# 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：

```java
package day12;
import java.util.Scanner;

public class Test_01 {
    public static void main(String[] args) {
        //创建多个worker对象添加到集合中
        ArrayList<Worker> list = new ArrayList<Worker>();
        list.add(new Worker("胡八一", 23, 10000));
        list.add(new Worker("刘胡兰", 24, 20000));
        list.add(new Worker("张三", 26, 12000));
        list.add(new Worker("李四", 20, 9000));

        // //打印所有工人信息
        for(Worker w:list){
            System.out.println(w);
        }

        // 获取平均工资
        double sum = 0.0;
        for(Worker w:list){
            sum += w.getSalary();
        }
        System.out.println("平均薪资："+sum/list.size());

        System.out.println("-----姓名中包含胡的工人信息-----");
        for(Worker w:list){
            String name = w.getName();
            if(name.contains("胡")){
                System.out.println(w);
            }
        }

        System.out.println("----打印输出姓胡的工人信息----");
        for(Worker w:list){
            String name = w.getName();
            if(name.startsWith("胡")){
                System.out.println(w);
            }
        }

        // 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“
        Scanner sc = new Scanner(System.in);
        System.out.println("请输入您要查询的员工: ");
        String name = sc.next();

        boolean flag = true;//是否打印的标志
        for(Worker w:list){
            if(name.equals(w.getName())){
                flag = false;
                System.out.println(w);
            }
        }
        if (flag) {
            System.out.println("查无此人");
        }
        System.out.println("------------");

        // 输入一个工资输出大于输入工资员工的信息
        System.out.println("请输入一个工资：");
        double salary = sc.nextDouble();
        for(Worker w : list){
            if(w.getSalary() > salary){
                System.out.println(w);
            }
        }

    }
}
//工人类
class Worker {
    private String name;//姓名
    private int age;//年龄
    private double salary;//工资

    public Worker() {
        super();
    }

    public Worker(String name, int age, double salary) {
        super();
        this.name = name;
        this.age = age;
        this.salary = salary;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }

    public double getSalary() {
        return salary;
    }

    public void setSalary(double salary) {
        this.salary = salary;
    }

    public String toString() {
        return "name=" + name + ", age=" + age + ", salary=" + salary;
    }
}
```

2. 参考答案如下：

```java
package day12;
```

```java
import java.util.Scanner;

public class Test_02 {
    public static void main(String[] args) {
        List<Worker> list = new ArrayList<Worker>();
        list.add(new Worker("zhang3", 18, 3000));
        list.add(new Worker("li4", 25, 3500));
        list.add(new Worker("wang5", 22, 3200));

        //在li4之前插入一个工人
        list.add(1, new Worker("zhao6", 23, 3300));

        //删除wang5的信息
        list.remove(3);

        //利用下标遍历打印输出所有工人信息
        System.out.println("--------下标遍历--------");
        for(int i=0;i<list.size();i++){
            System.out.println(list.get(i));
        }

        //利用foreach遍历，打印输出所有年龄大于20的工人信息
        System.out.println("-------forEach遍历-------");
        for (Worker w : list) {
            if(w.getAge() > 20){
                System.out.println(w);
            }
        }

        // 利用forEach遍历，打印输出所有年龄大于20的工人信息
        list.forEach((w)->{
            if(w.getName().length()>5){
 System.out.println(w.getName()+"\t"+w.getAge()+"\t"+w.getSalary());
            }
        });
    }
}

class Worker {
    private String name;
    private int age;
    private double salary;

    public Worker() {
        super();
    }
    public Worker(String name, int age, double salary) {
        super();
        this.name = name;
        this.age = age;
        this.salary = salary;
    }
    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }

    public double getSalary() {
        return salary;
    }

    public void setSalary(double salary) {
        this.salary = salary;
    }

    public boolean equals(Object obj) {
        if (this == obj)
            return true;
        if (obj == null)
            return false;
        if (getClass() != obj.getClass())
            return false;
        Worker other = (Worker) obj;
        if(this.name.equals(other.name) && this.age == other.age
                           && this.salary == other.salary){
            return true;
        }else{
            return false;
        }
    }

    public String toString() {
        return "name=" + name + ", age=" + age + ", salary=" + salary;
    }
}
```

