# 常用API(三)





- > 常见算法
  - ◆ 简单认识算法 Arrays.sort(数组);
  - ◆ 排序算法
  - ◆ 查找算法
- > 正则表达式
- ▶ 异常



## 什么是算法?

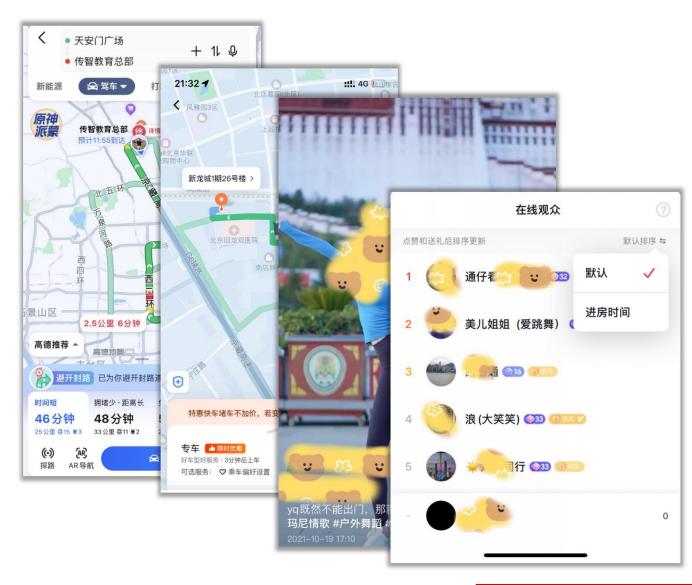
解决某个实际问题的过程和方法!

为什么要学习算法?

编程思维







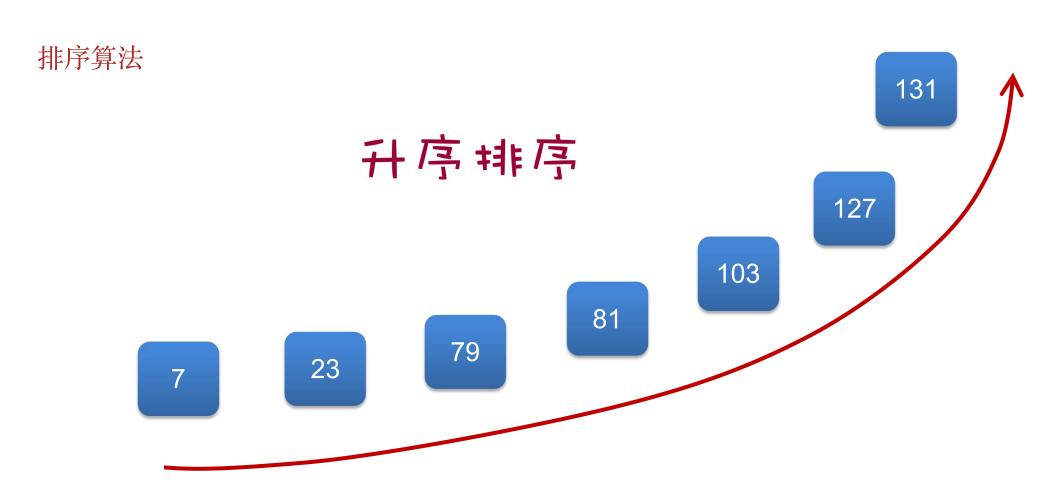


#### > 常见算法

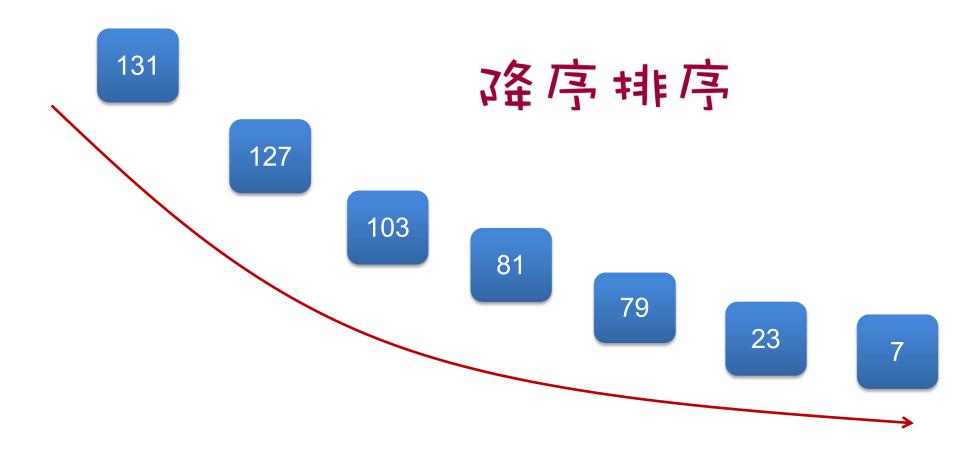
- ◆ 简单认识算法
- ◆ 排序算法
- ◆ 查找算法
- > 正则表达式
- ▶ 异常













冒泡排序

选择排序

...

