实验4 数组、字符串与常用类

班　级:　　软工1805
学　号:　　201806061219
姓　名:　　王程飞
完成日期:　2019.11.4

1. 编写一个类ArrayUtil，包括以下方法：

> (1)求一个数组的最大元素值：public static int maxElement(int a[])
> (2)求一个数组的所有元素的平均值： public static int average(int a[])
> (3) 查找某个数在数组中的位置：public static int search(int a[],int k)，要求从前往后查，返回在数组中第一次出现位置，如果未查到，则返回-1。
> (4)编写main方法，随机产生20个60～100之间的不重复整数赋值给一个数组，分别测试以上方法。

代码

```
class ArrayUtil {
    private static int maxElement(int[] a) {
        int max = a[0];
        for(int e: a) {
            if (e > max) {
                max = e;
            }
        }
        return max;
    }

    private static int average(int[] a) {
        int sum = 0;

        for (int e: a) {
            sum += e;
        }
        return sum/a.length;
    }

    private static int search(int[] a, int k) {
        for (int i = 0; i < a.length; i++) {
            if (a[i] == k) {
                return i;
            }
        }
        return -1;
    }

    public static void test0() {
        int[] num = new int[20];
        Random random = new Random();
        HashSet<Integer> set = new HashSet<>();

        do {
            set.add(random.nextInt(40) + 60);
        } while (set.size() < num.length);
```

```
        int i = 0;
        for(int e: set) {
            num[i++] = e;
            System.out.println(e);
        }
        System.out.println(Arrays.toString(num));
        System.out.println("max: " + ArrayUtil.maxElement(num));
        System.out.println("avg: " + ArrayUtil.average(num));
        System.out.println("find 70: " + ArrayUtil.search(num, 70));
    }
}
```

结果

```
[64, 65, 97, 98, 67, 68, 69, 73, 75, 79,
    84, 85, 86, 88, 89, 91, 92, 61, 62, 95]
max: 98
avg: 79
find 70: -1
```

2. 设有3个数组定义如下:

```
String[] x ={"zero", "one", "two", "three", "four", "five",
    "six", "seven", "eight", "nine"} ;
String[] y ={"ten", "eleven", "twelve", "thirteen", "fourteen",
    "fifteen", "sixteen", "seventeen", "eighteen", "nineteen"};
String[] z ={"twenty", "thirty", "forty", "fifty", "sixty",
    "seventy", "eighty", "ninety"};
```

> 试编写一个翻译程序Translate.java，实现用Java命令行输入一个小于100的整数，将其翻成英文表示，输入英文则输出相应的整数。
> 例如：
> 输入32，输出：thirty two
> 输入8， 输出：eight
> 输入fourteen，输出14

代码

```
class Translate {
    private static String[] x ={"zero", "one", "two", "three",
        "four", "five", "six", "seven", "eight", "nine"} ;
    private static String[] y ={"ten", "eleven", "twelve",
        "thirteen", "fourteen", "fifteen", "sixteen",
        "seventeen", "eighteen", "nineteen"};
    private static String[] z ={"twenty", "thirty", "forty",
        "fifty", "sixty", "seventy", "eighty", "ninety"};
    static void test2() {
        ArrayList<String> ax = new ArrayList<String>(Arrays.asList(x));
        ArrayList<String> ay = new ArrayList<String>(Arrays.asList(y));
        ArrayList<String> az = new ArrayList<String>(Arrays.asList(z));
        ArrayList<String> at = new ArrayList<>();
        at.addAll(ax);
        at.addAll(ay);
        Scanner scanner = new Scanner(System.in);
        System.out.println("输入: ");
```

```
        while (scanner.hasNextLine()) {
            String s = scanner.nextLine();

            try {
                int num = Integer.parseInt(s);

                if (num >= 20 && num < 100) {
                    System.out.print(z[(num/10) -2]);
                    if (num%10 != 0) {
                        System.out.println(" "+x[num-(num/10)*10]);
                    } else {
                        System.out.println();
                    }
                } else if (num >= 0 && num < 100) {
                    System.out.println(at.get(num));
                } else {
                    System.out.println("...");
                }
            } catch (NumberFormatException e) {
                String[] nums = s.split(" ");
                int sum = 0;
                for (String item : nums) {
                    if (ax.contains(item)) {
                        sum += ax.indexOf(item);
                    } else if (ay.contains(item)) {
                        sum += ay.indexOf(item) + 10;
                    } else if (az.contains(item)) {
                        sum += (az.indexOf(item) + 2) * 10;
                    }
                }
                System.out.println(sum);
            }
        }
    }
}
```

3. 编写一个程序，包含如下方法：

> (1)输出当前的年份
> (2)输出当前的季度
> (3)输出当前的月份
> (4)输出当前是星期几
> (5)输出当前的日期，格式为××××年××月××日，如2013年11月1日
> (6)输出当前的北京时间，格式为时：分：秒，如08:21:36
> (7)输出当前的日期和时间，格式为××××-××-×× 时：分：秒，如 2013-11-1 08:21:36

