

数组

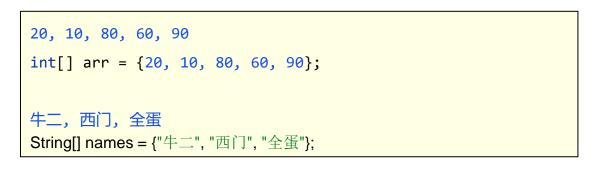




数组是什么

● 数组就是用来存储一批同种类型数据的内存区域(可以理解成容器)。

例子







随机点名

张誉	刘疏桐	田启峰
尹成元	赵志刚	解天野
高续升	王世举	周宇伦
刘帅	陈小伟	陈伟鸿
高明	毕振宇	王乾麟
丁佳峰	李鑫宇	孙孟想
陈耀	尹于林	摄定廉
马煜	林圣凯	范洪庆
黄豪龙	罗磊	金胜锋
孙胜贤	陈陇红	李骁昂
谢明宏	张瑜	潘进东
陆鑫	吴玉成	徐圣博
高凯	卢贞旭	刘明辉
武善涛	钱嘉怡	沙含川
梁才焜	王卫平	王挺希
朱华安	游志康	姜振兴
钱天奇	赵志强	高耀东
金牮刚	王景	赵山剑
曹道华	李振清	候衡山
白哲东	王素素	杨宇航
萬金龙	刘永超	田假桥
戴天舒	崔彬涛	宋健强
林永超	张学颖	李泌霖
陈侃		

假如用变量存储这些名字,然后完成随机点名功能,怎么实现?存在有什么问题?

```
String name1 = "张誉";
String name2 = "刘疏桐";
String name3 = "田启峰";
...
String name68= "张学颖";
String name69= "李沁霖";
String name70= "陈侃";
```

- 代码繁琐:大量变量的定义。
- 业务功能实现麻烦。



其实你真的很好 只是我们不合适

使用数组完成:

String[] names = {"张誉", "刘疏桐", "田启峰", ... "张学颖", "李沁霖", "陈侃",};

结论: 数组适合做一批同种类型数据的存储。





关于数组同学们需要学会什么

怎么定义数组存储数据

怎么操作数组元素

怎么解决实际问题

数组内存原理

数组使用的注意点

在程序中如何去定 义数组存储数据, 具体格式是什么样 的 怎么去获取,修改数 组中的数据

怎么结合数组解决比 如随机点名等一些实 际业务

数组在内存中具体 是怎么去工作的

避免入坑,同时可以在 出现问题后立即排查问 题所在

> 数组的定义

- ◆ 静态初始化数组
 - 数组的访问
 - 数组的几个注意事项

◆ 动态初始化数组

- 动态初始化数组的元素默认值
- > 数组的遍历
- > 数组的案例
- > 数组的内存图
- 数组使用的常见问题
- Debug工具的使用





静态初始化数组

● 定义数组的时候直接给数组赋值。

静态初始化数组的格式:

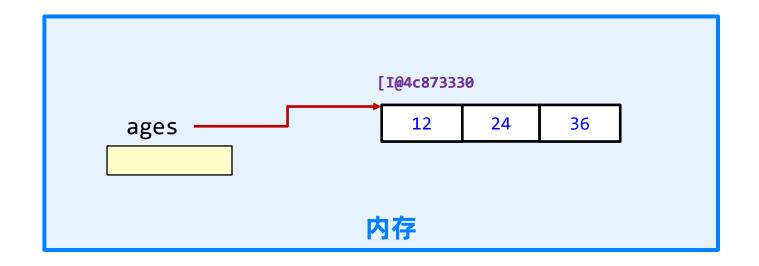
```
// 完整格式
数据类型[] 数组名 = new 数据类型[]{元素1,元素2,元素3...};
double[] scores = new double[]{89.9, 99.5, 59.5, 88.0};
int[] ages = new int[]{12, 24, 36}
```

```
//简化格式
数据类型[]数组名 = { 元素1,元素2 ,元素3,… };
int[] ages = {12, 24, 36};
```