## 学习算法的转巧

1、先搞清楚算法的流程。

2、直接去推敲如何写代码。



## 冒泡排序

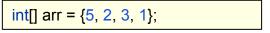
● 每次从数组中找出最大值放在数组的后面去。

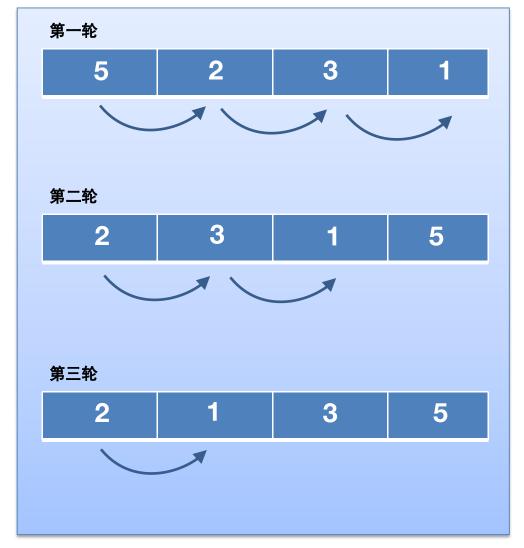
#### 实现冒泡排序的关键步骤分

- 确定总共需要做几轮: 数组的长度-1.
- 每轮比较几次:

i(第几轮)	比较位	次数规律: 数组的长度 - i-1
0	0 1 2	(3次)
1	0 1	(2次)
2	0	(1次)

● 当前位置大于后一个位置则交换数据







冒泡排序

选择排序

...



## 选择排序

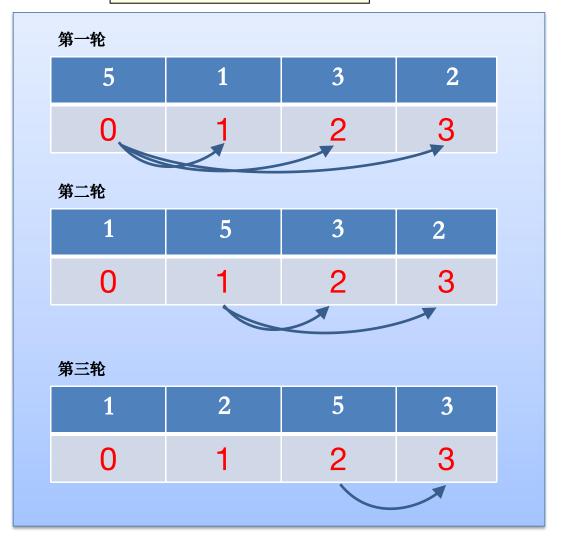
● 每轮选择当前位置,开始找出后面的较小值与该位置交换

#### 选择排序的关键

- 确定总共需要选择几轮: 数组的长度-1.
- 控制每轮从以前位置为基准,与后面元素选择几次。

i(第几轮)	比较位	次数
0	1 2 3	(3次)
1	2 3	(2次)
2	3	(1次)

int[] arr = {5, 1, 3, 2};





#### > 常见算法

- ◆ 简单认识算法
- ◆ 排序算法
- ◆ 查找算法
- ▶ 正则表达式
- ▶ 异常



#### 基本查找/顺序查找



注意: 在数据量特别大的时候,基本查找这种从前往后挨个找的形式,性能是很差的!



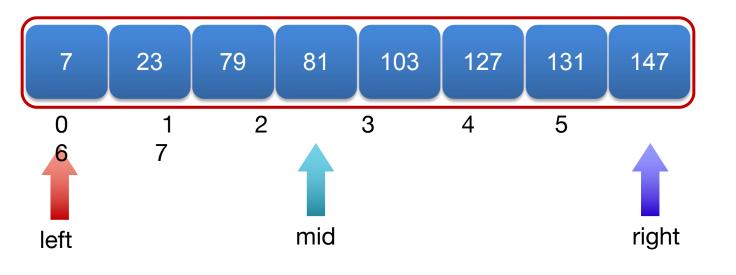


前提条件:数组中的数据必须是有序的

核心思想:每次排除一半的数据,查询数据的性能明显提高极多!

