

## 05.01\_Java 语言基础（数组概述和定义格式说明）（了解）

A: 为什么要有数组（容器）

为了存储同种数据类型的多个值

B: 数组概念

数组是存储同一种数据类型多个元素的集合，也可以看成是一个容器

数组既可以存储基本数据类型，也可以存储引用数据类型（面向对象时讲）

C: 数组定义格式

数据类型[] 数组名 = new 数据类型[数组的长度];

## 05.02\_Java 语言基础（数组的初始化动态初始化）（掌握）

A: 什么是数组的初始化

就是为数组开辟连续的内存空间，并为数组的每个元素赋值

B: 如何对数组进行初始化

a: 动态初始化，只指定长度，由系统给出初始化值

```
int[] arr = new int[10];
```

b: 静态初始化，给出初始化值，由系统决定长度

C: 动态初始化的格式：

数据类型[] 数据名 = new 数据类型[数组长度]，就是定义数组的格式

## 05.03\_Java 语言基础（java 中内存分配及栈和堆的区别）（掌握）

A: 栈：存储局部变量

局部变量：定义在方法里的变量

B: 堆：存储 new 出来的数组或对象

C: 方法区：面向对象部分讲解

```
public class Demo9 {
    public static void main(String[] args) {
        int[] a = new int[3];
        System.out.println(a); // [I@10f87f48
        /*
        [I@10f87f48
        [ : 代表一位数组
        I : 代表int 类型
        @ : 就是@符号
        10f87f48 : 是一个十六进制的整数
        */
        System.out.println(a[0]);
        System.out.println(a[1]);
        System.out.println(a[2]);
        System.out.println("-----");
        a[1] = 3;
        a[2] = 4;
        System.out.println(a[0]);
        System.out.println(a[1]);
        System.out.println(a[2]);
    }
}
```

```

    }
}

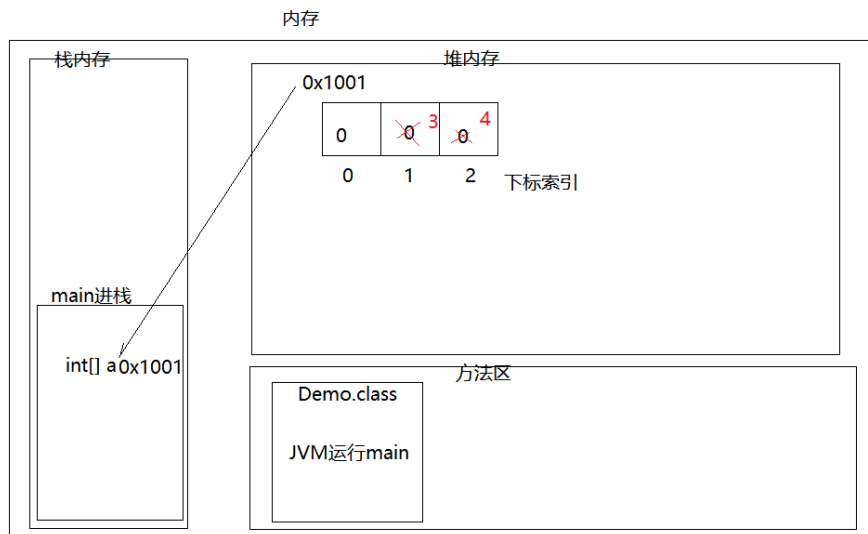
```

```

public class Demo9 {
    public static void main(String[] args) {
        int[] a = new int[3];

        a[1] = 3;
        a[2] = 4;
    }
}

