

day10-综合练习

练习1

简易计算器

创建一个简单的控制台计算器程序，可以：

- 让用户选择加、减、乘、除运算
- 输入两个数字
- 输出运算结果

效果如下：

```
-----黑马计算器-----
1. +
2. -
3. ×
4. ÷
请选择算法：
3
请输入第一个数：
2
请输入第二个数：
3
2 × 3 = 6
```

```
public static void main(String[] args) {
    System.out.println("-----黑马计算器-----");
    System.out.println("1. +");
    System.out.println("2. -");
    System.out.println("3. ×");
    System.out.println("4. ÷");
    System.out.println("请选择算法：");
    Scanner sc = new Scanner(System.in);
    String option = sc.next();
    System.out.println("请输入第一个数：");
    int num1 = sc.nextInt();
    System.out.println("请输入第二个数：");
    int num2 = sc.nextInt();

    switch (option){
```

```
        case "1":
            System.out.println(num1 + " + " + num2 + " = " + (num1 + num2));
            break;
        case "2":
            System.out.println(num1 + " - " + num2 + " = " + (num1 - num2));
            break;
        case "3":
            System.out.println(num1 + " x " + num2 + " = " + (num1 * num2));
            break;
        case "4":
            System.out.println(num1 + " ÷ " + num2 + " = " + (num1 / num2));
            break;
        default:
            System.out.println("输入有误!");
            break;
    }
}
```

练习2

数组与对象结合Book[] arr

创建一个 **Book** 类，包含书名、作者和价格属性。在主程序中：

- 创建一个Book数组存储3本书的信息
- 遍历显示所有图书信息
- 找出并输出价格最高的书
- 计算所有书的平均价格
- 让用户输入一个价格，输出所有高于该价格的书籍（只能循环一次!）

效果如下：

情况一：

```
-----  
三国演义...罗贯中... 188  
西游记...吴承恩... 288  
红楼梦...曹雪芹... 198  
-----
```

```
价格最高的书为：西游记，售价：288  
所有图书平均价格为：224  
请输入一个价格：  
300  
暂无图书售价高于300
```

情况二：

```
-----  
三国演义...罗贯中... 188  
西游记...吴承恩... 288  
红楼梦...曹雪芹... 198  
-----
```

```
价格最高的书为：西游记，售价：288  
所有图书平均价格为：224  
请输入一个价格：  
100
```

售价高于100的图书信息如下：

```
三国演义...罗贯中... 188  
西游记...吴承恩... 288  
红楼梦...曹雪芹... 198
```

```
public class Test2 {  
    public static void main(String[] args) {  
        Book[] books = new Book[3];  
        books[0] = new Book("三国演义", "罗贯中", 188);  
        books[1] = new Book("西游记", "吴承恩", 288);  
        books[2] = new Book("红楼梦", "曹雪芹", 198);  
  
        System.out.println("-----");  
  
        // 显示所有图书信息  
        for (int i = 0; i < books.length; i++) {  
            Book book = books[i];  
            book.show();  
        }  
    }  
}
```

```

        System.out.println("-----");
        // 找出价格最高的
        Book maxPrice = books[0];
        for (int i = 1; i < books.length; i++) {
            if (books[i].getPrice() > maxPrice.getPrice()) {
                maxPrice = books[i];
            }
        }
        System.out.println("价格最高的书为: " + maxPrice.getName() + ", 售价: " + maxPrice.getPrice());

        // 计算所有书的平均价格
        int sum = 0;
        for (int i = 0; i < books.length; i++) {
            sum += books[i].getPrice();
        }
        System.out.println("所有图书平均价格为: " + (sum / books.length));

        // 让用户输入一个价格, 输出所有高于该价格的书籍
        System.out.println("请输入一个价格: ");
        Scanner sc = new Scanner(System.in);
        int price = sc.nextInt();

        // 假设没有高于输入的价格的书本
        int count = 0;

        for (int i = 0; i < books.length; i++) {
            if (books[i].getPrice() > price) {
                if (++count == 1) { // 在打印第一本之前打印一个标题信息
                    System.out.println("售价高于" + price + "的图书信息如下: ");
                }
                books[i].show();
            }
        }

        if (count == 0) {
            System.out.println("暂无图书售价高于" + price);
        }
    }
}

class Book {
    private String name;
    private String author;
    private int price;

    public void show() {
        System.out.println(name + "..." + author + "..." + price);
    }

    public Book(String name, String author, int price) {
        this.name = name;
        this.author = author;
    }
}