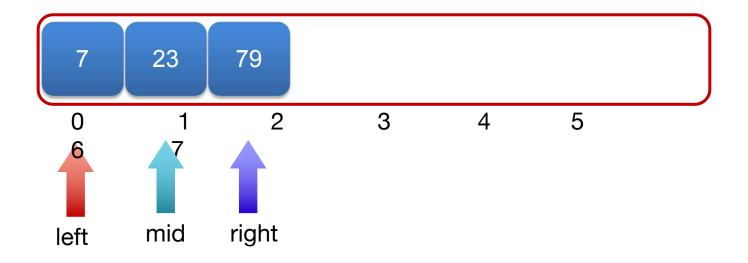




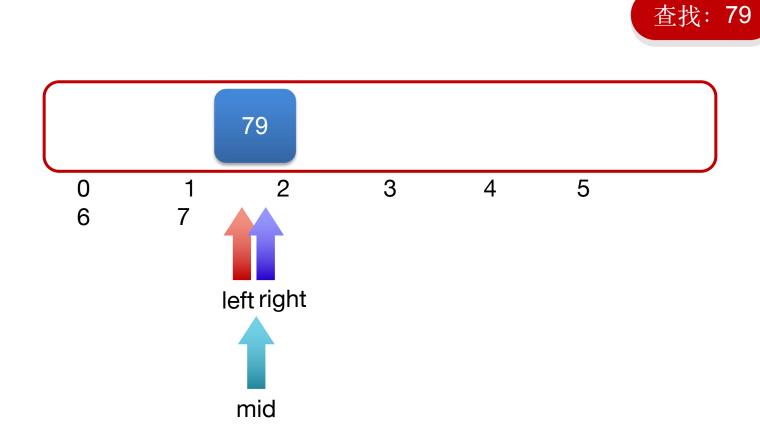


mid= (left+right) /2

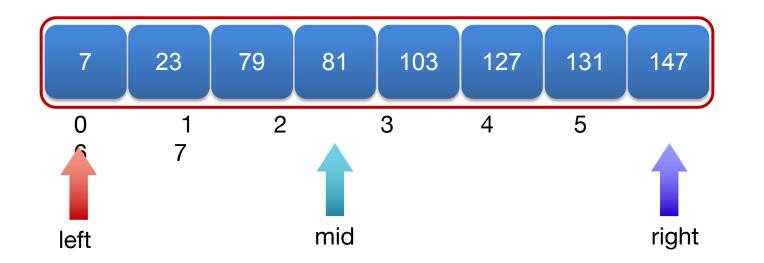




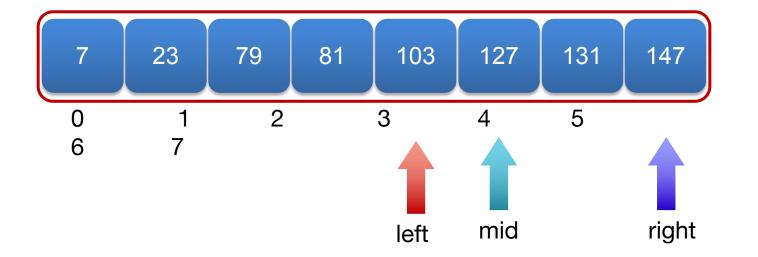




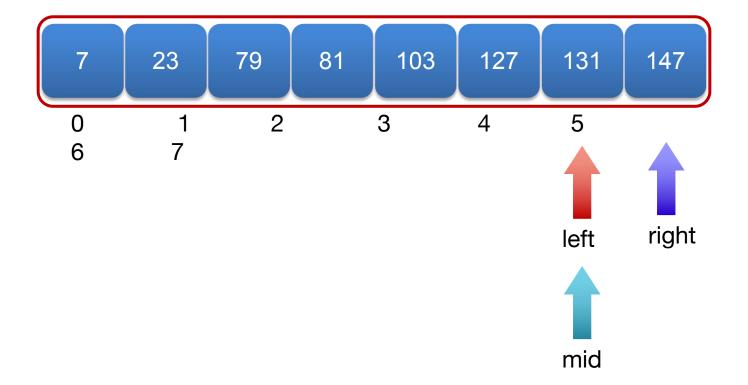




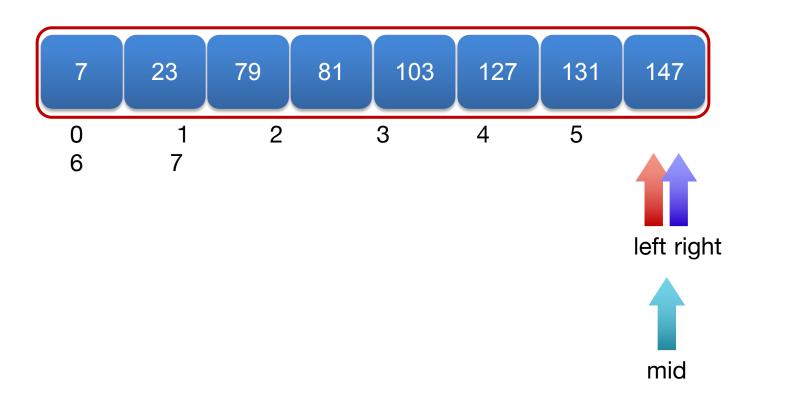






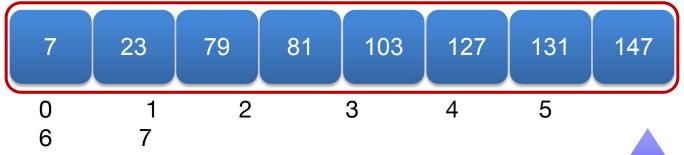




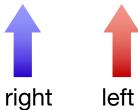




查找: 150



结论: 二分查找正常的折半条件应该是开始位置left <= 结束位置right







#### 1、二分查找的实现步骤是什么样的?

- 定义变量记录左边和右边位置。
- 使用while循环控制二分查询(条件是左边位置<=右边位置)
- 循环内部获取中间元素索引
- 判断当前要找的元素如果大于中间元素,左边位置=中间索引+1
- 判断当前要找的元素如果小于中间元素,右边位置=中间索引-1
- 判断当前要找的元素如果等于中间元素,返回当前中间元素索引。



- > 常见算法
- 正则表达式
  - ◆ 概述、初体验
  - ◆ 书写规则
  - ◆ 应用案例
  - ◆ 用于查找信息
  - ◆ 用于搜索替换、分割内容
- ▶ 异常



#### 正则表达式

● 就是由一些特定的字符组成,代表的是一个规则。

作用一:

03

用来校验数据格式是否合法

更简单!更便捷!

**01** 电话: 13811112222, 10233233A, 020-3212323

02 邮箱: dlei@163.com, dlei#gmail.com

QQ号: 12abc, 25666666, 09999

#### 作用二:

#### 在一段文本中查找满足要求的内容

身份证号码和姓名 性别 所属	属地区	Σ
向玉宇 41080119930228457x	男	河南省焦作市市辖区