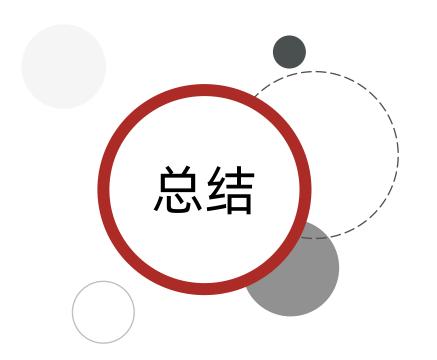
数组的基本原理

$$int[] ages = {12, 24, 36};$$



注意:数组变量名中存储的是数组在内存中的地址,数组是引用类型。





1. 数组的静态初始化的写法是和特点什么样的?

```
数据类型[] 数组名 = { 元素1, 元素2 , 元素3, ... };
int[] ages = {12, 24, 36, 48, 60};
double[] scores = {89.9, 99.5, 59.5};

// 完整格式
数据类型[] 数组名 = new 数据类型[]{ 元素1, 元素2 , 元素3... };
int[] ages = new int[]{12, 24, 36, 48, 60};
```

- 2. 数组是属于什么类型,数组变量名中存储的是什么?
 - 引用数据类型,存储的数组在内存中的地址信息。



- > 数组的定义
 - ◆ 静态初始化数组
 - 数组的访问
 - 数组的几个注意事项
 - ◆ 动态初始化数组
 - 动态初始化数组的元素默认值
- > 数组的遍历
- > 数组的案例
- > 数组的内存图
- 数组使用的常见问题
- Debug工具的使用



数组的访问

数组名称[索引]

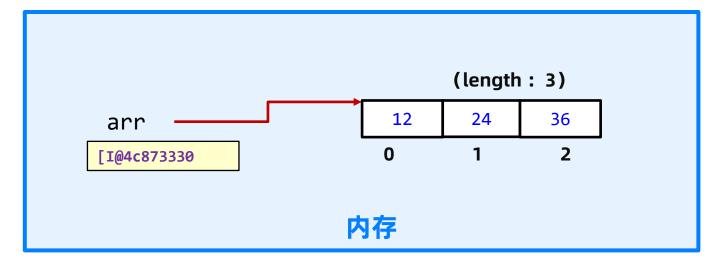
```
// 取值
System.out.println(arr[0]); // 12

// 赋值
arr[2] = 100;
System.out.println(arr[2]); // 100
```

数组的长度属性: length

```
// 获取数组的长度(就是数组元素的个数)
System.out.println(arr.length); // 3
```

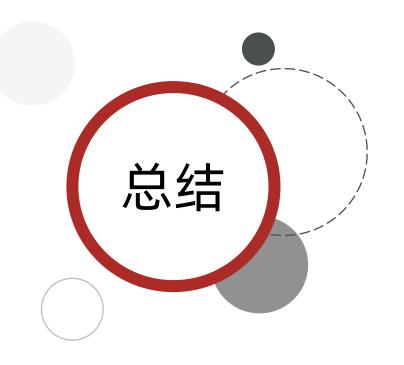
 $int[] arr = \{12, 24, 36\};$



问题:数组的最大索引可以怎么表示?

数组名. length - **1** // 前提: 元素个数大于0





1. 如何访问数组的元素?

数组名称[索引]

- 2. 如何访问数组的长度?
 - 数组名称.length
- 3. 数组的最大索引怎么获取?

数组名. length - 1 // 前提: 元素个数大于0



- > 数组的定义
 - ◆ 静态初始化数组
 - 数组的访问
 - 数组的几个注意事项
 - ◆ 动态初始化数组
 - 动态初始化数组的元素默认值
- > 数组的遍历
- > 数组的案例
- > 数组的内存图
- 数组使用的常见问题
- Debug工具的使用



数组的几个注意事项:

● "数据类型[] 数组名"也可以写成 "数据类型 数组名[]"。

```
int[] ages =...;
int ages[] =...;

double[] scores = ...;
double scores[] = ...;
```