```



```

public class Demo9 {
    public static void main(String[] args) {
        double[] a = new double[10];
        System.out.println(a[0]);
        System.out.println(a[1]);
        System.out.println(a[2]);

    }
}

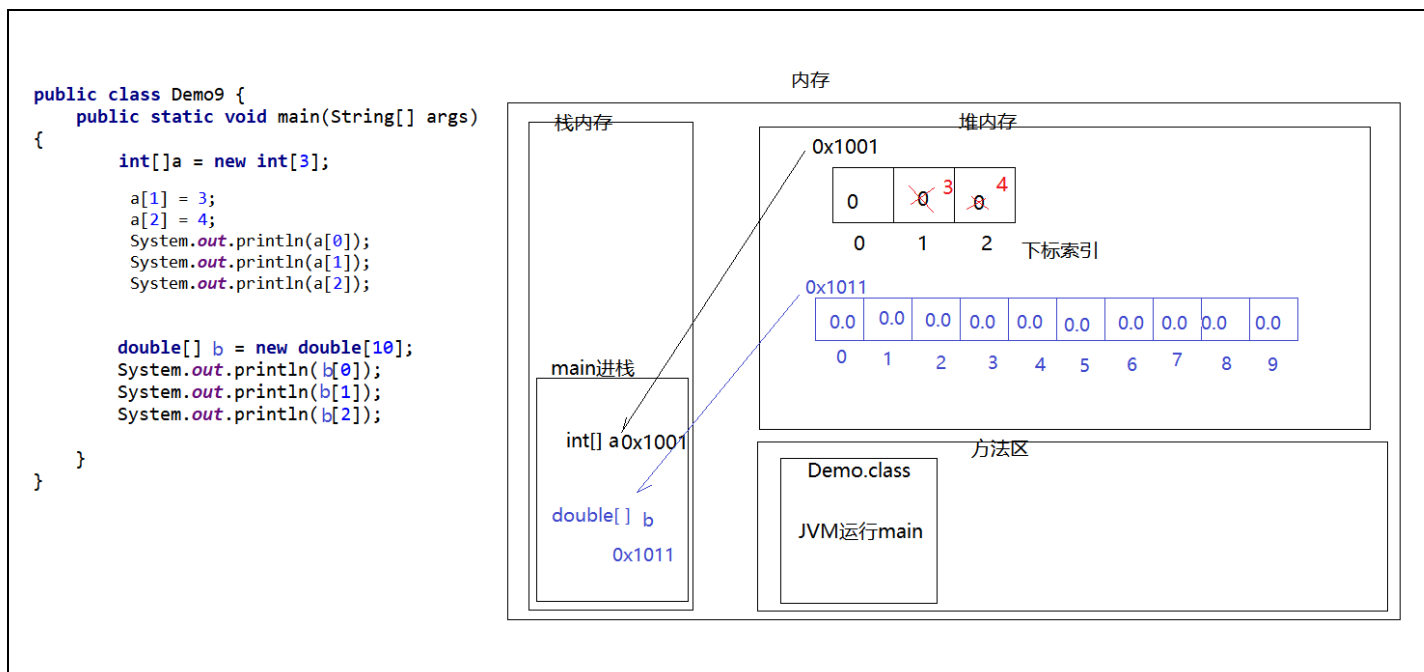
```

```

/*
各数据类型的默认值
byte\short\int\Long: 0
float:0.0f;
double:0.0
boolean:false
char:'\0000' =0  char 是两个字节: 0000 0000 0000 0000 二进制的四位合并为一位: 0000=0
String: null
*/

```

## 数组内存图解 2：两个数组



## 数组内存图解 3：三个引用两个数组

```
public class Demo9 {
    public static void main(String[] args) {
        int[] a = new int[3];
        System.out.println(a[1]);
        System.out.println(a[2]);
        System.out.println("数组 a 的内存地址: "+a);

        double[] b = new double[10];
        System.out.println(b[0]);
        System.out.println(b[1]);
        System.out.println(b[2]);

        int[] c = a;
        System.out.println("数组 c 的内存地址: "+c);
        System.out.println("a[0]="+a[0]);

        c[0] = 20;
        System.out.println("c[0]="+c[0]);
        System.out.println("a[0]="+a[0]);
    }
}

/*
各数据类型的默认值
byte\short\int\Long: 0
float:0.0f;
double:0.0
*/
```