高新瑶 510801197609022309	女	四川省广元市市辖区
孔弘济 150401198107053872	男	内蒙古自治区赤峰市市辖区
林冬卉 130133197204039024	女	河北省石家庄市赵县
邱迎海 430102197606046442	女	湖南省长沙市芙蓉区
刘如松 632722197112040806	女	青海省玉树藏族自治州杂多县
侯含雁 130683199011300601	女	河北省保定市安国市
赵弘新 350111199409241690	男	福建省福州市晋安区
杜力强 522323198705037737	男	贵州省黔西南布依族苗族自治州普安县
白香彤 510182197109294463	女	四川省成都市彭州市
万明辉 653221197910077436	男	新疆维吾尔族自治区和田地区和田县
张元彤 533526197206260908	女	云南省临沧地区双江拉祜族佤族布朗族傣
族自治县		
肖乐珍 230305198909078721	女	黑龙江省鸡西市梨树区
石初曼 232304198204030301	女	黑龙江省绥化地区海伦市
董华采 411425198812189711	男	河南省商丘市虞城县
高乐音 350521197404071798	男	福建省泉州市惠安县
孔弘阔 542128198709025957	男	西藏自治区昌都地区左贡县
刘含灵 350321198401316749	女	福建省莆田市莆田县
金丹琴 440804197710034663	女	广东省湛江市坡头区
陆雨信 372900197507012999	男	山东省菏泽地区
向玉宇 41080119930228457x	男	河南省焦作市市辖区
高新瑶 510801197609022309	女	四川省广元市市辖区
孔弘济 150401198107053872	男	内蒙古自治区赤峰市市辖区
林冬卉 130133197204039024	女	河北省石家庄市赵县
邱迎海 430102197606046442	女	湖南省长沙市芙蓉区
刘如松 632722197112040806	女	青海省玉树藏族自治州杂多县
侯含雁 130683199011300601	女	河北省保定市安国市