```

```
        this.price = price;
    }

    public Book() {
    }

    public String getName() {
        return name;
    }

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

    public String getAuthor() {
        return author;
    }

    public void setAuthor(String author) {
        this.author = author;
    }

    public int getPrice() {
        return price;
    }

    public void setPrice(int price) {
        this.price = price;
    }
}
```

练习3

编写程序，输入一个0-100的成绩：

- 使用if-else转换为等级 (A:[90,100], B:[80-89],C:[70,79],D:[60,69],E:[0,59])
- 使用集合存储各等级人数统计
- 输出等级分布情况

效果如下：

请输入第1个人的成绩: [0, 100]

666

成绩输入有误, 请重新输入!

请输入第1个人的成绩: [0, 100]

66

请输入第2个人的成绩: [0, 100]

67

请输入第3个人的成绩: [0, 100]

88

请输入第4个人的成绩: [0, 100]

87

D=2

B=2

```
public class Test3 {
    public static void main(String[] args) {
        ArrayList<String> list = new ArrayList<>(); // ["A=1","B=0"]
        Scanner sc = new Scanner(System.in);

        for (int i = 1; i <= 4; i++) {
            System.out.println("请输入第"+i+"个人的成绩: [0,100]");
            int score = sc.nextInt();
            String grade = "";

            // 判断成绩等级
            if (score >= 90 && score <= 100){
                grade = "A";
            }else if (score >= 80 && score < 90){
                grade = "B";
            }else if (score >= 70 && score < 80){
                grade = "C";
            }else if (score >= 60 && score < 70){
                grade = "D";
            }else if (score >= 0 && score < 60){
                grade = "E";
            }else{
                System.out.println("成绩输入有误, 请重新输入!");
                i--;
                continue;
            }

            // 判断本次的成绩等级在集合中是否存在
            int count = isExists(grade, list);
            count++;
            if (count == 1) {
                list.add(grade + "=1");
            }else{
```

```

        // 将原本记录的移除，记录新的次数
        removeGrade(grade,list);
        list.add(grade+"="+count);
    }
}

for (int i = 0; i < list.size(); i++) {
    System.out.println(list.get(i));
}

}

public static void removeGrade(String grade,ArrayList<String> list){
    for (int i = 0; i < list.size(); i++) {
        String str = list.get(i);
        String[] split = str.split("=");
        if (grade.equals(split[0])){
            list.remove(i);
        }
    }
}

// 判断指定成绩在集合中是否存在，并返回对应次数，不存在则返回0
public static int isExists(String grade, ArrayList<String> list){
    int res = 0 ;        // 假设不存在

    for (int i = 0; i < list.size(); i++) {
        String str = list.get(i);
        String[] split = str.split("=");
        if (grade.equals(split[0])){
            res = Integer.parseInt(split[1]);
            break;
        }
    }

    return res;
}
}

```

练习4

抽奖系统

创建一个包含5个奖项的集合（如["一等奖","二等奖",..., "谢谢参与"]）：

- 实现随机抽奖功能，抽完为止
- 记录已抽中的奖项并避免重复

效果如下：

本次抽到的是：一等奖
本次抽到的是：二等奖
本次抽到的是：谢谢参与
本次抽到的是：四等奖
本次抽到的是：三等奖

```
public class Test4 {  
    public static void main(String[] args) {  
        ArrayList<String> list = new ArrayList<>();  
        list.add("一等奖");  
        list.add("二等奖");  
        list.add("三等奖");  
        list.add("四等奖");  
        list.add("谢谢参与");  
  
        Random r = new Random();  
        while (list.size() != 0) {  
            int index = r.nextInt(list.size());    //[0,list.size()-1]  
            String res = list.get(index);  
            System.out.println("本次抽到的是：" + res);  
  
            //将本次抽到的奖项移除  
            list.remove(index);  
        }  
    }  
}
```

