Java基础(5)[1]

数组

数组是具有相同数据类型且按一定次序排列的一组变量的集合体。即用一个变量名表示一批数据。Java为数组在内存中分配一段连续的空间，这段空间中存储数据的个数是固定的，数组就是一个容器，用来存一批同种类型的数据的。

> 遇到批量数据的存储和操作时，数组比变量更适合

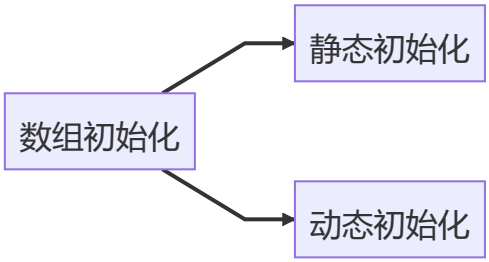
定义数组

Java中定义数组有两种语法格式：`数据类型 数组名[ ]`；或`数据类型[ ] 数组名`；

例如：`String names[]`或`String[] names`推荐第二种方式

语法解析：

- 数组是什么数据类型，数组的元素就是什么数据类型
- 数组的特征是[ ]
- 数组是引用类型



静态初始化数组

定义数组、为数组元素分配内存、数组元素初始化，这三步可以合并在一起写，例如：`int[] scores = new int[]{12,56,34,78};`或`int[] scores = {12,56,34,78};`

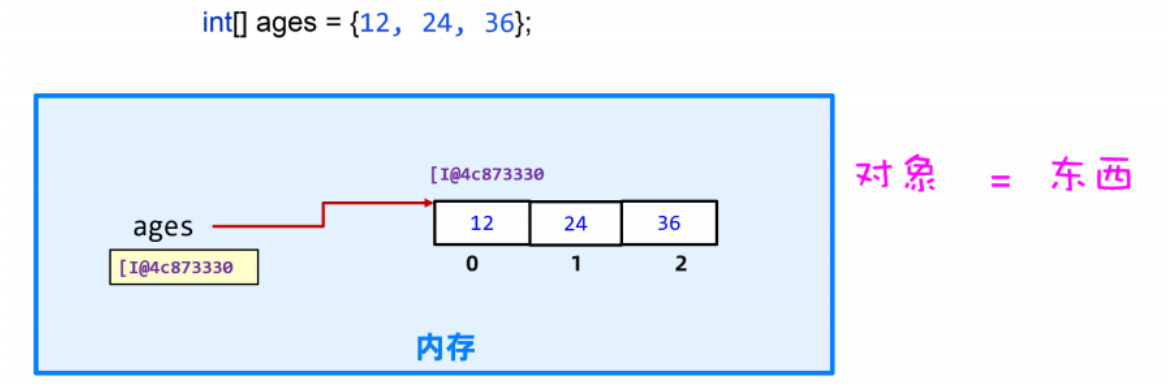
在定义数组时直接给数组中的数据赋值这种方式称为静态初始化。标准格式是

```
数据类型[] 变量名 = new 数据类型[]{元素1,元素2,元素3};

简化为:
数据类型[] 变量名 = {元素1,元素2,元素3};

String[] names = {"金文涛", "李瑶瑶", "小樊同学"...};
System.out.println(names[1]);
```

