【济南中心】JAVA 编程阶梯：基础篇之第十四章

正则表达式：

指一个用来描述或者匹配一系列符合某个语法规则的字符串的单个字符串。其实就是一种规则。有自己特殊的应用。

作用:比如注册邮箱,邮箱有用户名和密码,一般会对其限制长度,这个限制长度的事情就是正则表达式做的。

正则对字符串的常见功能操作：

1，匹配。 使用 String 类中的 matches 方法。结果是 boolean

2，切割。 使用 String 类中的 split 方法。结果 String[]

3，替换。 使用 String 类中的 replaceAll(regex ,string); 结果是一个新的字符串

4，获取。 其他三个功能内部最终使用的都是 Pattern 正则表达式对象

现在需要其他功能时，字符串 String 类中没有对应的方法，只能找 Pattern 对象

正则表达式的常见组成规则：字符、字符类、预定义字符类、边界匹配器、数量词

在这简单的介绍各个类型，具体的可以查找 Api 文档 Pattern 类

字符

x 字符 x。举例：'a'表示字符 a

\\ 反斜线字符。

\n 新行（换行）符（'\u000A'）

\r 回车符 ('\u000D')

字符类

[abc] a、b 或 c（简单类）

[^abc] 任何字符，除了 a、b 或 c（否定）

[a-zA-Z] a 到 z 或 A 到 Z，两头的字母包括在内（范围）

[0-9] 0 到 9 的字符都包括

预定义字符类

. 任何字符。我的就是.字符本身，怎么表示呢? \.

\d 数字：[0-9]

\w 单词字符：[a-zA-Z_0-9]

在正则表达式里面组成单词的东西必须有这些东西组成

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 world?haha;xixi

Greedy 数量词

X? X，一次或一次也没有

X* X，零次或多次

X+ X，一次或多次

X{n} X，恰好 n 次

X{n,} X，至少 n 次

X{n,m} X，至少 n 次，但是不超过 m 次

常见功能：(分别用的是谁呢?)

判断功能

String 类的 public boolean matches(String regex)

分割功能

String 类的 public String[] split(String regex)

替换功能

String 类的 public String replaceAll(String regex,String replacement)

获取功能

Pattern 和 Matcher

Pattern p = Pattern.compile("a*b");

Matcher m = p.matcher("aaaaab");

m.find():查找存不存在

m.group():获取刚才查找过的数据

```
/**
 * 对 QQ 号进行校验。 要求：5-15 位， 0 不可以开头。必须都是数字。
 *
 * @author Somnus
 *
 */
public class Demo {

	public static void main(String[] args) {

		String qq = "986512132";
		boolean b = checkQQ(qq);
```

```
                System.out.println(qq + ":" + b);
                boolean b1 = qq.matches("[1-9][0-9]{4,14}");
                System.out.println(qq + ":" + b1);
        }

        public static boolean checkQQ(String qq) {
                boolean flag = false;
                int len = qq.length();
                if (len >= 5 && len <= 15) {
                        if (!qq.startsWith("0")) {
                                try {
                                        Long.parseLo
ng(qq);
                                        flag = true;
                                } catch
(NumberFormatException e) {
                                        System.out.p
rintln("出现非法数字");
                                }
                        }
                }
                return flag;
        }
}
```

Math 类

Math 类包含用于执行基本数学运算的方法，如初等指数、对数、平方根和三角函数。

成员变量

E：比任何其他值都更接近 e（即自然对数的底数）的 double 值。

PI：比任何其他值都更接近 pi（即圆的周长与直径之比）的 double 值。

成员方法

publicstatic int abs(int a) 返回 double 值的绝对值

publicstatic double ceil(double a) 返回最小的（最接近负无穷大）double值，该值大于等于参数，并等于某个整数,向上取整

publicstatic double floor(double a) 返回最大的（最接近正无穷大）double值，该值小于等于参数，并等于某个整数，向下取整

publicstatic int max(int a,int b) 返回两个值中较大的那个

public static int min(int a,int b) 返回两个值中较小的那个

publicstatic double pow(double a,double b) 返回第一个参数的第二个参数次幂的值

publicstatic double random() 返回带正号的 double 值，该值大于等于 0.0 且小于 1.0

publicstatic int round(float a) 返回最接近参数的 int。

publicstatic double sqrt(double a) 返回正确舍入的 double 值的正平方根。

```
/**
  * Math 数序运算
  *
  * @author Somnus
  *
  */
public class Demo {
            public static void main(String[] args) {

                        int abs = Math.abs(-1);
                        double ceil = Math.ceil(-1.1);
                        double floor = Math.floor(1.1);
                        double round = Math.round(1.5);// 四舍五入。
                        System.out.println("abs=" + abs);
                        System.out.println("ceil=" + ceil);
                        System.out.println("floor=" + floor);
                        System.out.println("round=" + round);
                        System.out.println(Math.pow(10, 3));

            }
```

```
}
```