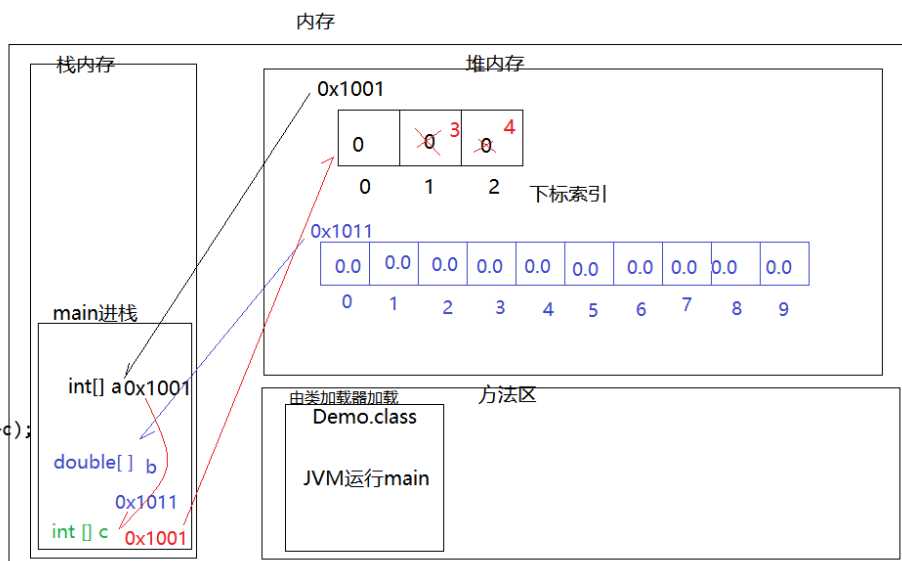
```
boolean:false
```

```
char:'\0000' =0 char 是两个字节: 0000 0000 0000 0000 十六进制是二进制的四位合并为一位: 0000=0
```

```
String: null
```

```
*/
```

```
public class Demo9 {  
    public static void main(String[] args)  
    {  
        int[] a = new int[3];  
        a[1] = 3;  
        a[2] = 4;  
        System.out.println(a[0]);  
        System.out.println(a[1]);  
        System.out.println(a[2]);  
  
        double[] b = new double[10];  
        System.out.println(b[0]);  
        System.out.println(b[1]);  
        System.out.println(b[2]);  
  
        int[] c = a;  
        System.out.println("数组c的内存地址: "+c);  
    }  
}
```



## 05\_Java 语言基础（静态初始化的格式）（掌握）

A: 静态初始化的格式:

格式: 数据类型[] 数组名 = new 数据类型[] {元素 1, 元素 2, .....}

简化格式:

数据类型[] 数组名 = {元素 1, 元素 2, .....}

B: 案例演示

对数组的解释

输出数组名称和数组元素

C: 画图演示

```
public class Demo10 {  
    public static void main(String[] args) {  
        // 静态初始化  
        // 方式一  
        int[] a1 = new int[] {10, 20, 30, 40, 50};  
        System.out.println(a1[0]);  
        System.out.println(a1[1]);  
        System.out.println(a1[2]);  
        System.out.println(a1[3]);  
        System.out.println(a1[4]);  
        System.out.println("-----");  
        // 方式二:  
        int[] a2 = {12, 23, 34, 45, 56}; // 开发中推荐这么写  
        System.out.println(a2[0]);  
        System.out.println(a2[1]);  
        System.out.println(a2[2]);  
        System.out.println(a2[3]);  
    }  
}
```

```

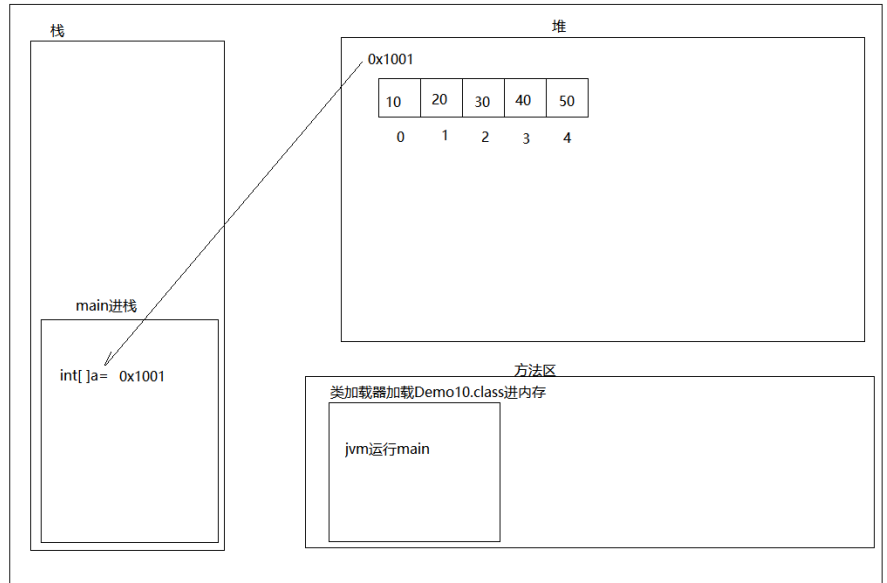
        System.out.println(a2[4]);
    }
}