数组在计算机中的基本原理

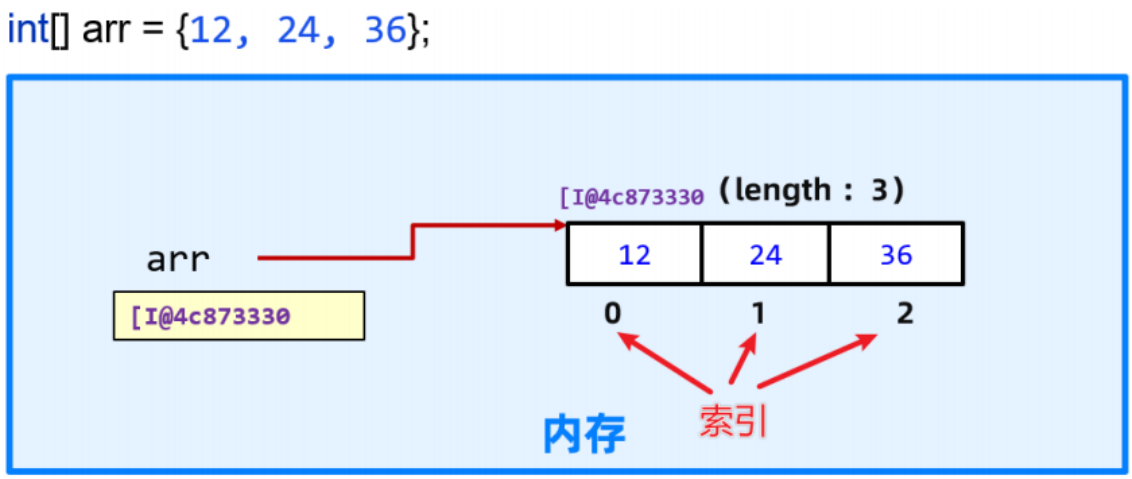
我们知道数组是怎么定义的之后，那么接下来看一下数组在计算机中的基本原理。我们以 `int[] ages = {12,24,36};` 这句话为例，看一下这句话到底在计算机中做了那些事情。

- 首先，左边 `int[] ages` 表示定义了一个数组类型的变量，变量名叫 `ages`
- 其次，右边 `{12,24,36}` 表示创建一个数组对象，你完全可以把它理解成一个
- 能装数据的东西。这个对象在内存中会有一个地址值 `[I@4c873330`，每次创建一个数组对象都会有不用的地址值。
- 然后，把右边的地址值 `[I@4c873330` 赋值给左边的ages变量
- 所以，ages变量就可以通过地址值，找到数组这个东西。



注意：数组变量名中存储的是数组在内存中的地址，数组是一种引用数据类型。

数组的元素访问



要想访问数组中的元素，格式如下

```
// 数组名可以找到数组对象的地址，再通过索引就可以定位到具体的元素了

数组名[索引] // 索引 0 ⟶ 长度 - 1

//技巧：获取数组的最大索引：arr.length - 1(前提是数组中存在数据)
```

数组的遍历

将数组中的元素一个一个的取出来。

```
int[] ages = {12, 24, 36};
for (int i = 0; i < ages.length; i++) {
// i的取值 = 0, 1, 2
System.out.println(ages[i]);
}
```