3. 参考答案如下：

```java
package day12;

public class Test_03 {
    public static void main(String[] args) {
        // 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
        Set<Product> set = new HashSet<Product>();
        set.add(new Product("卫龙", 2.5, 100, "北京市沙河"));
        set.add(new Product("加多宝", 4.0, 25, "北京市海淀区"));
        set.add(new Product("加多宝", 4.0, 20, "天津武清区"));
        set.add(new Product("龙井", 1500.0, 2, "北京市朝阳区"));

        // 显示所有商品信息
        for (Product p : set) {
            System.out.println(p);
        }
        System.out.println("--------------");

        // 输出价格1000以上的
        for (Product p : set) {
            if (p.getPrice() > 1000) {
                System.out.println(p);
            }
        }

        System.out.println("售空的商品信息如下：");
        // 打印售空的商品信息
        for (Product p : set) {
            if (p.getNum() == 0) {
                System.out.println(p);
            }
        }

        System.out.println("产地在北京的商品信息：");

        // 打印产地北京的商品
        for (Product p : set) {
            if (p.getAddress().contains("北京")) {
                System.out.println(p);
            }
        }

        // 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"
        Scanner sc = new Scanner(System.in);
        System.out.println("请输入商品名称:");
        String brand = sc.next();
        int count = 0;//打印标志
        for (Product p : set) {
            if (p.getBrand().equals(brand)) {
                count++;
                System.out.println(p);
            }
        }
        if (count==0) {
            System.out.println("商场无此产品");
        }

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

        // 输入一个价格段，展示所有区间的商品
        System.out.println("接下来输入按照价格搜索商品");
        System.out.println("请输入最低价格:");
        double start = sc.nextDouble();
        System.out.println("请输入最高价格:");
        double end = sc.nextDouble();
        System.out.println("价格在："+start+"到"+end+"之间的商品信息：");
        for (Product p : set) {
            if (p.getPrice() >= start && p.getPrice() <= end) {
                System.out.println(p);
            }
        }
    }
}

// 商品
class Product {
    private String brand;// 商品名
    private double price;// 价格
    private int num;// 商品数量
    private String address;// 产地
    public Product(){}
    public Product(String brand,double price,int num,String address){
        super();
        this.brand = brand;
        this.price = price;
        this.num = num;
        this.address = address;
    }

    public String getBrand() {
        return brand;
    }
    public void setBrand(String brand) {
        this.brand = brand;
    }
    public double getPrice() {
        return price;
    }
    public void setPrice(double price) {
        this.price = price;
    }
    public int getNum() {
        return num;
    }
    public void setNum(int num) {
        this.num = num;
    }
    public String getAddress() {
        return address;
    }
    public void setAddress(String address) {
        this.address = address;
    }

    public int hashCode(){
        return this.brand.hashCode();
    }
    public boolean equals(Object obj) {
        if (this == obj)
            return true;
        if (obj == null)
            return false;
        if (getClass() != obj.getClass())
            return false;
        Product other = (Product) obj;
        if (this.brand.equals(other.brand)) {
            return true;
        } else {
            return false;
        }
    }

    public String toString() {
        return "brand=" + brand + ", price=" + price + ", num=" + num
                + ", address=" + address;
    }
}
```

4. 参考答案如下：

```
package day12;

public class Test_04 {
    public static void main(String[] args) {
        Set<Integer> set = new HashSet<Integer>();

        Random r = new Random();
        /* 每次循环获取一个随机数存储到集合中，如果集合中
         * 已经存在则不存储，不存在则存储
        */
        while (set.size()!=10) {
            int random = r.nextInt(51);
            set.add(random);
        }

        for(Integer i:set){
            System.out.println(i);
        }
    }
}
```

5. 参考答案如下：

```
package day12;
import java.util.*;

public class Test_05 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("请输入一个字符串:");
        String s = sc.next();

        Set<Character> set = new HashSet<Character>();

        for (int i = 0; i < s.length(); i++) {
            char c = s.charAt(i);
            set.add(c);
        }
        //打印去除重复后的字符
        for(Character c:set){
            System.out.println(c);
        }

        System.out.println("--------------------------------");
        // 如果希望字符 按照字符串中的顺序进行打印输出
        Set<Character> set2 = new LinkedHashSet<Character>();
        for (int i = 0; i < s.length(); i++) {
            char c = s.charAt(i);
            set2.add(c);
        }
        //打印去除重复后的字符
        for(Character c:set2){
            System.out.println(c);
        }
    }
}
```