● 什么类型的数组存放什么类型的数据,否则报错。

```
int[] arrs = new int[]{30, 40, "黑马"};
```

● 数组一旦定义出来,程序执行的过程中,长度、类型就固定了。



> 数组的定义

- ◆ 静态初始化数组
 - 数组的访问
 - 数组的几个注意事项
- ◆ 动态初始化数组
 - 动态初始化数组的元素默认值
- > 数组的遍历
- > 数组的案例
- > 数组的内存图
- 数组使用的常见问题
- Debug工具的使用



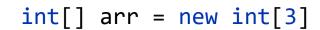
数组的动态初始化

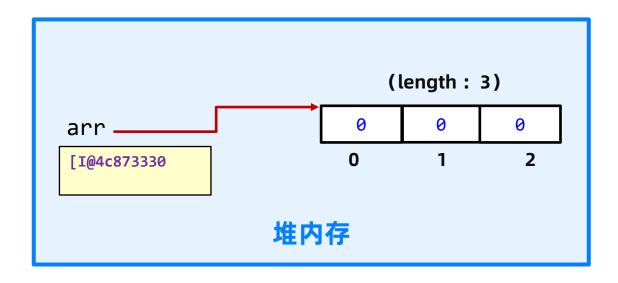
● 定义数组的时候只确定元素的类型和数组的长度,之后再存入具体数据。

数组的动态初始化格式:

```
数据类型[] 数组名 = new 数据类型[长度];
int[] arr = new int[3];
```

```
// 后赋值
arr[0] = 10;
System.out.println(arr[0]); // 10
```









1. 动态初始化的写法是什么样的?

```
数据类型[] 数组名 = new 数据类型[长度];
int[] ages = new int[4];
```

- 2. 两种数组定义时的特点和场景有什么区别
 - 当前已经知道存入的元素值,用静态初始化。
 - 当前还不清楚要存入哪些数据,用动态初始化。



- > 数组的定义
 - ◆ 静态初始化数组
 - 数组的访问
 - 数组的几个注意事项
 - ◆ 动态初始化数组
 - 动态初始化数组的元素默认值
- > 数组的遍历
- > 数组的案例
- > 数组的内存图
- 数组使用的常见问题
- Debug工具的使用



元素默认值规则:

数据类型	明细	默认值
	byte、short、char、int、long	0
基本类型	float、double	0.0
	boolean	false
引用类型	类、接口、数组、 String	null

两种初始化的的使用场景总结、注意事项说明:

● 动态初始化:只<mark>指定数组长度,后期赋值,适合开始知道数据的数量,但是不确定具体元素值的业务场景。</mark>

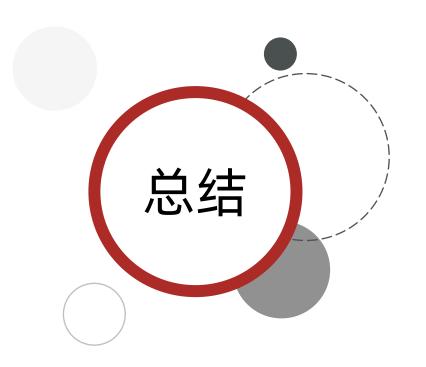
● 静态初始化:开始就存入元素值,适合一开始就能确定元素值的业务场景。

● 两种格式的写法是独立的,不可以混用。

int[] arrs = new int[3]{30, 40,50};







- 1. 动态初始化数组后元素的默认值是什么样的?
 - byte、short、int、char、long类型数组元素的默认值都是0
 - float、double类型数组元素的默认值都是0.0
 - boolean类型数组元素的默认值是<mark>false</mark>、String类型数组元素的默认值是<mark>null</mark>



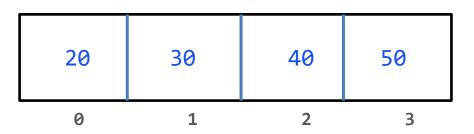
- > 数组的定义
- > 数组的遍历
- > 数组的案例
- > 数组的内存图
- 数组使用的常见问题
- **▶** Debug工具的使用



