的课程为：化学、历史、物理
31      List<Course> list3 = s3.getCourses();
32      list3.add(c2);
33      list3.add(c4);
34      list3.add(c6);
35
36      List<Student> list = new ArrayList<Student>();
37      // 4. 将以上的所有的学生存储在List集合中
38      list.add(s1);
39      list.add(s2);
40      list.add(s3);
41
42      // 5. 遍历输出所有学生的姓名和对应的课程名称
43      for(Student s:list){
44          System.out.print(s.getName()+"选择的课程为：");
45          // 该生选择的课程
46          List<Course> cs = s.getCourses();
47          for(Course c:cs){
48              System.out.print(c.getCourseName()+" ");
49          }
50          System.out.println();
51      }
52  }
53 }
54
55 class Student{
56     private String name;
57     private Integer age;
58     private List<Course> courses = new ArrayList<Course>();
59     public Student() {}
60     public Student(String name, Integer age) {
61         super();

```

```

62         this.name = name;
63         this.age = age;
64     }
65     public Student(String name, Integer age, List<Course> courses)
66     {
67         super();
68         this.name = name;
69         this.age = age;
70         this.courses = courses;
71     }
72     public String getName() {
73         return name;
74     }
75     public void setName(String name) {
76         this.name = name;
77     }
78     public Integer getAge() {
79         return age;
80     }
81     public void setAge(Integer age) {
82         this.age = age;
83     }
84     public List<Course> getCourses() {
85         return courses;
86     }
87     public void setCourses(List<Course> courses) {
88         this.courses = courses;
89     }
90     public String toString() {
91         return "name=" + name + ", age=" + age + ", courses="
92             + courses ;
93     }
94 }
95 // 课程
96 class Course{
97     private Integer courseId;
98     private String courseName;
99     public Course() {}
100    public Course(Integer courseId, String courseName) {
101        this.courseId = courseId;
102        this.courseName = courseName;
103    }
104    public Integer getCourseId() {
105        return courseId;
106    }
107    public void setCourseId(Integer courseId) {

```

```

107         this.courseId = courseId;
108     }
109     public String getCourseName() {
110         return courseName;
111     }
112     public void setCourseName(String courseName) {
113         this.courseName = courseName;
114     }
115     public String toString() {
116         return "courseId=" + courseId + ", courseName=" +
courseName;
117     }
118 }

```

23. 参考答案如下：

```

1  package day12;
2  import java.util.*;
3
4  public class Test_09 {
5      public static void main(String[] args) {
6          String str = "ewrwrw4356436sgdsg6t5gdsa";
7          //Map：键存储字符串中的出现的字符，值存储对应字符出现的次数
8          Map<Character,Integer> map = new HashMap<Character,Integer>
();
9
10         // 遍历字符串：获取每一个字符
11         for (int i = 0; i < str.length(); i++) {
12             char c = str.charAt(i); //获取字符
13             // 判断该字符是否出现过(Map的键)
14             if (map.containsKey(c)) {
15                 // 出现过：原有次数上加1
16                 int m = map.get(c);
17                 map.put(c, m + 1);
18             } else {
19                 // 第一次出现：个数为1
20                 map.put(c, 1);
21             }
22         }
23         // 遍历Map集合
24         Set<Character> keys = map.keySet();
25         for (char c : keys) {
26             int i = map.get(c);
27             System.out.println(c + " 出现的次数： " + i);
28         }

```

```
29     }  
30 }
```