- > 常见算法
- 正则表达式
  - ◆ 概述、初体验
  - ◆ 书写规则
  - ◆ 应用案例
  - ◆ 用于查找信息
  - ◆ 用于搜索替换、分割内容
- ▶ 异常



#### String提供了一个匹配正则表达式的方法

public boolean matches(String regex)

判断字符串是否匹配正则表达式, 匹配返回true, 不匹配返回false。

#### 正则表达式的书写规则

[abc] : 只能是a, b, 或c

[^abc] : 除了a, b, c之外的任何字符

[a-zA-Z] : a到z A到Z,包括(范围)

[a-d[m-p]] : a到d,或m到p

[a-z&&[def]] : d, e, 或f(交集)

[a-z&&[^bc]] : a到z,除了b和c(等同于[ad-z])

[a-z&&[^m-p]]: a到z,除了m到p(等同于[a-lq-z])

字符类(只匹配单个字符)

	任何字符
\d	一个数字: [0-9]
\D	非数字: [^0-9]
\s	一个空白字符:
\S	非空白字符: [^\s]
\w	[a-zA-Z_0-9]
\W	[^\w] 一个非单词字符

 X?
 X,一次或0次

 X\*
 X,零次或多次

 X+
 X,一次或多次

 X {n}
 X,正好n次

 X {n, }
 X,至少n次

 X {n,m}
 X,至少n但不超过m次

数量词

预定义字符(只匹配单个字符)



## 小结

1、String类提供了哪个方法与正则表达式进行匹配?

判断字符串是否匹配正则 public boolean matches(String regex) 表达式,匹配返回true,不匹配返回false。

2、正则表达式的书写规则有哪些?

符号	含义	举例
?	0次或1次	\\d?
*	0次或多次	\\d* (abc)*
+	1次或多次	\\d+ (abc)+
{}	具体次数	a{7} \\d{7,19}
(?i)	忽略后面字符的大小写	(?i)abc
a((?i)b)c	只忽略b的大小写	α((?i)b)c

符号	含义	举例
0	里面的内容出现一次	[abc]
^	取反	[^abc]
&&	交集,不能写单个的&	[a-z&&m-p]
	任意字符	\n 回车符号不匹配
\	转义字符	\\d
\\d	0-9	\\d+
\\D	非0-9	\\D+
\\s	空白字符	
\\S	非空白字符	[^\s]
\\w	单词字符	[a-zA-Z_0-9]
\\W	非单词字符	[^\w]
0	分组	a(bc)+
I	写在方括号外面表示并集	ab   AB



- > 常见算法
- 正则表达式
  - ◆ 概述、初体验
  - ◆ 书写规则
  - ◆ 应用案例
  - ◆ 用于查找信息
  - ◆ 用于搜索替换、分割内容
- ▶ 异常





请使用正则表达式完成如下需求

### 需求

校验用户输入的电话、邮箱、时间是否合法



- > 常见算法
- > 正则表达式
  - ◆ 概述、初体验
  - ◆ 书写规则
  - ◆ 应用案例

作用一: 用来

用来校验数据格式是否合法

◆ 用于查找信息

作用二:

在一段文本中查找满足要求的内容

- ◆ 用于搜索替换、分割内容
- ▶ 异常





#### 使用正则表达式查找一段文本中的内容

需求:请把下面文本中的电话,邮箱,座机号码,热线都爬取出来。

来黑马程序员学习Java,

电话: 18512516758, 18512508907

或者联系邮箱: boniu@itcast.cn

座机电话: 01036517895, 010-98951256

邮箱: bozai@itcast.cn,

邮箱2: <u>dlei0009@163.com</u>,

热线电话: 400-618-9090, 400-618-4000,

4006184000, 4006189090



## 1 案例

#### 使用正则表达式查找一段文本中的内容

需求:请把下面文本中的电话,邮箱,座机号码,热线都爬取出来。

来黑马程序员学习Java,

电话: 18512516758, 18512508907

或者联系邮箱: boniu@itcast.cn

座机电话: 01036517895, 010-98951256

邮箱: <u>bozai@itcast.cn</u>,

邮箱2: <u>dlei0009@163.com</u>,

热线电话: 400-618-9090, 400-618-4000,

4006184000, 4006189090

```
String data = "来黑马程序员学习Java, \n"+
         电话: 1866668888, 18699997777\n"+
         或者联系邮箱: boniu@itcast.cn, \n"+
         座机电话: 01036517895, 010-98951256\n"+
         邮箱: bozai@itcast.cn, \n"+
         邮箱2: dlei0009@163.com, \n"+
         热线电话: 400-618-9090, 400-618-4000, 4006184000, 4006189090";
// 1. 定义爬取规则(正则表达式)
String regex = ((w{1,}@((x,10)((...)w{2,10})(1,2)))" +
    "(1[3-9]\d{9})\d{2,5}-?\d{5,15})\d{0.7}\d{3,8}-?\d{3,8}";
// 2.把正则表达式封装成一个Pattern对象
Pattern pattern = Pattern.compile(regex);
// 3. 通过pattern 对象得到查找内容的匹配器
Matcher matcher = pattern.matcher(data);
// 4. 通过匹配器开始去内容中查找信息
while(matcher.find()){
  System.out.println(matcher.group()); //取出信息
```





#### 使用正则表达式查找一段文本中的内容

需求2: 只需要把每个邮箱中的用户名爬取出来。

来黑马程序员学习Java,

电话: 18512516758, 18512508907

或者联系邮箱: boniu@itcast.cn

座机电话: 01036517895, 010-98951256

邮箱: bozai@itcast.cn,

邮箱2: dlei0009@163.com,

热线电话: 400-618-9090, 400-618-4000,

4006184000, 4006189090

```
String data = "来黑马程序员学习Java, \n"+
        电话: 1866668888, 18699997777\n"+
        或者联系邮箱: boniu@itcast.cn, \n"+
        座机电话: 01036517895, 010-98951256\n"+
         邮箱: bozai@itcast.cn, \n"+
        邮箱2: dlei0009@163.com, \n"+
        热线电话: 400-618-9090, 400-618-4000, 4006184000, 4006189090";
// 1. 定义爬取规则(正则表达式)
String regex = (\w{2,})@\w{2,20}(\.\w{2,10}){1,2}";
// 2. 把正则表达式封装成一个Pattern对象
Pattern pattern = Pattern.compile(regex);
// 3. 通过pattern对象得到查找内容的匹配器
Matcher matcher = pattern.matcher(data);
// 4.通过匹配器开始去内容中查找信息
while(matcher.find()){
  System.out.println(matcher.group(1)); // 取出第一组信息
```





#### 使用正则表达式查找一段文本中的内容

需求3: 某系统的日志文件记录了当天进入系统的全部用户信息,需要把这些用户的名字爬取出来另作他用

欢迎张全蛋光临本系统!他删库并跑路

欢迎李二狗子光临本系统!

欢迎马六子光临本系统!它浏览了很多好看的照片!

欢迎夏洛光临本系统!他在六点钟送出了一个嘉年华

```
String data = "欢迎张全蛋光临本系统! 他删库并跑路,欢迎李二狗子光临本系统!" +
        "欢迎马六子光临本系统! 它浏览了很多好看的照片! 欢迎夏洛光临本系统! 他在六点钟购买了一台拖拉机!";

// // , 定义爬取规则

String regex = "欢迎(.+)光临"; // .+是贪婪匹配

String regex2 = "欢迎(.+?)光临"; // .+是贪婪匹配

// 2、把正则表达式封装成一个Pattern对象

Pattern pattern = Pattern.compile(regex2);

// 3、通过pattern对象去获取查找内容的匹配器对象。

Matcher matcher = pattern.matcher(data);

// 4、定义一个循环开始爬取信息

while (matcher.find()){

String rs = matcher.group(); // 获取到了找到的内容了。
System.out.println(rs);

String rs2 = matcher.group(1); // 获取第一组内容(就是人的名字)
System.out.println(rs2);

}
```



- 》 常见算法
- 正则表达式
  - ◆ 概述、初体验
  - ◆ 书写规则
  - ◆ 应用案例
  - ◆ 用于查找信息
  - ◆ 用于搜索替换、分割内容
- **异常**



## 正则表达式用于搜索替换、分割内容,需要结合String提供的如下方法完成:

方法名	说明
public String replaceAll(String regex , String newStr)	按照正则表达式匹配的内容进行替换
public String[] split(String regex):	按照正则表达式匹配的内容进行分割字符串,反回一个字符串数组。



- 》 常见算法
- 正则表达式
- > 异常 异常就是代表程序出现的问题
  - ◆ 认识异常
  - ◆ 自定义异常
  - ◆ 异常的处理



# 什么是异常?

#### 异常就是代表程序出现的问题

 $int[] arr = {10, 20, 30};$ 

System.out.println(arr[3]);

System.out.println(10 / 0);