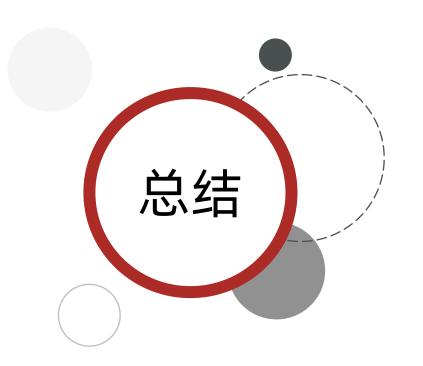
数组遍历介绍

- 遍历: 就是一个一个数据的访问。
- 为什么要遍历? 搜索、数据统计等等都需要用到遍历。

```
int[] ages = {20, 30, 40, 50};
for (int i = 0; i < ages.length; i++) {
    System.out.println(ages[i]);
}</pre>
```







- 1. 什么是遍历?
 - 一个一个的把数据访问一遍
- 2. 如何遍历数组?

```
int[] ages = {20, 30, 40, 50};
for (int i = 0; i < ages.length; i++) {
    System.out.println(ages[i]);
}</pre>
```



- > 数组的定义
- > 数组的遍历
- > 数组的案例
 - ◆ 数组元素求和
 - ◆ 数组求最值
 - ◆ 猜数字游戏
 - ◆ 随机排名
 - ◆ 数组排序
- > 数组的内存图
- > 数组使用的常见问题
- Debug工具的使用





数组遍历-求和

需求:某部门5名员工的销售额分别是:16、26、36、6、100,请计算出他们部门的总销售额。

分析:

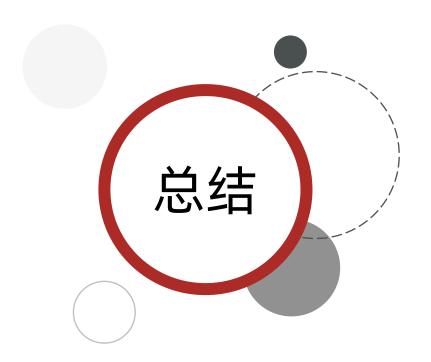
① 把这5个数据拿到程序中去 ---> 使用数组

```
int[] money = {16, 26, 36, 6, 100};
```

② 遍历数组中的每个数据,然后在外面定义求和变量把他们累加起来。

```
int sum = 0;
for (int i = 0; i < money.length; i++) {
    // i = 0 1 2 3 4
    sum += money[i];
}</pre>
```





- 1. 如何实现批量数据的求和?
 - 使用数组存储批量数据

```
int[] money = {16, 26, 36, 6, 100};
```

● 遍历数组中的每个数据,然后定义变量把他们累加起来。

```
int sum = 0;
for (int i = 0; i < money.length; i++) {
    // i = 0 1 2 3 4
    sum += money[i];
}</pre>
```



- > 数组的定义
- > 数组的遍历
- > 数组的案例
 - ◆ 数组元素求和
 - ◆ 数组求最值
 - ◆ 猜数字游戏
 - ◆ 随机排名
 - ◆ 数组排序
- > 数组的内存图
- > 数组使用的常见问题
- Debug工具的使用





数组元素求最大值





1 案例

数组元素求最大值

颜值: 15 颜值: 9000 颜值: 10000 颜值: 20000 颜值: 9500 颜值: -5

分析:

① 把颜值数据拿到程序中去,用数组装起来。

```
int[] faceScores = {15, 9000, 10000, 20000, 9500, -5};
```

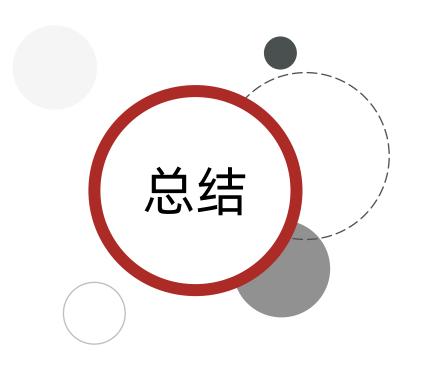
② 定义一个变量用于记录最大值,这个变量建议默认存储第一个元素值作为参照。

```
int max = faceScores[0];
```

③ 遍历数组的元素,如果该元素大于变量存储的元素,则替换变量存储的值为该元素。

④ 循环结束后输出最大值变量即可。





1. 数组元素求最大值如何实现的?