```

```

public class Demo10 {
    public static void main(String[] args) {
        // 静态初始化
        // 方式一
        int[] a1 = new int[]{10,20,30,40,50};
        System.out.println(a1[0]);
        System.out.println(a1[1]);
        System.out.println(a1[2]);
        System.out.println(a1[3]);
        System.out.println(a1[4]);
    }
}

```



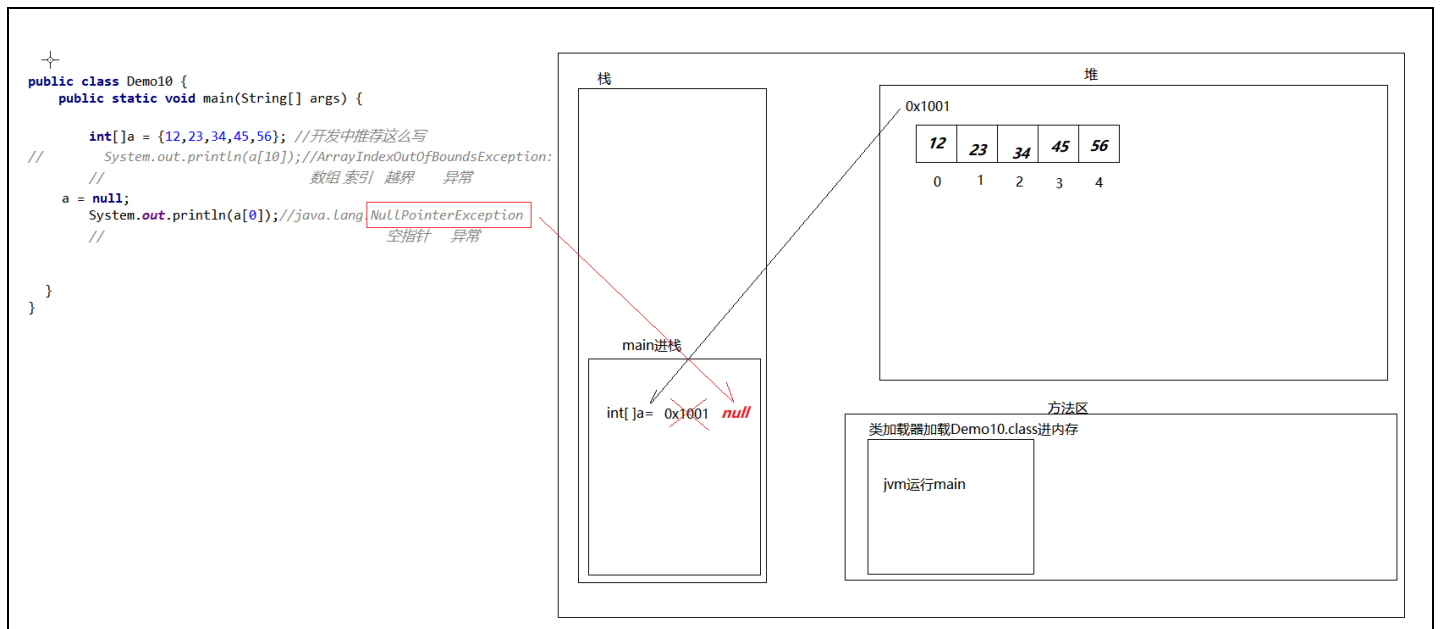
## 05\_Java 语言基础（数组操作的两个常见小问题：越界和空指针）（掌握）

```

public class Demo10 {
    public static void main(String[] args) {

        int[] a = {12,23,34,45,56}; // 开发中推荐这么写
        //      System.out.println(a[10]); // ArrayIndexOutOfBoundsException:
        //                                     数组 索引 越界 异常
        a = null;
        System.out.println(a[0]); // java.lang.NullPointerException
        //                                     空指针 异常
    }
}