代码

```
class DateOutput {
    private static Calendar calendar = Calendar.getInstance();

    static void test3() {
        int year = calendar.get(Calendar.YEAR);
        System.out.println(year + "年");
        int month = calendar.get(Calendar.MONTH) + 1;
        System.out.println(month + "月");
        String season;
```

```
        switch (month) {
            case 1:
            case 2:
            case 3:
                season = "春";
                break;
            case 4:
            case 5:
            case 6:
                season = "夏";
                break;
            case 7:
            case 8:
            case 9:
                season = "秋";
                break;
            case 10:
            case 11:
            case 12:
                season = "冬";
                break;
            default:
                season = "?";
        }
        System.out.println(season);
        int week = calendar.get(Calendar.DAY_OF_WEEK);
        switch (week) {
            case 1:
                System.out.println("星期日");
                break;
            case 2:
                System.out.println("星期一");
                break;
            case 3:
                System.out.println("星期二");
                break;
            case 4:
                System.out.println("星期三");
                break;
            case 5:
                System.out.println("星期四");
                break;
            case 6:
                System.out.println("星期五");
                break;
            case 7:
                System.out.println("星期六");
                break;
            default:
                System.out.println("?");
                break;
        }
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy年MM月dd日");
        System.out.println(sdf.format(calendar.getTime()));
        sdf = new SimpleDateFormat("HH:mm:ss");
        System.out.println(sdf.format(calendar.getTime()));
        sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
        System.out.println(sdf.format(calendar.getTime()));
    }
}
```

结果

```
2019年
11月
冬
星期一
2019年11月04日
13:30:20
2019-11-04 13:30:20
```

4. 假设某停车场的收费标准为：15分钟以内免费，超过15分钟每小时收费5元，每天30元封顶。有效停车时间=实际停车时间-15分钟，如实际停车1小时10分，收费5元；实际停车时间1小时20分钟，收费10元，以此类推。请编写一个简单的停车收费程序ParkingFeeCollection1.java。

代码

```
class ParkingFeeCollection1 {
    static void test4() {
        String[] _in = new String[]{
                "2014-10-08 12:02:13",
                "2014-10-08 13:12:15",
                "2014-10-08 14:52:17",
                "2014-10-08 15:12:15",
                "2014-10-08 16:12:15",
                "2014-10-08 20:12:15",
                "2014-10-08 16:12:15",
                "2014-10-08 17:12:15",
        };

        String[] _out = new String[]{
                "2014-10-08 12:13:56",
                "2014-10-08 13:48:42",
                "2014-10-08 16:28:22",
                "2014-10-08 20:38:49",
                "2014-10-09 07:29:52",
                "2014-10-09 07:45:26",
                "2014-10-09 13:49:53",
                "2014-10-11 15:12:12"
        };
        SimpleDateFormat sdf =
            new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
        for (int i = 0; i< _in.length; i++) {
            try {
                Calendar in = Calendar.getInstance();
                Calendar out = Calendar.getInstance();
                System.out.println("停车时间: " + _in[i]);
                in.setTime(sdf.parse(_in[i]));
                System.out.println("离开时间: " + _out[i]);
                out.setTime(sdf.parse(_out[i]));
                long diff = out.getTimeInMillis() - in.getTimeInMillis();
                if (diff < 0) {
                    System.out.println("输入时间有误");
                    return;
                }
                int sum = 0;
                int diffDay = out.get(Calendar.DAY_OF_YEAR)
                    - in.get(Calendar.DAY_OF_YEAR);
                if (diffDay == 0) {
```

```
                    int diffMinuit = (int) (diff/60000 - 15);
                    if (diffMinuit < 0) {
                        sum = 0;
                    } else {
                        sum = ((diffMinuit)/30)*5;
                    }
                    if (sum > 30) {
                        sum = 30;
                    }
                    System.out.println(sum);
                } else {
                    sum += (diffDay -1)*30;
                    Calendar c0 = Calendar.getInstance();
                    c0.setTime(in.getTime());
                    c0.set(Calendar.HOUR_OF_DAY, 23);
                    c0.set(Calendar.MINUTE, 59);
                    Calendar c1 = Calendar.getInstance();
                    c1.setTime(out.getTime());
                    c1.set(Calendar.HOUR_OF_DAY, 0);
                    c1.set(Calendar.MINUTE, 0);
                    int diff0 = (int) (c0.getTimeInMillis()
                        - in.getTimeInMillis())/60000;
                    int diff1 = (int) (out.getTimeInMillis()
                        - c1.getTimeInMillis())/60000;
                    if(diff0>15){
                        int s0 = ((diff0-15)/30)*5;
                        if (s0 > 30) s0 = 30;
                        sum += s0;
                    }
                    int s1 = (diff1/30)*5;
                    if (s1 > 20) {
                        s1 = 30;
                    }
                    sum += s1;

                    System.out.println(sum);
                }

            } catch (ParseException e) {
                System.out.println("输入时间有误");
            }
        }

    }
}
```

结果

```
停车时间: 2014-10-08 12:02:13
离开时间: 2014-10-08 12:13:56
0
停车时间: 2014-10-08 13:12:15
离开时间: 2014-10-08 13:48:42
0
停车时间: 2014-10-08 14:52:17
离开时间: 2014-10-08 16:28:22
10
停车时间: 2014-10-08 15:12:15
离开时间: 2014-10-08 20:38:49
```

30
停车时间: 2014-10-08 16:12:15
离开时间: 2014-10-09 07:29:52
60
停车时间: 2014-10-08 20:12:15
离开时间: 2014-10-09 07:45:26
60
停车时间: 2014-10-08 16:12:15
离开时间: 2014-10-09 13:49:53
60
停车时间: 2014-10-08 17:12:15
离开时间: 2014-10-11 15:12:12
120