- ① 数据拿到程序中去,用数组装起来。
- ② 定义一个变量用于记录最大值,这个变量建议默认存储第一个元素值作为参照。
- ③ 遍历数组的元素,如果该元素大于变量存储的元素,则替换变量存储的值为该元素。
- ④ 循环结束后输出最大值变量即可。



- > 数组的定义
- > 数组的遍历
- > 数组的案例
 - ◆ 数组元素求和
 - ◆ 数组求最值
 - ◆ 猜数字游戏
 - ◆ 随机排名
 - ◆ 数组排序
- > 数组的内存图
- > 数组使用的常见问题
- **Debug工具的使用**





猜数字游戏

需求

开发一个幸运小游戏,游戏规则如下:

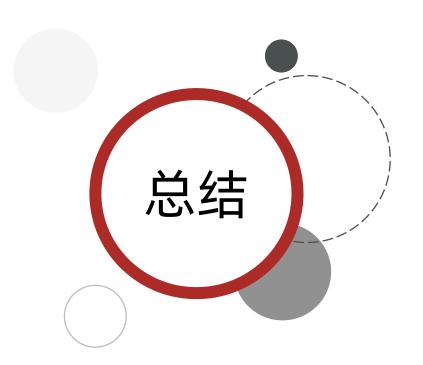
游戏后台随机生成1-20之间的5个数(无所谓是否重复),然后让大家来猜数字:

- ▶ 未猜中提示: "未命中", 并继续猜测
- 猜中提示: "运气不错,猜中了",并输出该数据第一次出现的位置,且输出全部5个数据,最终结束本游戏。

分析

- ① 随机生成5个1-20之间的数据存储起来 ---> 使用数组
- ② 定义一个死循环,输入数据猜测,遍历数组,判断数据是否在数组中,如果在,进行对应提示并结束死循环;如果没有猜中,提示继续猜测直到猜中为止。





1. 猜数字游戏的实现步骤?

- ① 动态初始化数组,存入5个随机的1-20之间的数据
- ② 定义一个死循环,不断的猜数据,遍历数组,判断数据是否在数组中,如果在,进行 对应提示并结束死循环;如果没有猜中,提示继续。



- > 数组的定义
- > 数组的遍历
- > 数组的案例
 - ◆ 数组元素求和
 - ◆ 数组求最值
 - ◆ 猜数字游戏
 - ◆ 随机排名
 - ◆ 数组排序
- > 数组的内存图
- 数组使用的常见问题
- **Debug工具的使用**





随机排名

需求

某公司开发部5名开发人员,要进行项目进展汇报演讲,现在采取随机排名后进行汇报。请先依次录入5名员工的工号,然后展示出一组随机的排名顺序。

分析

- ① 在程序中录入5名员工的工号存储起来 ---> 使用数组。
- ② 依次遍历数组中的每个元素,随机一个索引数据,让当前元素与该索引位置处的元素进行交换。





元素交换的原理图







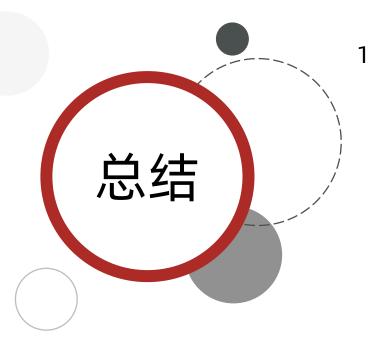
22 33 35 13 88



随机交换排名的其他使用场景







1. 如何实现随机排名的?

- ① 定义一个动态初始化的数组用于录入数据。
- ② 遍历数组中的每个元素,每次随机一个索引值,让当前元素与该索引位置处的元素进行交换。
- ③ 遍历输出数组中的内容即可。



- > 数组的定义
- > 数组的遍历
- > 数组的案例
 - ◆ 数组元素求和
 - ◆ 数组求最值
 - ◆ 猜数字游戏
 - ◆ 随机排名
 - ◆ 数组排序
- > 数组的内存图
- 数组使用的常见问题
- Debug工具的使用



数组排序

● 就是对数组中的元素,进行升序(由小到大)或者降序(由大到小)的操作。

数组排序的技术

- 冒泡排序
- 选择排序
- 快速排序
- 插入排序
- ...