```



## 05\_Java 语言基础（数组遍历）（掌握）

案例演示：

数组遍历：就是依次输出数组中的每个元素

数组的属性：`arr.length` 数组的长度

数组的最大索引：`length-1`

```

public class Demo10 {
    public static void main(String[] args) {

        int[] a = {12,23,34,45,56,67,78,89};
        /* System.out.println(a[0]);
        System.out.println(a[1]);
        System.out.println(a[2]);
        System.out.println(a[3]);
        System.out.println(a[4]);
        System.out.println(a[5]);
        System.out.println(a[6]);
        System.out.println(a[7]);*/
        for(int i=0;i<8;i++){
            System.out.print(a[i]+"\\t");
        }
        System.out.print("\\n");
        int len = a.length; // 可以获取数组的长度
        System.out.println("数组的长度="+len);
        for(int i=0;i<a.length;i++){
            System.out.print(a[i]+"\\t");
        }
        System.out.println();
        System.out.println("-----");
        //while 循环遍历数组
        int n=0;
        while(n<a.length){

```

```

        System.out.println(a[n]);
        n++;
    }
}
}

```

练习:

写一个方法专门用来遍历数组

```

public class Demo11 {
    public static void main(String[] args) {
        int[] a = {20,30,40,50,60,70,80};
        printArray(a);
    }
    //遍历数组的方法
    public static void printArray(int[] arr){ //等同于int[] arr = {20,30,40,50,60,70,80}
        for(int i=0;i<arr.length;i++){
            System.out.print(arr[i]+"\\t");
        }
    }
}

```

1. 数组中第二个空间的值为“李四”，第三个空间的值为“王二”，第四个空间的值为“麻子”，第五个空间的值为“小红”。然后再输出所有空间中的值

```

public class Demo11 {
    public static void main(String[] args) {
        /*
        1. 数组中第二个空间的值为“李四”，第三个空间的值为“王二”，第四个空间的值为“麻子”，
        第五个空间的值为“小红”。然后再输出所有空间中的值
        */
        String[] s = new String[5];
        s[1] = "李四";
        s[2] = "王二";
        s[3] = "麻子";
        s[4] = "小红";
        for(int i=0;i<s.length;i++){
            System.out.println(s[i]);
        }
    }
}

```

2. 请将' A' , ' B' , ' C' 存入数组，然后再输出

```

public class Demo11 {
    public static void main(String[] args) {
        /*
        'A','B','C'

```

```

        */
        char[] c = {'A','B','C'};
        for(int i=0;i<c.length;i++){
            System.out.println(c[i]);
        }
    }
}

```

3. 请将”我” “和” “我” “的” “祖” “国” 存入数组，然后正着和反着输出

```

public class Demo11 {
    public static void main(String[] args) {
        /*
        'A','B','C'
        */
        char[] c = {'我','和','我','的','祖','国'};
        for(int i=0;i<c.length;i++){
            System.out.println(c[i]);
        }
        System.out.println("-----");
        for(int i=c.length-1;i>=0;i--){
            System.out.println(c[i]);
        }
    }
}

```

案例：

数组的复制

```

public class Demo11 {
    public static void main(String[] args) {
        char[] c1 = {'我','和','我','的','祖','国'};
        char[] c2 = new char[c1.length]; // 创建一个新的数组长度为c1 数组的长度
        for(int i=0;i<c1.length;i++){
            c2[i] = c1[i];
        }
        System.out.println("-----");
        for(int i=0;i<c2.length;i++){
            System.out.print(c2[i]+"\\t");
        }
    }
}

```

案例：

获取数组中最大的那个元素



```

public class Demo11 {
    public static void main(String[] args) {
        // 获取数组中最大的那个元素
        int[] a = {3,5,7,10,2,41};
        int max = a[0];
        for(int i=1;i<a.length;i++){
            if(a[i]>=max){
                max = a[i];
            }
        }
        System.out.println("最大的元素是: "+max);
    }
}

```

## 擂台

**int max =** ~~3~~~~5~~~~7~~ **10**

**int[] a = {3,5,7,10,2,4};**

案例:

查找数组中的元素第一次出现时的索引

```

public class Demo11 {
    public static void main(String[] args) {
        int[] a = {3,5,2,7,10,2,41};
        int index = getIndex(a,7);
        System.out.println("索引是: "+index);
    }
    // 查找数组中的元素第一次出现时的索引