6. 参考答案如下：

```
package day12;
import java.util.HashMap;
import java.util.Map;
import java.util.Scanner;
import java.util.Set;

public class Test_06 {
    public static void main(String[] args) {
         // 创建多个用户存储到hashMap集合中
        Map<Integer, BankUser> map = new HashMap<Integer, BankUser>();
        BankUser bu1 = new BankUser(1, "小明", 200000.0);
        BankUser bu2 = new BankUser(2, "小花", 400000.0);
        BankUser bu3 = new BankUser(3, "小刚", 600000.0);
```

```java
        BankUser bu4 = new BankUser(4, "小红", 80000.0);

        // id作为键,User对象作为值
        map.put(bu1.getId(), bu1);
        map.put(bu2.getId(), bu2);
        map.put(bu3.getId(), bu3);
        map.put(bu4.getId(), bu4);

        // 展示所有的name 和对应的余额
        Set<Integer> keys = map.keySet();
        for (Integer key : keys) {
            BankUser user = map.get(key);
            System.out.println(user.getUserName()+"  "
                               +user.getBalance());
        }

        // 键盘录入一个name查看是否存在
        Scanner sc = new Scanner(System.in);
        System.out.println("请输入一个用户名:");
        String name = sc.next();

        boolean flag = true;//是否打印的标识
        for (Integer key : keys) {
            BankUser u = map.get(key);
            if (u.getUserName().equals(name)) {
                System.out.println(u);
                flag = false;
            }
        }
        if (flag) {
            System.out.println("查无此人");
        }

        System.out.println("--余额大于200000的用户信息：");
        for (Integer key : keys) {
            BankUser u = map.get(key);
            if (u.getBalance() > 200000){
                System.out.println(u);
            }
        }
    }
}

// 银行的用户类
class BankUser {
    private Integer id;
    private String userName;
    private double balance;
    public BankUser() {}
    public BankUser(Integer id, String userName, double balance) {
        this.id = id;
        this.userName = userName;
        this.balance = balance;
    }
    public Integer getId() {
        return id;
    }
    public void setId(Integer id) {
        this.id = id;
    }
    public String getUserName() {
        return userName;
    }
    public void setUserName(String userName) {
        this.userName = userName;
    }
    public double getBalance() {
        return balance;
    }
    public void setBalance(double balance) {
        this.balance = balance;
    }
    public String toString() {
        return "id=" + id + ", userName=" + userName + ", balance="
                + balance;
    }
}
```

7. 参考答案如下：

```java
package day12;
import java.util.*;

public class Test_07 {
    public static void main(String[] args) {
        Map<String,String> worldCupMap=new HashMap<String,String>();
        worldCupMap.put("1930", "乌拉圭");
        worldCupMap.put("1934", "意大利");
        worldCupMap.put("1938", "意大利");
        worldCupMap.put("1950", "乌拉圭");
        worldCupMap.put("1954", "德国");
        worldCupMap.put("1958", "巴西");
        worldCupMap.put("1962", "巴西");
        worldCupMap.put("1966", "英格兰");
        worldCupMap.put("1970", "巴西");
        worldCupMap.put("1974", "德国");
        worldCupMap.put("1978", "阿根廷");
        worldCupMap.put("1982", "意大利");
        worldCupMap.put("1986", "阿根廷");
        worldCupMap.put("1990", "德国");
        worldCupMap.put("1994", "巴西");
        worldCupMap.put("1998", "法国");
        worldCupMap.put("2002", "巴西");
        worldCupMap.put("2006", "意大利");

        Scanner sc = new Scanner(System.in);
        System.out.println("请输入年份: ");
        String year = sc.next();
        if(worldCupMap.containsKey(year)){
            System.out.println(worldCupMap.get(year));
        }else{
            System.out.println("没有举办");
        }
    }
}
```

8. 参考答案如下：

```
package day12;
import java.util.*;

public class Test_08 {
    public static void main(String[] args) {
        Map<String, String> worldCupMap=new HashMap<String,String>();
        worldCupMap.put("1930", "乌拉圭");
        worldCupMap.put("1934", "意大利");
        worldCupMap.put("1938", "意大利");
        worldCupMap.put("1950", "乌拉圭");
        worldCupMap.put("1954", "德国");
        worldCupMap.put("1958", "巴西");
        worldCupMap.put("1962", "巴西");
        worldCupMap.put("1966", "英格兰");
        worldCupMap.put("1970", "巴西");
        worldCupMap.put("1974", "德国");
        worldCupMap.put("1978", "阿根廷");
        worldCupMap.put("1982", "意大利");
        worldCupMap.put("1986", "阿根廷");
        worldCupMap.put("1990", "德国");
        worldCupMap.put("1994", "巴西");
        worldCupMap.put("1998", "法国");
        worldCupMap.put("2002", "巴西");
        worldCupMap.put("2006", "意大利");

        Scanner sc = new Scanner(System.in);
        System.out.println("请输入国家: ");
        String value = sc.next();

        if(worldCupMap.containsValue(value)){
            Set<String> keys = worldCupMap.keySet();
            for (String key : keys) {
                String value2 = worldCupMap.get(key);
                if(value.equals(value2)){
                    System.out.println(key);
                }
            }
        }else{
            System.out.println("没有获得过");
        }
    }
}
```