数组搜索相关的技术

- 二分搜索
- 分块查找
- 哈希表查找
- ..



冒泡排序的思想

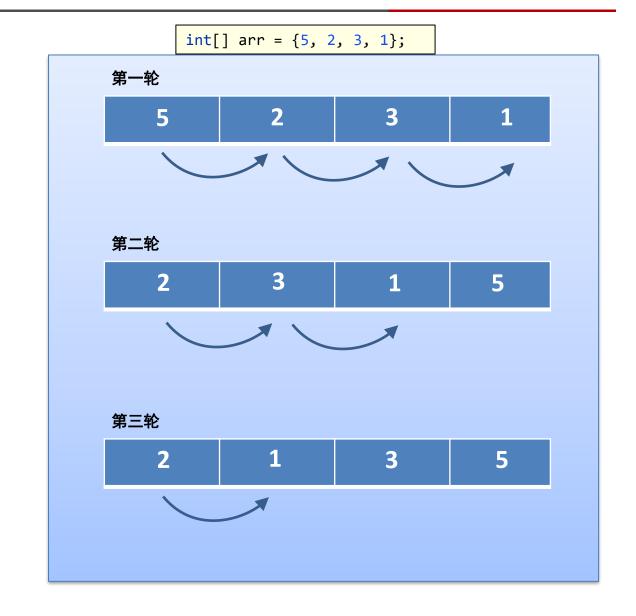
● 每次从数组中找出最大值放在数组的后面去。

实现冒泡排序的关键步骤分析

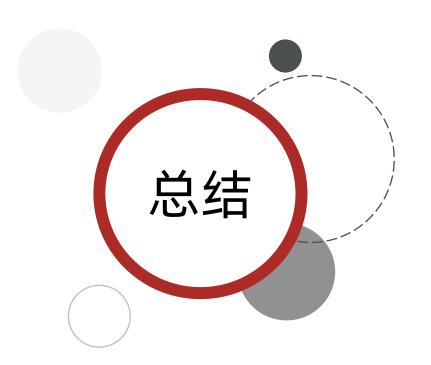
- 确定总共需要做几轮: 数组的长度-1.
- 每轮比较几次:

i(轮数)	次数	次数规律:数组的长度 - i
1	3	
2	2	
3	1	

● 当前位置大于后一个位置则交换数据







1. 冒泡排序的思想

- 从头开始两两比较,把较大的元素与较小的元素进行交换
- 每轮把当前最大的一个元素存入到数组当前的末尾。
- 2. 冒泡排序的实现步骤。
 - 定义一个外部循环控制总共需要冒几轮(数组的长度-1)
 - 定义一个内部循环,控制每轮依次往后比较几个位置(数组长度-i-1)。
 - 如果当前位置的元素值>后一个位置的元素值,两者交换。

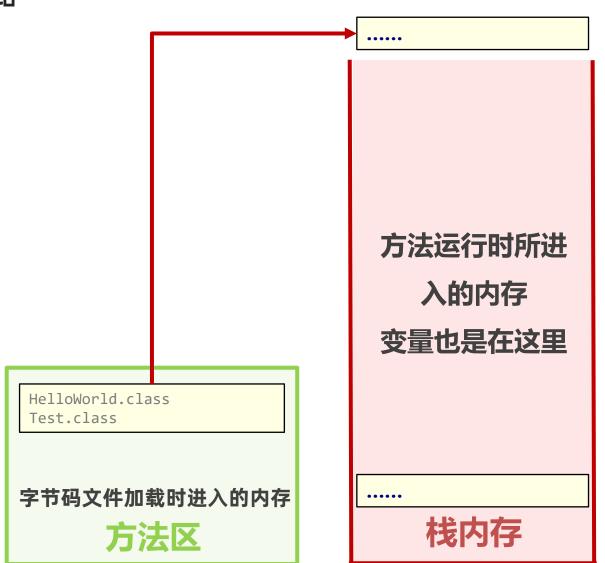


- > 数组的定义
- > 数组的遍历
- > 数组的案例
- > 数组的内存图
 - ◆ Java内存分配、数组内存图
 - ◆ 两个变量指向同一个数组
- 数组使用的常见问题
- **Debug工具的使用**



