

验证QQ号

2018年7月12日 15:14

```
import java.util.Scanner;

/*
 *   校验qq号码:
 *       要求必须为5-15位数字
 *       0不能开头
 */
public class RegexDemo {
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        System.out.println("请输入QQ号码");
        String qq=sc.nextLine();
        System.out.println(checkQQ(qq));
    }

    public static boolean checkQQ(String qq) {
        boolean flag = true;
        // 校验长度
        if (qq.length() >= 5 && qq.length() <= 15) {
            // 0不能开头
            if (!qq.startsWith("0")) {
                char[] chs = qq.toCharArray();
                for (int x = 0; x < chs.length; x++) {
                    char ch = chs[x];
                    if (!Character.isDigit(ch)) {
                        flag = false;
                        break;
                    }
                }
            } else {
                flag = false;
            }
        } else {
            flag = false;
        }
        return flag;
    }
}
```

}
}

正则表达式

2018年7月1日 14:58

```
import java.util.Scanner;

/*
 * 正则表达式:符合一定规则的字符串
 */
public class RegexDemo2 {
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        System.out.println("请输入QQ号码");
        String qq=sc.nextLine();
        System.out.println(checkQQ(qq));
    }

    public static boolean checkQQ(String qq) {
        /*String regex= "[1-9][0-9]{4,14}";
        //public boolean matches(String regex):告知次字符串是否匹配正则表达式
        boolean flag=qq.matches(regex);
        return flag;*/

        return qq.matches("[1-9][0-9]{4,14}");
    }
}
```

正则表达式的组成规则

2018年7月12日 15:21

正则表达式 Pattern

1. 字符 x 字符 x 例: 'a' 表示字符 a

\ 反斜线字符

\n 换行符

\t 回车符

2. 字符类 [abc] 表示 a、b 或 c

[^abc] 表示除 a、b、c 以外的其它字符

[a-zA-Z] a~z A~Z 所有字符

[0-9] 0~9 的所有字符

3. 预定义字符类

. 表示任何字符 \. 表示 . 本身

\d 数字 0~9 [0-9] \Rightarrow \d

\w 单词字符 [a-zA-Z_0-9]

在正则表达式中组成的单词必须由此组成

4. 边界匹配器

^ 行的开头

\$ 行的结尾

\b 单词边界: 就是不是单词字符的地方

举例: hello**o**world**o** haha**o**wawa

5. Greedy 数量词

$x?$ 一次或一次也没有 没有给定次数, 默认一次

x^* 零或多次

x^+ 一次或多次

$x\{n\}$ 恰好 n 次

$x\{n, \}$ 至少 n 次

$x\{n, m\}$ 至少 n 次, 但不超过 m 次

判断功能

2018年7月12日 15:46

```
import java.util.Scanner;

/*
 * 判断功能 matches
 * 判断手机号码是否符合规则
 */
public class RegexDemo {
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        System.out.println("请输入手机号码");
        String phone=sc.nextLine();

        String regex="1[38]\\d{9}";
        System.out.println(phone.matches(regex));
    }
}
```

校验邮箱

2018年7月12日 15:55

```
public class RegexTest {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        System.out.println("请输入邮箱");  
        String email = sc.nextLine();  
  
        // String regex = "[a-zA-Z_0-9]+@[a-zA-Z_0-9]{2,6}(\\.[a-zA-Z_0-9]{2,3})+";  
        String regex = "\\w+@\\w{2,6}(\\.\\w{2,3})+";  
        boolean flag = email.matches(regex);  
        System.out.println(flag);  
    }  
}
```

分割功能

2018年7月12日 16:12

```
/*
 *      分割功能
 *      String类的public String[] split(String regex)
 *      根据正则表达式的匹配拆分次字符串
 *      举例
 *      搜索好友
 *      性别：女
 *      范围：18-24
 */
public class RegexDemo {
    public static void main(String[] args) {
        String ages="18-24";

        String regex="-";

        String[] strArray=ages.split(regex);

        int startage=Integer.parseInt(strArray[0]);
        int endtage=Integer.parseInt(strArray[1]);
    }
}
```