9. 参考答案如下：

```
package day12;
import java.util.*;

public class Test_09 {
    public static void main(String[] args) {
        Map<String, String> map = new HashMap<String, String>();
        map.put("Tom", "CoreJava");
        map.put("John", "Oracle");
        map.put("Susan", "Oracle");
        map.put("Jerry", "JDBC");
        map.put("Jim", "Unix");
        map.put("Kevin", "JSP");
        map.put("Lucy", "JSP");

        //增加了一个Allen教JDBC
        map.put("Allen", "JDBC");

        //Lucy改教CoreJava
        map.put("Lucy", "CoreJava");

        //遍历Map
        Set<String> set = map.keySet();
        for(String key:set){
            String value = map.get(key);
            System.out.println(key+"   "+value);
        }

        //输出所有教JSP的老师
        System.out.println("------JSP老师------");
        for(String key:set){
            String value = map.get(key);
            if("JSP".equals(value)){
                System.out.println(key+"   "+value);
            }
        }

        System.out.print("教授CoreJava和JDBC的老师人数为：");
        int count = 0;
        for (String key : set) {
            String value = map.get(key);
            if("CoreJava".equals(value)||"JDBC".equals(value)){
                count++;
            }
        }

        System.out.println(count);
    }
}
```

## 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：**

    -
      ```
      public int hashCode(){
          return name.hashCode()+age+(int)salary;
      }
      ```

    **(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：

```java
package day12;
import java.util.*;

public class Test_09 {
    public static void main(String[] args) {
        // 目前的课程
        Course c1 = new Course(1,"生物");
        Course c2 = new Course(2,"化学");
        Course c3 = new Course(3,"地理");
        Course c4 = new Course(4,"历史");
        Course c5 = new Course(5,"政治");
        Course c6 = new Course(6,"物理");

        // 目前的学生和所选择的课程
        Student s1 = new Student("杨颖",30);
        // 1. 该生选择的课程为：生物、化学、物理
        List<Course> list1 = s1.getCourses();
        list1.add(c1);
        list1.add(c2);
        list1.add(c6);

        Student s2 = new Student("郑凯",35);
        // 2. 该生选择的课程为：化学、历史、物理
        List<Course> list2 = s2.getCourses();
        list2.add(c1);
        list2.add(c4);
        list2.add(c6);

        Student s3 = new Student("邓紫棋",28);
        // 3. 该生选择的课程为：化学、历史、物理
        List<Course> list3 = s3.getCourses();
        list3.add(c2);
        list3.add(c4);
        list3.add(c6);

        List<Student> list = new ArrayList<Student>();
        // 4. 将以上的所有的学生存储在List集合中
        list.add(s1);
        list.add(s2);
        list.add(s3);

        // 5. 遍历输出所有学生的姓名和对应的课程名称
        for(Student s:list){
            System.out.print(s.getName()+"选择的课程为：");
            // 该生选择的课程
            List<Course> cs = s.getCourses();
            for(Course c:cs){
                System.out.print(c.getCourseName()+"  ");
            }
            System.out.println();
        }
    }
}

class Student{
    private String name;
    private Integer age;
    private List<Course> courses = new ArrayList<Course>();
    public Student() {}
    public Student(String name, Integer age) {
        super();
        this.name = name;
        this.age = age;
    }
    public Student(String name, Integer age, List<Course> courses) {
        super();
        this.name = name;
        this.age = age;
        this.courses = courses;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public Integer getAge() {
        return age;
    }
    public void setAge(Integer age) {
        this.age = age;
    }
    public List<Course> getCourses() {
        return courses;
    }
    public void setCourses(List<Course> courses) {
        this.courses = courses;
    }
    public String toString() {
        return "name=" + name + ", age=" + age + ", courses="
                + courses ;
    }
}
// 课程
class Course{
    private Integer courseId;
    private String courseName;
    public Course() {}
    public Course(Integer courseId, String courseName) {
        this.courseId = courseId;
        this.courseName = courseName;
    }
    public Integer getCourseId() {
        return courseId;
    }
    public void setCourseId(Integer courseId) {
        this.courseId = courseId;
    }
    public String getCourseName() {
        return courseName;
    }
    public void setCourseName(String courseName) {
        this.courseName = courseName;
    }
    public String toString() {
        return "courseId=" + courseId + ", courseName=" + courseName;
    }
}
```

23. 参考答案如下：

```java
package day12;
import java.util.*;

public class Test_09 {
    public static void main(String[] args) {
        String str = "ewrwrw4356436sgdsg6t5gdsa";
        //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
        Map<Character,Integer> map = new HashMap<Character,Integer>();

        // 遍历字符串：获取每一个字符
        for (int i = 0; i < str.length(); i++) {
            char c = str.charAt(i);//获取字符
            // 判断该字符是否出现过(Map的键)
            if (map.containsKey(c)) {
                // 出现过：原有次数上加1
                int m = map.get(c);
                map.put(c, m + 1);
            } else {
                // 第一次出现：个数为1
                map.put(c, 1);
            }
        }
        // 遍历Map集合
        Set<Character> keys = map.keySet();
        for (char c : keys) {
            int i = map.get(c);
            System.out.println(c + " 出现的次数：  " + i);
        }
    }
}
```