Java 内存分配介绍

- 栈
- 堆
- 方法区
- 本地方法栈
- 寄存器

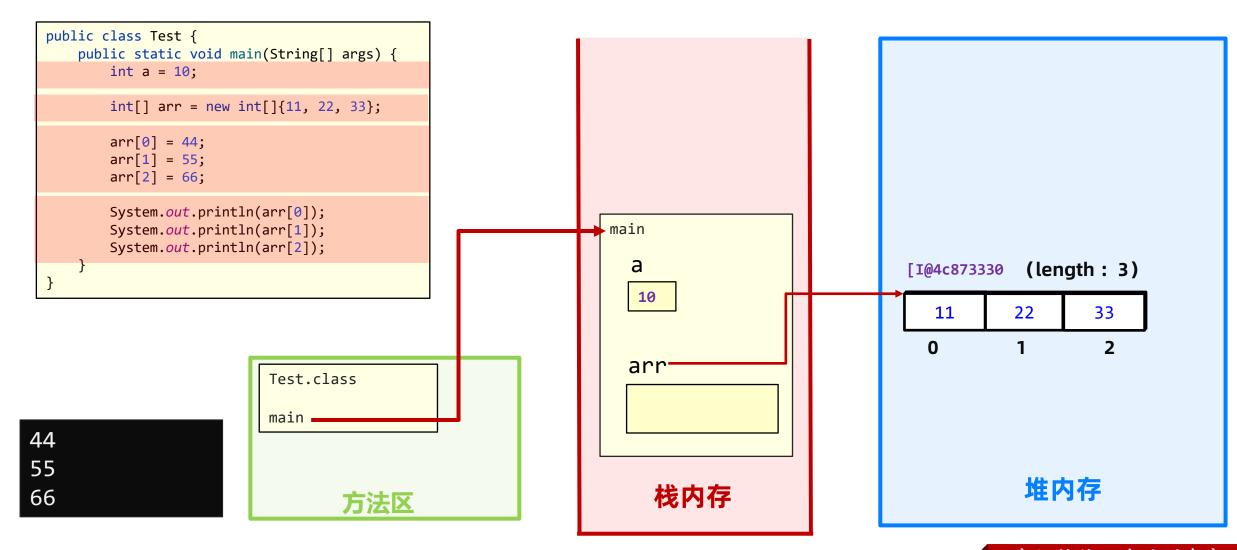


new 出来的东西会 在这块内存中开辟 空间并产生地址

堆内存



Java内存分配介绍

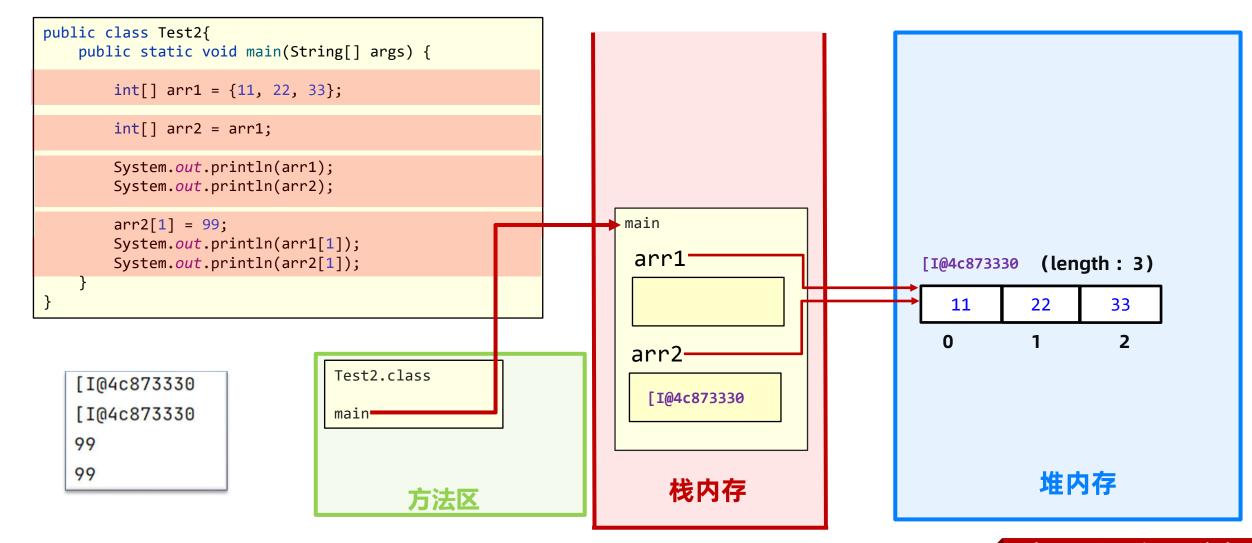




- > 数组的定义
- > 数组的遍历
- > 数组的案例
- > 数组的内存图
 - ◆ Java内存分配、数组内存图
 - ◆ 两个变量指向同一个数组
- 数组使用的常见问题
- **▶** Debug工具的使用



两个数组变量指向同一个数组对象





- > 数组的定义
- > 数组的遍历
- > 数组的案例
- > 数组的内存图
- 数组使用的常见问题
- Debug工具的使用



数组使用常见问题

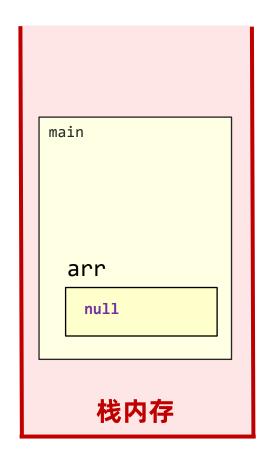
- 问题1: 如果访问的元素位置超过最大索引,执行时会出现ArrayIndexOutOfBoundsException(数组索引越界异常)
- 问题2: 如果数组变量中没有存储数组的地址,而是null,在访问数组信息时会出现NullPointerException(空指针异常)