```

**/\*\*** 这是文档注释: 一般写在类上面或者方法上面, 用来解释类或方法的相关参数信息

```

* @param arr: 接收数组的形式参数
* @param a: 指定要获取下标索引的元素值
* @return: 返回索引
*/
public static int getIndex(int[] arr, int a){

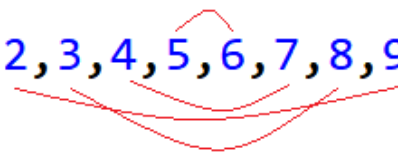
    for(int i=0; i<arr.length; i++){
        if(arr[i] == a){
            return i; // 返回索引, return 在返回值的同时会结束当前 getIndex 的方法, 所以下面那个
return -1; 没有机会执行
        }
    }

    return -1; // 加这个 return 的目的是, 因为 for 循环存在死循环的风险, 可能出现这个循环不会结束,
那么从整个程序来看, 就没有 return 语句了
}
}

```

案例:  
数组的反转

## 进行位置的交换

`int a[] = {2, 3, 4, 5, 6, 7, 8, 9};`  
  
 $a[i]$                        $a[a.length-1-i]$

交换的次数是数组长度的一半

```

public class Demo6 {
    public static void main(String[] args) {
        // 数组的反转
        int a[] = {2, 3, 4, 5, 6, 7, 8, 9, 10};
        for (int i = 0; i < a.length/2; i++) { // 注意交换的次数是 length 的一半
            int temp = a[i];
            a[i] = a[a.length-1-i];
            a[a.length-1-i] = temp;
        }
        for (int i = 0; i < a.length; i++) {
            System.out.print(a[i] + "\t");
        }
    }
}

```

```
}  
}
```

案例：

根据输入的索引查找对应的月份

```
public class Demo6 {  
    public static void main(String[] args) {  
        // 根据输入的索引查找对应的月份  
        String result =getMonth(2);  
        System.out.println(result);  
    }  
    public static String getMonth(int index){  
        String[]months = {"一月","二月","三月","四月","五月","六月","七月","八月","九月","十月",  
"十一月","十二月"};  
        return months[index-1];  
    }  
}
```

案例：

对数组排序

参加笔试之前看一下这个题，出现的概率比较高

```
public class Demo6 {  
    public static void main(String[] args) {  
        // 冒泡排序法  
        int a[]={5,4,3,7,1};  
  
        for(int j=0;j<a.length;j++) {  
            // 下面是交换一次  
            for (int i = 0; i < a.length - 1; i++) {  
                if (a[i] > a[i + 1]) {  
                    // 交换前后的位置  
                    int temp = a[i];  
                    a[i] = a[i + 1];  
                    a[i + 1] = temp;  
                }  
            }  
        }  
  
        for (int i = 0; i <a.length ; i++) {  
            System.out.print(a[i]+" ");  
        }  
    }  
}
```

### 冒泡排序法

```
int a[]={5,4,3,7,1};
```

5	4	3	7	1
0	1	2	3	4

第一次

<del>5</del> 4	<del>4</del> <del>5</del> 3	<del>3</del> 5	<del>7</del> 1	<del>1</del> 7
交换				

第二次

<del>4</del> 3	<del>3</del> 4	<del>5</del> 1	<del>1</del> 5	7
----------------	----------------	----------------	----------------	---

第三次

<del>4</del> 3	<del>1</del> 4	<del>4</del> 1	<del>1</del> 5	7
----------------	----------------	----------------	----------------	---

第四次

<del>1</del> 3	<del>1</del> 4	<del>4</del> 1	<del>1</del> 5	7
----------------	----------------	----------------	----------------	---