数组的动态初始化

刚才我们初始化数组时，都是直接将元素写出来。但是还有另一个初始化数组的方式叫 动态初始化。

动态初始化不需要我们写出具体的元素，而是指定元素类型和长度就行。格式如下

`数据类型[] 数组名;`

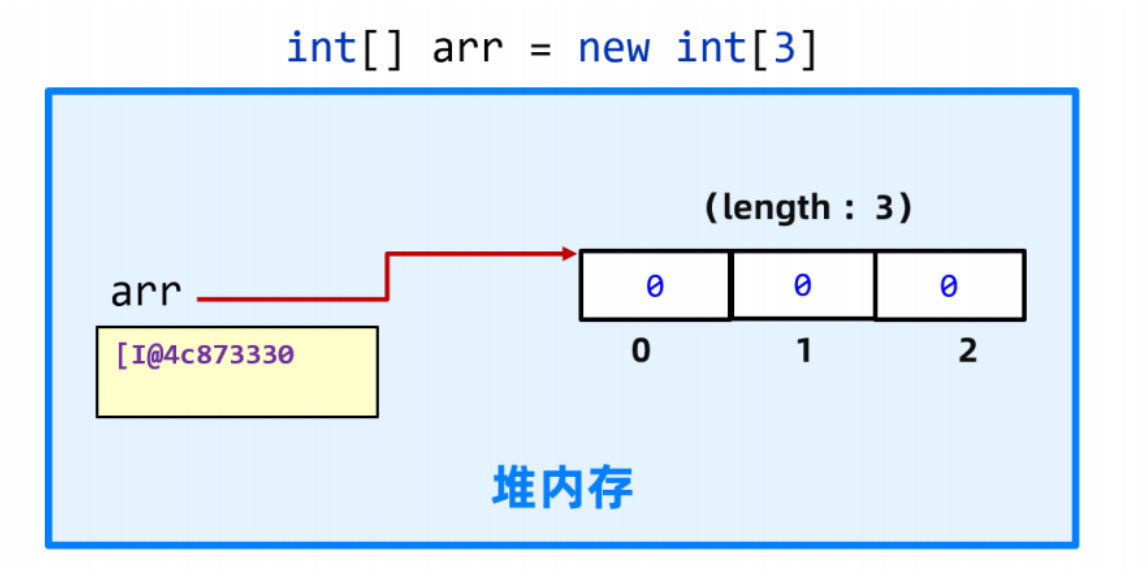
`数组名 = new 数据类型[数组长度];`

例如： `names = new String[5];`

定义数组和为数组元素分配内存，这两步可以合并在一起写，例如：`String[] names = new String[5];`

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

`int[] arr`其实就是一个变量，它记录了数组对象的地址值，而且数组中的元素默认值是0。



> 注意：
>
> 使用动态初始化定义数组时，根据元素类型不同，默认值也有所不同。

数组元素类型	默认初始值
byte，short，int，long	0
float，double	0.0
char	'\u0000'（空字符）
boolean	false
引用数据类型	null

数组使用过程可能出现的问题

```java
public class ArrayDemo03{
    public static void main(String[] args){
        int[] scores = {32,45,45,76};
        System.out.println(scores[5]); // 下标越界
        int[] ages = {32,43,444,32,'a'};
        System.out.println(ages[4]); // 97
        int[] ages1 = {32,43,444,32L};
        ages1[2] = 100;
        ages[2] = 100L;
        System.out.println(ages1[3]); // 报错
    }
}
```

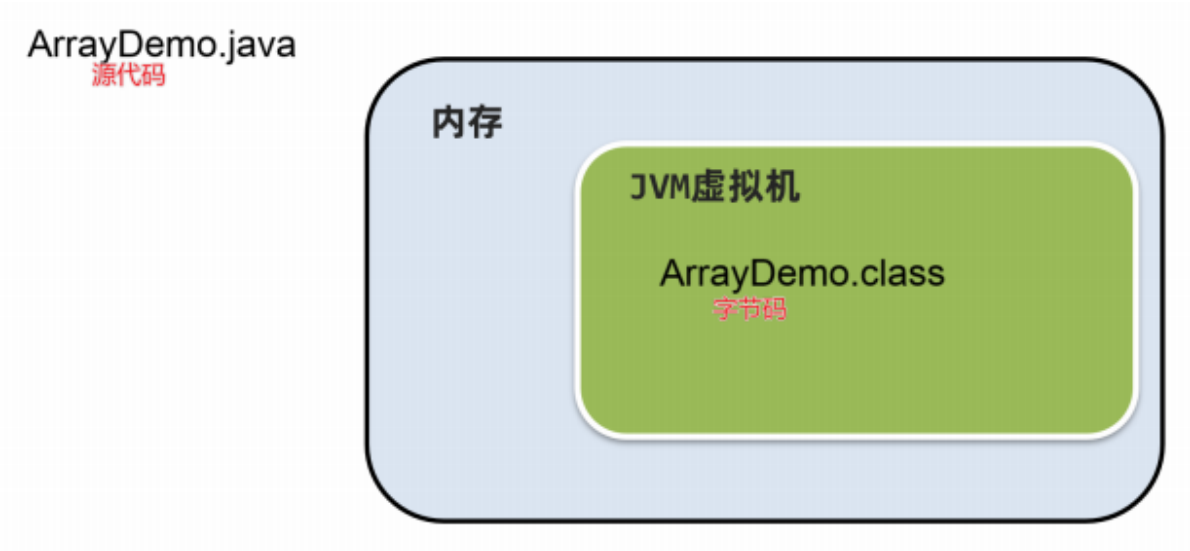
- 如果在数组中保存的元素可以自动提升(自动类型转化)为数组自己的类型，那是可以保存的
- 数组下标越界