练习5

字符串处理

编写程序处理用户输入的字符串：

- 统计每个单词出现的次数
- 找出最长的单词

请输入5个单词，每个单词间用空格隔开：

hello springboot java java world

统计结果如下：

hello=1

springboot=1

java=2

world=1

其中长度最长的单词为：springboot

```
public class Test5 {
    public static void main(String[] args) {
        System.out.println("请输入5个单词，每个单词间用空格隔开：");
        Scanner sc = new Scanner(System.in);
        String str = sc.nextLine();

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

        // 定义一个集合用来统计每个单词次数
        ArrayList<String> list = new ArrayList<>(); //{ "hello=1" }

        for (int i = 0; i < split.length; i++) {
            String word = split[i];
            int count = isExists(word, list);
            count++;
            if (count == 1) {
                list.add(word + "=1");
            } else {
                // 将原本记录的移除，记录新的次数
                removeGrade(word, list);
                list.add(word + "=" + count);
            }
        }

        // 找出最长单词
        String max = split[0];
        for (int i = 1; i < split.length; i++) {
            if (split[i].length() > max.length()) {
                max = split[i];
            }
        }

        System.out.println("统计结果如下：");

        for (int i = 0; i < list.size(); i++) {
            System.out.println(list.get(i));
        }
        System.out.println("其中长度最长的单词为： " + max);
    }
}
```

```

    }

    public static void removeGrade(String word, ArrayList<String> list){
        for (int i = 0; i < list.size(); i++) {
            String str = list.get(i);
            String[] split = str.split("=");
            if (word.equals(split[0])){
                list.remove(i);
            }
        }
    }
}

// 判断指定成绩在集合中是否存在，并返回对应次数，不存在则返回0
public static int isExists(String word, ArrayList<String> list){
    int res = 0 ;        // 假设不存在

    for (int i = 0; i < list.size(); i++) {
        String str = list.get(i);
        String[] split = str.split("=");
        if (word.equals(split[0])){
            res = Integer.parseInt(split[1]);
            break;
        }
    }

    return res;
}
}

```

练习6

密码验证器

设计一个方法完成密码验证，要求密码：

- 长度8-16位
- 包含大小写字母和数字

效果如图：

请输入一个密码：

A1234bcd

A1234bcd, 该密码是否验证通过: true

```

public class Test6 {
    public static void main(String[] args) {
        System.out.println("请输入一个密码: ");
    }
}

```

```

Scanner sc = new Scanner(System.in);
String password = sc.next();

boolean res = verifyPassword(password);
System.out.println(password+" 该密码是否验证通过: " + res);
}

private static boolean verifyPassword(String password) {
    if (password.length() < 8 || password.length() > 16){
        return false;
    }

    // 包含大写字母和数字
    boolean containsUpperCase = false;
    boolean containsLowerCase = false;
    boolean containsNumber = false;

    for (int i = 0; i < password.length(); i++) {
        char ch = password.charAt(i);
        if (ch >= '0' && ch <= '9'){
            containsNumber = true;
        }else if (ch >= 'A' && ch <= 'Z'){
            containsUpperCase = true;
        }else if (ch >= 'a' && ch <= 'z'){
            containsLowerCase = true;
        }
    }

    if (!containsUpperCase || !containsNumber || !containsLowerCase){
        return false;
    }

    return true;
}
}

```

练习7

2016年里约热内卢奥运会足球比赛，共有16只球队参赛，分别是 巴西、阿根廷、丹麦、德国、葡萄牙、瑞典、斐济、洪都拉斯、墨西哥、阿尔及利亚、尼日利亚、南非、伊拉克、日本、韩国、哥伦比亚。请使用所学集合知识，编写程序把这16支球队随机分成四组，输出结果如下：

第1组： 哥伦比亚 韩国 伊拉克 葡萄牙 第2组： 阿尔及利亚 南非 斐济 尼日利亚 第3组： 阿根廷 墨西哥 日本 洪都拉斯 第4组： 瑞典 丹麦 德国 巴西

练习8

模拟搜索功能：

- 使用ArrayList存储多个字符串
- 实现关键词搜索（返回包含关键词的字符串）