分割练习

2018年7月12日 16:22

```
public class RegexTest {
    public static void main(String[] args) {
        String s1="aa,bb,cc";
        String[] str1Array=s1.split(",");
        for(int x=0;x<str1Array.length;x++) {
            System.out.println(str1Array[x]);
        }
        System.out.println("-----");

        String s2="aa.bb.cc";
        String[] str2Array=s1.split("\\.");
        for(int x=0;x<str2Array.length;x++) {
            System.out.println(str2Array[x]);
        }
        System.out.println("-----");

        String s3="aa bb      cc";
        String[] str3Array=s1.split(" +");
        for(int x=0;x<str3Array.length;x++) {
            System.out.println(str3Array[x]);
        }
        System.out.println("-----");

        //硬盘上的路径用\\代替\
        String s4="C:\\Users\\TJtulong\\Desktop\\java\\day14";
        String[] str4Array=s4.split("\\\\");
        for(int x=0;x<str4Array.length;x++) {
            System.out.println(str4Array[x]);
        }
        System.out.println("-----");
    }
}
```

```
/*
 * 将字符串"91 27 46 38 50"转为数组
 */
public class RegexTest2 {
    public static void main(String[] args) {
        String s = "91 27 46 38 50";
        String[] strArray = s.split(" ");
        int[] arr = new int[strArray.length];
        for (int x = 0; x < strArray.length; x++) {
            arr[x] = Integer.parseInt(strArray[x]);
        }
        Arrays.sort(arr);
        StringBuffer sb = new StringBuffer();
        for (int x = 0; x < arr.length; x++) {
            sb.append(arr[x]).append(" ");
        }
        String result = sb.toString().trim();
        System.out.println(result);
    }
}
```


替换功能

2018年7月12日 16:42

```
/*
 * 替换功能:
 *      public String replaceAll(String regex,String replacement)
 *      replaceAll替换所有符合正则表达式的子字符串
 */
public class RegexDemo {
    public static void main(String[] args) {
        String s = "hello11531534941kh239943892java";

        // 去除所有数字
        String regex = "\\d";
        String ss = "*";
        String result = s.replaceAll(regex, ss);
        System.out.println(result);
    }
}
```

匹配功能

2018年7月12日 16:53

```
import java.util.regex.Matcher;
import java.util.regex.Pattern;

/*
 * 获取字符串中由三个字符组成的单词
 * da jia ting wo shuo,jin tian yao xia yu,bu shang wan zi xi,gao xing bu
 */
public class RegexDemo2 {
    public static void main(String[] args) {
        String s = "da jia ting wo shuo,jin tian yao xia yu,bu shang wan zi xi,gao xing bu";
        // 规则
        String regex = "\\b\\w{3}\\b";

        Pattern p = Pattern.compile(regex);
        Matcher m = p.matcher(s);
        // 调用匹配器的功能，通过find查找有没有满足条件的字符串
        // public boolean find()
        boolean flag = m.find();
        System.out.println(flag);

        // public String group
        String ss = m.group();
        System.out.println(ss);
        // 注意：先find再group
        while (m.find()) {
            System.out.println(m.group());
        }
    }
}
```

Math类

2018年7月12日 20:48

```
/*
 * Math:用于数学运算的类
 * 成员变量:
 *      public static final double PI
 *      public static final double PI
 * 成员方法:
 *      public static int abs(int a):绝对值
 *      public static double ceil(double a):向上取整
 *      public static double floor(double a):向下取整
 *      public static int max(int a,int b)
 *      public static double pow(double a,double b):a的b次幂
 *      public static double random();随机数[0.0,1.0)
 *      public static int round(float a):四舍五入
 *      public static double sqrt(double a):算术平方根
 */
public class MathDemo {
    public static void main(String[] args) {
        System.out.println("pi=" + Math.PI);
        System.out.println("e=" + Math.E);

        System.out.println("abs=" + Math.abs(-10));

        System.out.println("ceil=" + Math.ceil(12.34));
        System.out.println("ceil=" + Math.ceil(12.78));

        System.out.println("floor=" + Math.floor(12.34));
        System.out.println("floor=" + Math.floor(12.78));

        System.out.println("max=" + Math.max(12, 23));
        System.out.println("max=" + Math.max(Math.max(12, 23), 44));

        System.out.println("pow=" + Math.pow(2, 3));

        System.out.println("random=" + Math.random());
        System.out.println("random=" + ((int) (100 * Math.random()) + 1));
    }
}
```

```
System.out.println("round=" + Math.round(12.34f));// float类型
```

```
System.out.println("round=" + Math.round(12.56f));
```

```
System.out.println("sqrt=" + Math.sqrt(6));
```

```
}
```

```
}
```