数组在计算机中的执行原理

由于数组是一个容器，变量也是一个容器，在理解他们执行原理的时候，有些同学就

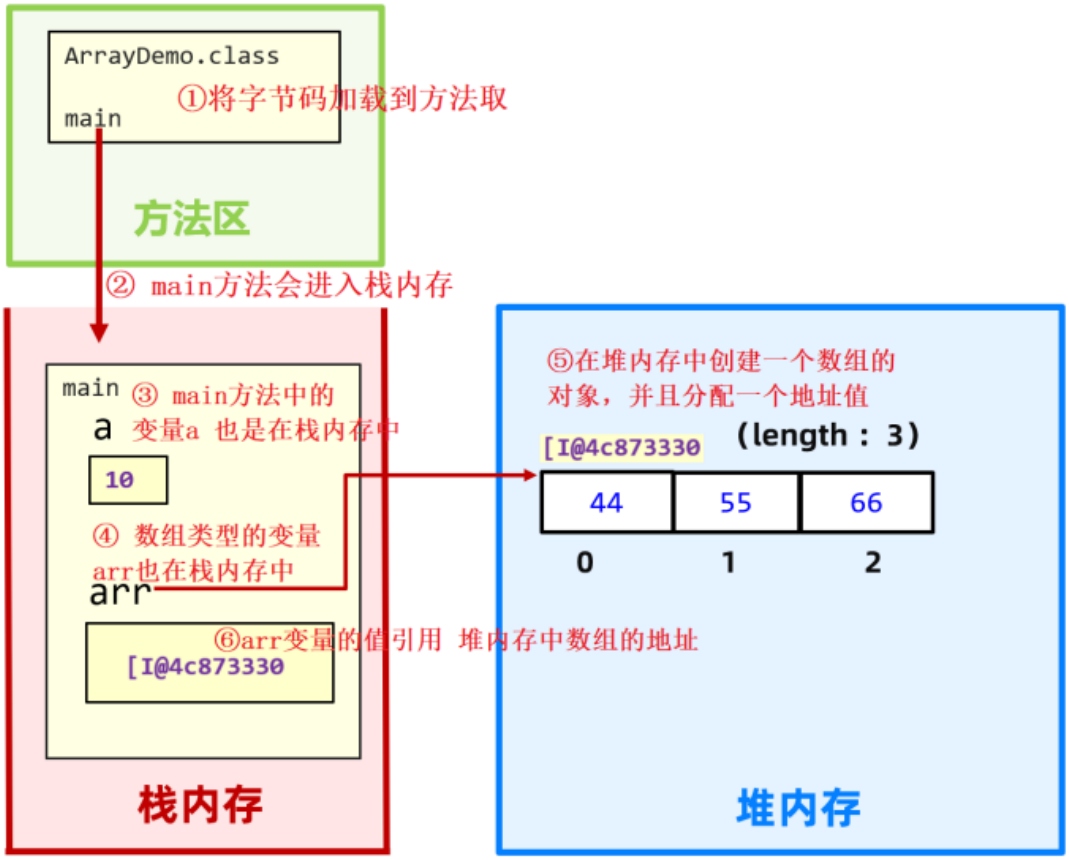
Java程序的执行的内存原理。

上Java程序是把编译后的字节码加载到Java虚拟机中执行的。



Java为了便于虚拟机执行Java程序，将虚拟机的内存划分为 方法区、栈、堆、本地方法栈、寄存器 这5块区域。同学们需要重点关注的是 方法区、栈、堆。

- 方法区：字节码文件先加载到这里
- ```
  Random rand = new Random();
  Scanner sc = new Scanner(System.in);
  ```
- 栈：方法运行时所进入的内存区域，由于变量在方法中，所以变量也在这一块区域中
- 堆：存储new出来的东西，并分配地址。由于数组是new 出来的，所以数组也在这块区域。



总结一下`int a = 10`与`int[] arr = new int[]{11,22,33}`的区别

- **a**是一个变量，在栈内存中，**a**变量中存储的数据就是**10**这个值。
- **arr**也是一个变量，在栈中，存储的是数组对象在堆内存中的地址值

```
// 这里的int a是一个基本类型变量，存储的是一个数值
int a = 10 ;
//这里的int[] arr是一个引用类型的变量，存储的是一个地址值
int[] arr = new int[]{44,55,66};
```