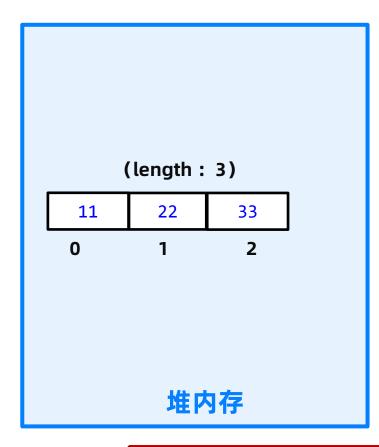
```
int[] arr = new int[]{11, 22, 33};
System.out.println(arr[2]);
// System.out.println(arr[3]) // 出现异常
```

出现ArrayIndexOutOfBoundsException异常

```
arr = null;
System.out.println(arr); // null
//System.out.println(arr.length) // 出现异常
```

出现NullPointerException异常







- > 数组的定义
- > 数组的遍历
- > 数组的案例
- > 数组的内存图
- 数组使用的常见问题
- **Debug工具的使用**

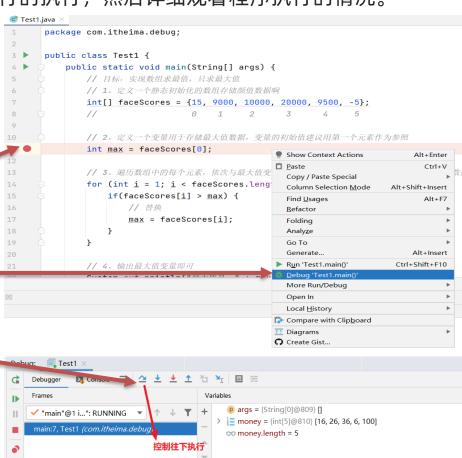


Debug工具

● IDEA自带的断点调试(排错)工具,可以控制代码从断点开始一行一行的执行,然后详细观看程序执行的情况。

DEBUG工具基本使用步骤

- ① 在需要控制的代码行左侧,点击一下,形成断点
- ② 选择使用Debug方式启动程序,启动后程序会在断点暂停
- ③ 控制代码一行一行的往下执行





传智教育旗下高端IT教育品牌