面试题

2018年7月12日 21:13

```
import java.util.Scanner;

/*
 * 设计一个方法：可以实现获取任意范围内的随机数
 */
public class MathTest {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("请输入开始数");
        int start = sc.nextInt();
        System.out.println("请输入结束数");
        int end = sc.nextInt();
        int num = getRandom(start, end);
        System.out.println(num);
    }

    public static int getRandom(int start, int end) {
        int number = (int) (Math.random() * (end - start + 1)) + start;
        return number;
    }
}
```

Random类

2018年7月12日 21:23

```
import java.util.Random;

/*
 * Random:产生随机数的类
 * 构造方法:
 *      public Random():没有给种子, 用默认的种子, 是当前时间的毫秒值
 *      public Random(long seed):给出指定的种子
 *      给定种子后, 每次得到的随机数是相同的
 * 成员方法:
 *      public int nextInt():返回的是int范围内的随机数
 *      public int nextInt(int n):返回的是[0,n]范围内的随机数
 *
 */
public class RandomDemo {
    public static void main(String[] args) {
        // Random r = new Random();
        Random r = new Random(111);
        for (int x = 0; x < 10; x++) {
            int num = r.nextInt(100) + 1;
            System.out.println(num);
        }
    }
}
```

System类中的gc ()方法

2018年7月12日 21:40

```
/*
 * System类包含一些有用的类字段和方法，它不能被实例化
 * 方法：
 *      public static void gc()
 */
public class SystemDemo {
    public static void main(String[] args) {
        Person p = new Person("xiao",22);
        System.out.println(p);

        p=null;
        System.gc();//当前对象被回收了Person [name=xiao, age=22]
    }
}
```

System类方法

2018年7月12日 22:03

```
/*
 * public static void exit(int status)
 *      终止当前正在运行的java虚拟机, 参数用作状态码, 根据惯例, 非0的状态码
表示异常终止
 * public static long currentTimeMillis()
 *      返回以毫秒为单位的当前时间
 */
public class SystemDemo2 {
    public static void main(String[] args) {
        System.out.println("hhhhh");
        // System.exit(0);
        System.out.println("bainiangzi");
        // System.out.println(System.currentTimeMillis());
        long start = System.currentTimeMillis();
        //可以用来测试运行时间
        for (int x = 0; x < 100000; x++) {
            System.out.println("hello" + x);
        }
        long end = System.currentTimeMillis();
        System.out.println("用时" + (end - start) + "毫秒");
    }
}
```


arraycopy

2018年7月12日 22:25

```
import java.util.Arrays;
```

```
/*
```

```
 * public static void arraycopy(Object src,int srcPos,Object dest,int  
destPos,intLength)
```

```
 *          从指定原数组中复制一个数组，复制从指定的位置开始，到目标数组的指定位置结束
```

```
 */
```

```
public class SystemDemo3 {
```

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

```
        int[] arr = { 11, 22, 33, 44, 55 };
```

```
        int[] arr2 = { 6, 7, 8, 9, 10 };
```

```
        System.arraycopy(arr, 1, arr2, 2, 2);
```

```
        System.out.println(Arrays.toString(arr));
```

```
        System.out.println(Arrays.toString(arr2));// [6, 7, 22, 33, 10]
```

```
    }
```

```
}
```