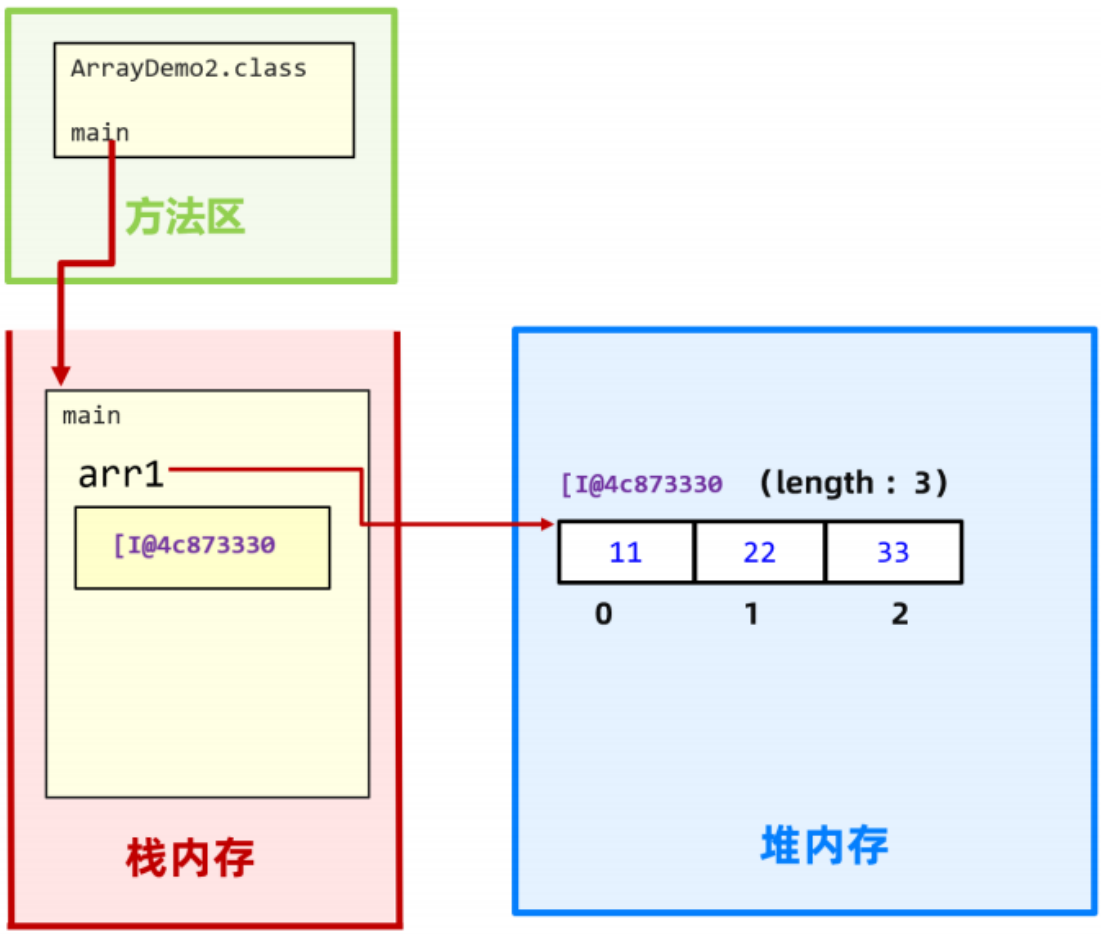
多个变量指向同一个数组的问题

数组类型的变量，指向的是堆内存中数组对象的地址。但是在实际开发中可能存在一种特殊情况，就是多个变量指向同一个数组对象的形式。

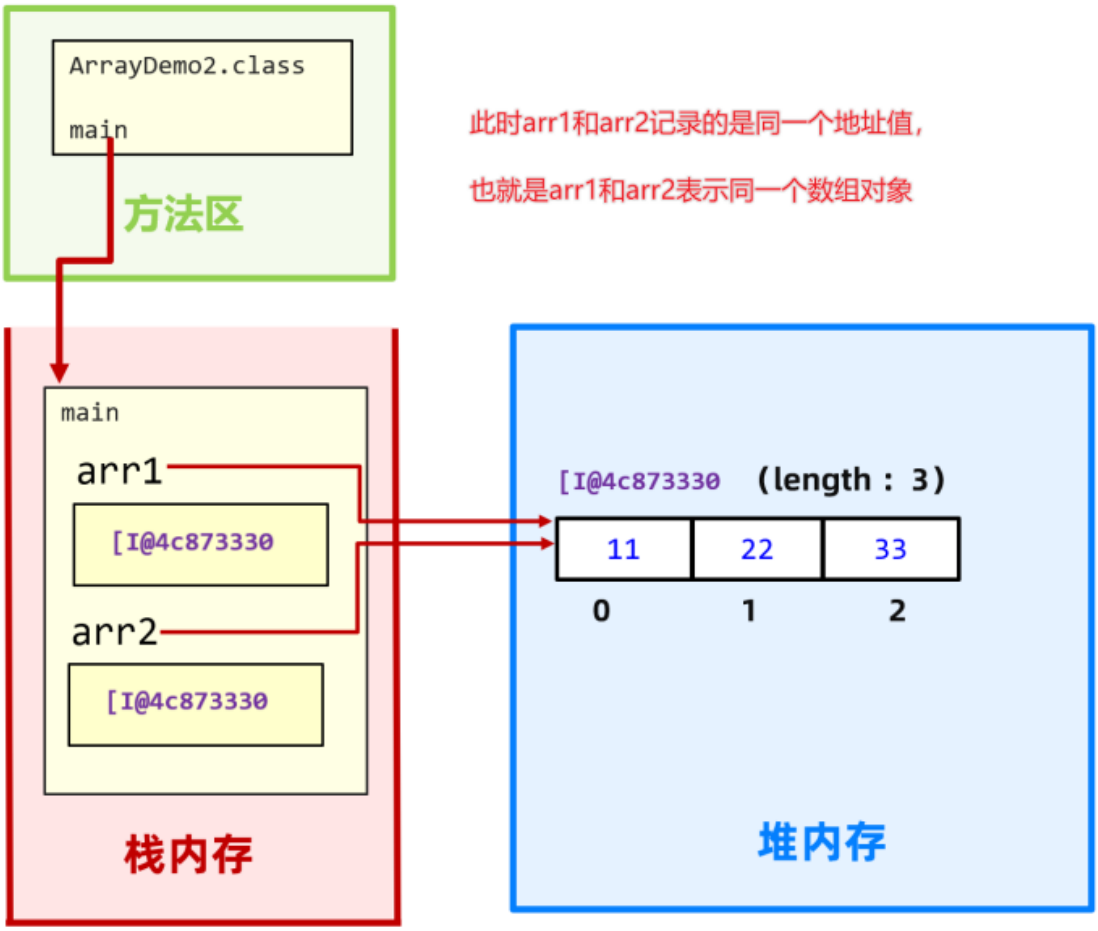
```
public class ArrayDemo2 {
```
```

```
2      public static void main(String[] args) {
3          // 目标：认识多个变量指向同一个数组对象的形式，并掌握其注意事项。
4          int[] arr1 = new int[]{11, 22, 33};
5          // 把int类型的数组变量arr1赋值给int类型的数组变量arr2
6          int[] arr2 = arr1;
7          int[] arr3 = new int[]{11, 22, 33};
8          System.out.println(arr1); // 地址