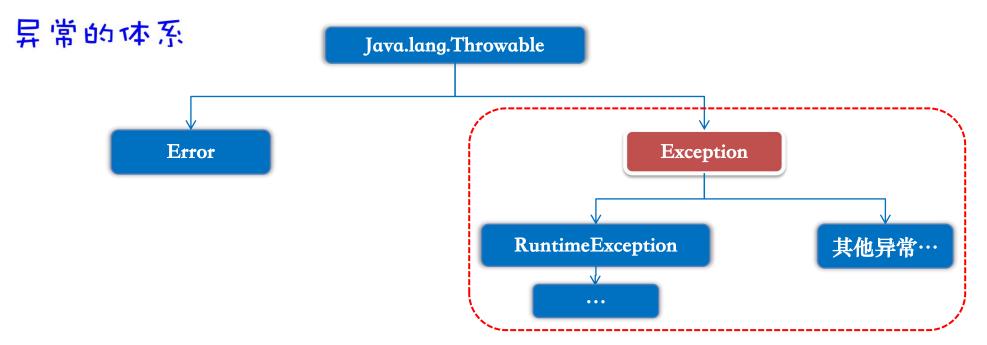
读取一个不存在的文件 读取网络数据

0 0 0

Exception in thread "main" java.lang.<a href="main" ArrayIndexOutOfBoundsException">ArrayIndexOutOfBoundsException</a> Create breakpoint: Index 3 out of bounds for length 3 at com.itheima.d3\_exception.ExceptionTest1.main(<a href="main">ExceptionTest1.main(ExceptionTest1.java:9)</a>)

Exception in thread "main" java.lang. <u>ArithmeticException</u> Create breakpoint: / by zero at com.itheima.d3\_exception.ExceptionTest1.main(<u>ExceptionTest1.java:8</u>)





**Error**:代表的系统级别错误(属于严重问题),也就是说系统一旦出现问题,sun公司会把这些问题封装成Error对象给出来,说白了,Error是给sun公司自己用的,不是给我们程序员用的,因此我们开发人员不用管它。

Exception: 叫异常,它代表的才是我们程序可能出现的问题,所以,我们程序员通常会用Exception以及它的孩子来封装程序出现的问题。

- 运行时异常: RuntimeException及其子类,编译阶段不会出现错误提醒,运行时出现的异常(如:数组索引越界异常)
- **编译时异常**:编译阶段就会出现错误提醒的。(如:日期解析异常)



#### 抛出异常 (throws)

● 在方法上使用throws关键字,可以将方法内部出现的异常抛出去给调用者处理。

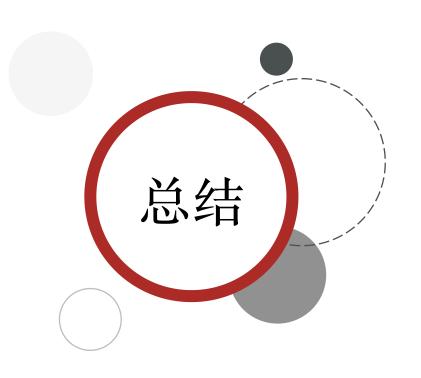
```
方法 throws 异常1 ,异常2 ,异常3 ..{
...
}
```

## 捕获异常(try…catch)

● 直接捕获程序出现的异常。

```
try{
    // 监视可能出现异常的代码!
}catch(异常类型1 变量){
    // 处理异常
}catch(异常类型2 变量){
    // 处理异常
}...
```





- 1. 异常是什么?
  - 异常是代码在编译或者执行的过程中可能出现的错误。
- 2. 异常的代表是谁? 分为几类?
  - Exception, 分为两类:编译时异常、运行时异常。
  - 编译时异常:没有继承RuntimeExcpetion的异常,编译阶段就会出错。
  - 运行时异常:继承自RuntimeException的异常或其子类,编译阶段不报错,运行时出现的。
- 3. 异常代码层面的处理有几种方式?
  - 用来查找bug;可以作为方法内部的特殊返回值,通知上层调用者底层的执行情况。