BigInteger类

2018年7月12日 22:36

```
import java.math.BigInteger;

/*
 * BigInteger:可以让超过Integer范围内的数据进行运算
 * BigInteger(String val)
 * 方法:
 *     add
 *     subtract
 *     multiply
 *     divide
 *     public BigInteger[] divideAndRemainder(BigInteger val):返回商和余数的数组
 */
public class BigIntegerDemo {
    public static void main(String[] args) {
        Integer i = new Integer("2147483647");
        System.out.println(i);

        BigInteger bi = new BigInteger("2147483648");
        System.out.println("bi=" + bi);

        // 方法练习
        BigInteger bi1 = new BigInteger("100");
        BigInteger bi2 = new BigInteger("50");
        System.out.println("add=" + bi1.add(bi2));
        System.out.println("subtract=" + bi1.subtract(bi2));
        System.out.println("multiply=" + bi1.multiply(bi2));
        System.out.println("divide=" + bi1.divide(bi2));
        System.out.println(bi1.divideAndRemainder(bi2)[0]);
    }
}
```

BigDecimal类

2018年7月12日 23:27

```
/*
 * float和double类型很容易丢失精度
 * 为了提高精度，计算浮点数，java提供了BigDecimal类
 * BigDecimal:不可变的、任意精度的有符号十进制数
 */
public class BigDecimalDemo {
    public static void main(String[] args) {
        System.out.println(0.09 + 0.01); // 0.09999999999999999
        System.out.println(1.0 - 0.32); // 0.6799999999999999
        System.out.println(1.015 * 100); // 101.49999999999999
        System.out.println(1.301 / 100); // 0.013009999999999999

        System.out.println(1 - 0.12); // 0.88
    }
}
```

BigDecimal类的方法

2018年7月12日 23:35

```
import java.math.BigDecimal;

/*
 * 构造方法:
 *      public BigDecimal(String val)
 */
public class BigDecimalTest {
    public static void main(String[] args) {
        BigDecimal bd1 = new BigDecimal("0.09");
        BigDecimal bd2 = new BigDecimal("0.01");
        System.out.println("add=" + bd1.add(bd2));

        BigDecimal bd3 = new BigDecimal("1.0");
        BigDecimal bd4 = new BigDecimal("0.32");
        System.out.println("subtract=" + bd3.subtract(bd4));

        BigDecimal bd5 = new BigDecimal("1.015");
        BigDecimal bd6 = new BigDecimal("100");
        System.out.println("multiply=" + bd5.multiply(bd6));

        // public BigDecimal divide(BigDecimal divisor,int scale,int roundingMode)
        // 商, 几位小数, 如何舍取
        BigDecimal bd7 = new BigDecimal("1.301");
        BigDecimal bd8 = new BigDecimal("100");
        System.out.println("divide=" + bd7.divide(bd8));
        System.out.println("divide=" + bd7.divide(bd8, 3,
            BigDecimal.ROUND_HALF_UP));
    }
}
```

Date类

2018年7月13日 10:58

```
import java.util.Date;

/*
 * Date()
 * 表示特定的瞬间，精确到毫秒
 * 构造方法：
 *     Date():根据当前的默认毫秒值创建日期对象
 *     Date(long date):根据给定的毫秒值创建日期对象
 *     注意要考虑时差问题
 */
public class DateDemo {
    public static void main(String[] args) {
        Date d=new Date();
        System.out.println("d="+d);

        long time=System.currentTimeMillis();
        Date d2 = new Date(time);
        System.out.println("d2="+d);
    }
}
```

Date类的方法

2018年7月13日 11:11

```
import java.util.Date;

/*
 * public long getTime():获取时间，以毫秒为单位
 * public long setTime():设置时间
 *      从Date得到毫秒值：getTime()
 *      把毫秒值转换为Date
 *      构造方法
 *      setTime(long time)
 *
 */
public class DateDemo {
    public static void main(String[] args) {
        Date d = new Date();
        long time = d.getTime();
        System.out.println(time);

        System.out.println("d=" + d);
        d.setTime(1000);
        System.out.println("d=" + d);
    }
}
```