Random 类

此类用于产生随机数如果用相同的种子创建两个 Random 实例，

则对每个实例进行相同的方法调用序列，它们将生成并返回相同的数字序列。

构造方法

publicRandom() 创建一个新的随机数生成器

publicRandom(long seed) 使用单个 long 种子创建一个新的随机数生成器

成员方法

public intnextInt() 返回下一个伪随机数 ,它是此随机数生成器的序列中均匀分布的 int 值

public intnextInt(int n) 返回一个伪随机数，它是取自此随机数生成器序列的,在 0（包括）和指定值（不包括）之间均匀分布的 int 值

```
/**
  * Random 数序运算
  *
  * @author Somnus
  *
  */
public class Demo {
                public static void main(String[] args) {
                                Random r = new Random();
                                for (int x = 0; x < 10; x++) {
                                                int num = r.nextInt(10) + 1;
                                                System.out.print(num + ", ");
                                }
                }
}
```

System 类

System 类包含一些有用的类字段和方法。它不能被实例化

成员方法

publicstatic void gc() 运行垃圾回收器

publicstatic void exit(int status) 结束 JVM,参数是状态码,0 表示正常退出

publicstatic long currentTimeMillis() 获取当前的系统时间的毫秒值；从 1970 年 1 月 1 日开始

pubiicstatic void arraycopy(Object src, int srcPos, Object dest, int destPos, intlength) 从指定源数组中复制一个数组，复制从指定位置开始，到目标数组的指定位置结束。

参数：src - 源数组。

srcPos - 源数组中的起始位置。

dest - 目标数组。

destPos - 目标数据中的起始位置。

length - 要复制的数组元素的数量。

BigInteger 类

可以让超过 Integer 范围内的数据进行运算

构造方法

publicBigInteger(String val)

成员方法

publicBigInteger add(BigInteger val) 加

publicBigInteger subtract(BigInteger val) 减

publicBigInteger multiply(BigInteger val) 乘

publicBigInteger divide(BigInteger val) 除

publicBigInteger[] divideAndRemainder(BigInteger val) 返回包含(this / val)后跟(this % val)两个 BIgInteger 的(长度为 2)数组

BigDecimal 类

由于在运算的时候，float 类型和 double 很容易丢失精度，所以，为了能精确的表示、计算浮点数，Java 提供了 BigDecimal 不可变的、任意精度的有符号十进制数。

构造方法

publicBigDecimal(String val)

成员方法

publicBigDecimal add(BigDecimal augend) 加

publicBigDecimal subtract(BigDecimal subtrahend) 减

publicBigDecimal multiply(BigDecimal multiplicand) 乘

publicBigDecimal divide(BigDecimal divisor) 除法(当除不尽时，此方法会抛出异常)

public BigDecimal divide(BigDecimal divisor,int scale,int roundingMode)：除法。指定精度，指定舍入模式

Date 类

类 Date 表示特定的瞬间，精确到毫秒

构造方法

publicDate() 使用当前系统时间构造一个 Date

publicDate(long date) 使用一个毫秒值构造一个 Date

成员方法

public longgetTime() 获取毫秒值

public voidsetTime(long time) 设置毫秒值

SimpleDateFormat 类

DateFormat 是日期/时间格式化子类的抽象类，它以与语言无关的方式格式化并解析日期或时间。是抽象类，所以使用其子类

构造方法

publicSimpleDateFormat() 用默认的模式和默认语言环境的日期格式符号构造 SimpleDateFormat

publicSimpleDateFormat(String pattern) 用给定的模式和默认语言环境的日期格式符号构

成员方法

public finalString format(Date date) 将一个 Date 转换为 String

public Dateparse(String source) 将一个 String 转换为 Date;注意：当前 SimpleDateFormat 的对象的格式一定要与参数 source 表示的格式一致，否则会抛出异常

```
/**
  * Date 练习
  * @author Somnus
  *
  */
public class Demo {

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

```
                long time = System.currentTimeMillis();
                time = 12684148342181;
                Date date = new Date(time);
                System.out.println(date.toString());
                DateFormat dateFormat =
DateFormat.getDateInstance(DateFormat.FULL);
                dateFormat = DateFormat.getDateTimeInstance(DateFormat.LONG,
                                                DateFormat.LONG);

                // 调用 format 方法对日期对象进行格式化。用默认风格。
                String str_date = dateFormat.format(date);
                System.out.println(str_date);
        }
}
```

Calendar 类

Calendar 类是一个抽象类，它为特定瞬间与一组诸如 YEAR、MONTH、DAYOFMONTH、HOUR 等日历字段之间的转换提供了一些方法，并为操作日历字段（例如获得下星期的日期）提供了一些方法。

成员方法

publicstatic Calendar getInstance() 获取 GregorianCalendar 实例对象

public intget(int field) 获取某个字段的值

成员方法

public voidadd(int field,int amount)

public finalvoid set(int year,int month,int date)

识别二维码 关注黑马程序员视频库
免费获得更多 IT 资源