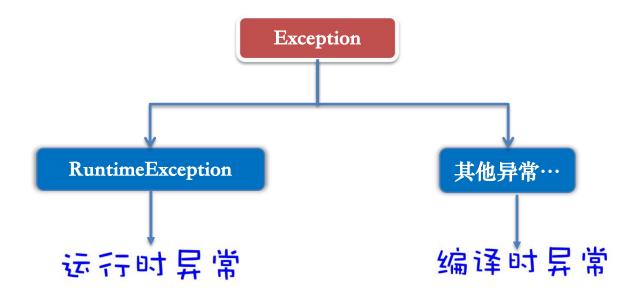


- > 常见算法
- > 正则表达式
- **异常** 
  - ◆ 认识异常
  - ◆ 自定义异常
  - ◆ 异常的处理



# 自定义异常

● Java无法为这个世界上全部的问题都提供异常类来代表,如果企业自己的某种问题,想通过异常来表示,以便用异常来管理该问题,那就需要自己来定义异常类了。





#### 自定义异常

● Java无法为这个世界上全部的问题都提供异常类来代表,如果企业自己的某种问题,想通过异常来表示,以便用异常来管理该问题,那就需要自己来定义异常类了。

## 自定义异常的种类

# 自定义运行时异常

- 定义一个异常类继承RuntimeException.
- 重写构造器。
- 通过throw new 异常类(xxx)来创建异常对象并抛出。 编译阶段不报错,提醒不强烈,运行时才可能出现!!

# 自定义编译时异常

- 定义一个异常类继承Exception.
- 重写构造器。
- 通过throw new 异常类(xxx)来创建异常对象并抛出。

编译阶段就报错,提醒更加强烈!

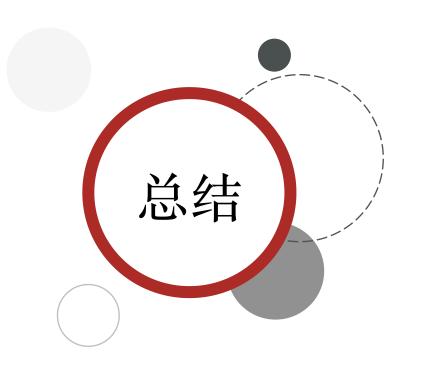


# 异常有什么作用?

1、异常是用来查寻系统Bug的关键参考信息!

Exception in thread "main" java.lang.<a href="mainto:ArrayIndexOutOfBoundsException">ArrayIndexOutOfBoundsException</a> Create breakpoint: Index 3 out of bounds for length 3 at com.itheima.d3\_exception.ExceptionTest1.main(<a href="mainto:ExceptionTest1.java:9">ExceptionTest1.java:9</a>)

2、异常可以作为方法内部的一种特殊返回值,以便通知上层调用者底层的执行情况!



1. 什么是自定义异常?

2. 自定义异常有几类? 分别怎么定义?

3. 异常的作用有哪些?



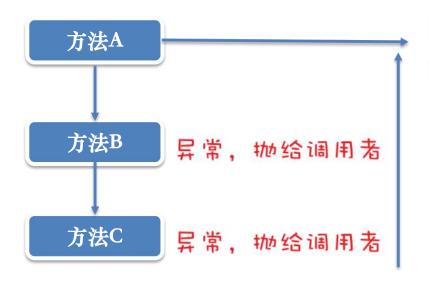
- > 常见算法
- 正则表达式
- **异常** 
  - ◆ 认识异常
  - ◆ 自定义异常
  - ◆ 异常的处理

throws

try-catch



## 开发中对于异常的常见处理方式



- 1、埔获异常,记录异常并响应合适的信息给用户
- 2、埔获异常,尝试重新修复



#### 抛出异常 (throws)

● 在方法上使用throws关键字,可以将方法内部出现的异常抛出去给调用者处理。

```
方法 throws 异常1 ,异常2 ,异常3 ..{
...
}
```

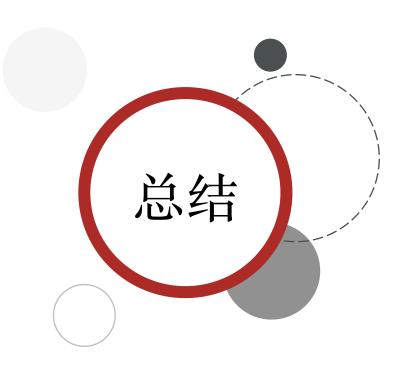
```
// 推荐方式
方法 throws Exception{
}
// Exception代表可以捕获一切异常
```

#### 捕获异常(try…catch)

● 直接捕获程序出现的异常。

```
//推荐方式
try{
    //可能出现异常的代码!
}catch (Exception e){
    e.printStackTrace(); //直接打印异常对象的信息
}
//Exception代表可以捕获一切异常
```





- 1、异常处理的总结?
- 在开发中异常的常见处理方式是:底层的异常抛出去给最外层,最外层集中捕获处理。







传智教育旗下高端IT教育品牌