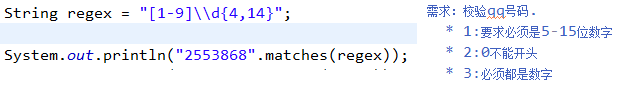
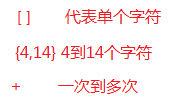
# 正则表达式

## 概念

是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。



## 分类



### 字符类



### 特殊字符转义



### Greedy 数量词

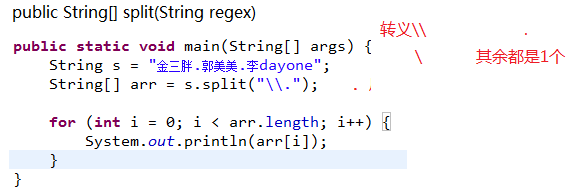


## 功能

### 验证功能



### 分割功能



**public** **class** Test1 {

/\*\*

\* 需求：我有如下一个字符串:”91 27 46 38 50”，请写代码实现最终输出结果是：”27 38 46 50 91”

\* 分析:

\* 1,将字符串切割成字符串数组

\* 2,将字符串转换成数字并将其存储在一个等长度的int数组中

\* 3,排序

\* 4,将排序后的结果遍历并拼接成一个字符串

\*/

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

String s = "91 27 46 38 50";

//1,将字符串切割成字符串数组

String regex = " ";

String[] sArr = s.split(regex);

//2 将字符串转换成数字并将其存储在一个等长度的int数组中

**int** [] arr = **new** **int**[sArr.length];

**for** (**int** i = 0; i < arr.length; i++) {

arr[i] = Integer.*parseInt*(sArr[i]);

}

//3,排序

Arrays.*sort*(arr);

//4,将排序后的结果遍历并拼接成一个字符串27 38 46 50 91

StringBuffer sb = **new** StringBuffer();

**for** (**int** i = 0; i < arr.length; i++) {

**if** (i == arr.length - 1) {

sb.append(arr[i]);

} **else** {

sb.append(arr[i] + " ");

}

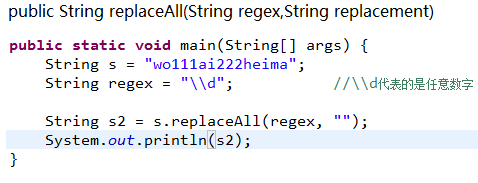
}

System.*out*.println(sb);

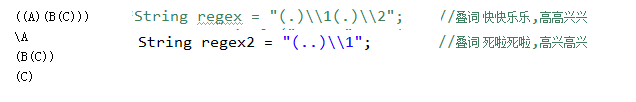
}

}

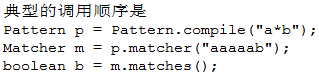
### 替换功能



### 分组功能



### 获取功能



# Pattern与Match类

## 模式对象

使用Pattern类创建一个对象，称作模式对象，模式对象是对正则表达式的封装

## 匹配对象

模式对象p调matcher(CharSequence input)方法返回一Matcher对象m，称作匹配对象



下列几个方法是Matcher对象m常用的方法：



获取电话号码