DateFormat类

2018年7月13日 11:48

```
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;

/*
 * 日期操作:
 *      Date---String(格式化)
 *      public final String format(Date date)
 *      String---Date(解析)
 *      public final Date Parse(String source)
 *
 * DateFormat:可以进行日期和字符串的格式化和解析,但因为是抽象类,所以使用具体子类
SimpleDateFormat
 * SimpleDateFormat的构造方法
 *      SimpleDateFormat()
 *      SimpleDateFormat(String pattern):给定的模式
 *          年: y
 *          月: M
 *          日: d
 *          时: H
 *          分: m
 *          秒: s
 *          2012年12月12日 12: 12: 12
 *
 */
public class DateFormatDemo {
    public static void main(String[] args) throws ParseException {
        //Date---String(格式化)
        Date d = new Date();
        //默认模式
        //SimpleDateFormat sdf = new SimpleDateFormat();
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy年MM月dd日
HH:mm:ss");
        String s = sdf.format(d);
        System.out.println(s);
    }
}
```

```
//String---Date(解析)
String str="2008-08-08 12:12:12";
//格式必须匹配
SimpleDateFormat sdf2 = new SimpleDateFormat("yyyy-MM-dd
HH:mm:ss");
Date dd = sdf2.parse(str);
System.out.println(dd);
}
}
```


DateUtil工具类

2018年7月13日 11:49

```
import java.text.ParseException;
import java.util.Date;

public class DateUtilDemo {
    public static void main(String[] args) throws ParseException {
        Date d=new Date();
        String s=DateUtil.dateToString(d, "yyyy-MM-dd HH:mm:ss");
        System.out.println(s);

        String str="2014-10-4";
        Date dd =DateUtil.StringToDate(str, "yyyy-MM-dd");
        System.out.println(dd);
    }
}
```

练习

2018年7月13日 14:43

```
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Scanner;

import org.omg.Messaging.SyncScopeHelper;

//算一下你开到世界多少天
public class DaysTest {
    public static void main(String[] args) throws ParseException {
        Scanner sc = new Scanner(System.in);
        System.out.println("请输入你的出生年月日");
        String line = sc.nextLine();

        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
        Date d = sdf.parse(line);

        long myTime = d.getTime();
        long nowTime = System.currentTimeMillis();

        long time = nowTime - myTime;
        long day = time / 1000 / 60 / 60 / 24;
        System.out.println("你来到世界" + day + "天");
    }
}
```

Calendar类

2018年7月13日 14:53

```
import java.util.Calendar;

/*
 * Calendar类：操作日历字段
 *      public int get(int field):返回给定日历字段的值
 *      日历类中的每个日历字段都是静态的成员变量，并且是int类型
 */
public class CalenderDemo {
    public static void main(String[] args) {
        Calendar rightNow = Calendar.getInstance();// 子类对象，多态

        int year = rightNow.get(Calendar.YEAR);
        int month = rightNow.get(Calendar.MONTH);
        int date = rightNow.get(Calendar.DATE);
        System.out.println(year + "-" + (month + 1) + "-" + date);
    }
}
```

Calendar类方法

2018年7月13日 15:08

```
import java.util.Calendar;

/*
 * Calendar类方法
 *      public void add(int field,int amount):根据给定的日历字段和对应时间。对
当前日历操作
 *      public final void set(int year,int month,int date):设定当前日历时间
 */
public class CalendarTest {
    public static void main(String[] args) {
        Calendar c = Calendar.getInstance();
        int year = c.get(Calendar.YEAR);
        int month = c.get(Calendar.MONTH);
        int date = c.get(Calendar.DATE);
        System.out.println(year + "-" + (month + 1) + "-" + date);

        //三年前的今天
        c.add(Calendar.YEAR,-3);
        year = c.get(Calendar.YEAR);
        month = c.get(Calendar.MONTH);
        date = c.get(Calendar.DATE);
        System.out.println(year + "-" + (month + 1) + "-" + date);

        c.set(2011,11,11);
        year = c.get(Calendar.YEAR);
        month = c.get(Calendar.MONTH);
        date = c.get(Calendar.DATE);
        System.out.println(year + "-" + month + "-" + date);//2011-11-11
    }
}
```

练习

2018年7月13日 15:26

```
import java.util.Calendar;
import java.util.Scanner;

/*
 * 获取任意一年二月有多少天
 */
public class CalendarTest2 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("请输入年份");
        int year = sc.nextInt();

        Calendar c = Calendar.getInstance();
        c.set(year, 2, 1);
        c.add(Calendar.DATE, -1);
        System.out.println(c.get(Calendar.DATE));
    }
}
```