9          System.out.println(arr2); // 地址 1 == 2 都不一样
10         System.out.println(arr3); // 地址 3 不一样
11         arr2[1] = 99;
12         System.out.println(arr1[1]);
13         arr2 = null; // 拿到的数组变量中存储的值是null
14         System.out.println(arr2);
15         //System.out.println(arr2[0]);
16         //System.out.println(arr2.length);
17     }
18 }
```

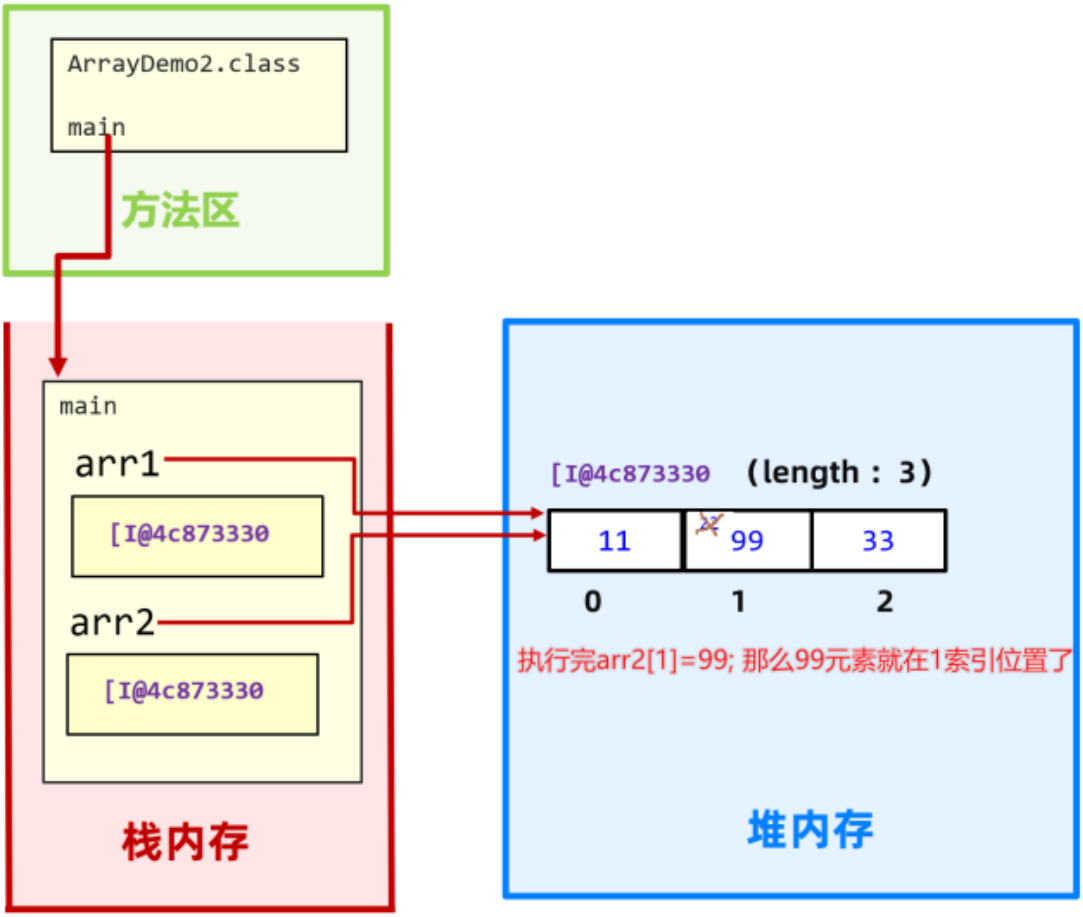
刚执行完`int[] arr1 = {11,22,33};`时，内存原理如下



当执行完`int[] arr2 = arr1;`后，内存原理如下



当执行到`arr2[1]=99;`时，内存原理如下



总结一下：

- 两个变量指向同一个数组时，两个变量记录的是同一个地址值。
- 当一个变量修改数组中的元素时，另一个变量去访问数组中的元素，元素已经被修改过了。

使用增强for循环遍历数组

JDK1.5及其之后的版本中提供了增强for循环语句，实现了Iterable接口的类都可以使

用增强for循环进行元素的迭代。增强for循环的语法规则如下：

```
for (元素类型 变量名 : 要迭代的对象) {
    System.out.println(变量名);
}
int[] arr = {15, 9000, 10000, 20000, 9500, -5};

for (int e : arr) {
    System.out.printn(e);
}

for (int i = 0; i < arr.length; i ++) {
    System.out.printn(arr[i]);
}
```

语法解析：

- 元素类型是指数组或集合中的元素的类型。
- 变量名在循环时用来保存每个元素的值。
- 冒号后面是要遍历的数组或集合的名称。

1. 学习Java第五天 ↩