### Arrays 类（工具类）的学习

```
import java.util.Arrays;
```

```
public class Demo6 {
```

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

```
        int a[]={5,4,3,7,1};
```

```
        Arrays.sort(a);//对数组进行升序排序
```

```
        for (int i = 0; i < a.length ; i++) {
```

```
            System.out.print(a[i]+" ");
```

```
        }
```

```
        System.out.println();
```

```
        //根据指定的元素查找在数组中的索引
```

```
        /**
```

```
         * 第一个参数 - 要搜索的数组
```

```
         * 第一个参数 - 要搜索的值
```

```
        */
```

```
        int index = Arrays.binarySearch(a,5);//使用这个功能需要先对数组进行排序
```

```
        System.out.println(index);
```

```
        //把a 数组里面的元素全部用20 覆盖
```

```
        Arrays.fill(a,20);
```

```
        for (int i = 0; i < a.length ; i++) {
```

```

        System.out.print(a[i]+" ");
    }
    System.out.println();
    //把数组a 中从索引0 开始到索引3-1 位置上的数用10 覆盖
    Arrays.fill(a,0,3,10);
    for (int i = 0; i < a.length ; i++) {
        System.out.print(a[i]+" ");
    }
    //把数组a 中从索引4 开始到索引5-1 位置上的数用50 覆盖
    Arrays.fill(a,4,5,50);
    System.out.println();
    //    for (int i = 0; i < a.length ; i++) {
    //        System.out.print(a[i]+" ");
    //    }
    String result =Arrays.toString(a); //把数组转成字符串的表现形式
    System.out.println(result);

    int b[] ={55,43,33,77,11}; //int[]b 等同于int b[]
    int[]c = Arrays.copyOf(b,a.length); //第一个参数是要被拷贝的数组，第二个参数是要拷贝的数
    组长
    String re = Arrays.toString(c); //把数组转成字符串的表现形式
    System.out.println(re);
    int[]d = Arrays.copyOf(b,3); //第一个参数是要被拷贝的数组，第二个参数是要拷贝的数组长度
    System.out.println("d="+Arrays.toString(d));
}
}

```

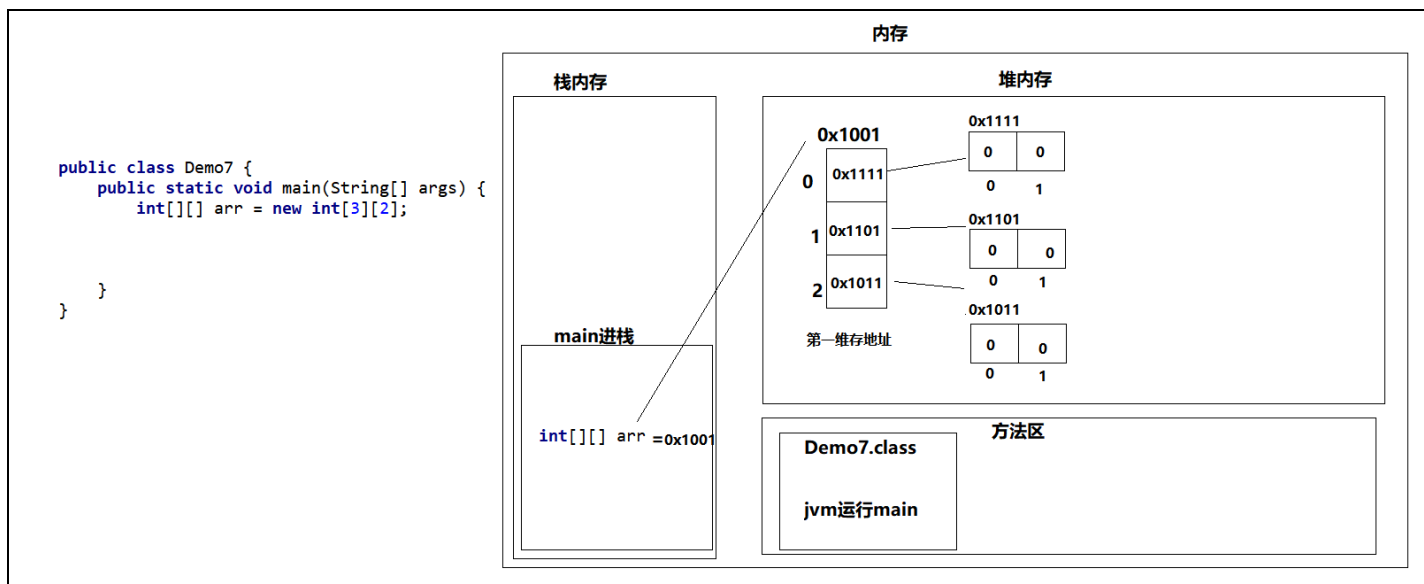
```

public class Demo6 {
    public static void main(String[] args) {
        int a[]={5,4,3,7,1};
        //增强for 循环
        for(int i:a){ //    : 的左边是数组里面元素的类型的变量用来接收：右边数组中每一个元素；i 可
        以随便取名
            System.out.println(i);
        }
    }
}

```

## 05\_Java 语言基础（二维数组）（掌握）

1. 二维数组的概述
2. 二维数组的格式
  - int[][] arr = new int[3][2];
3. 二维数组的格式解释



```
public class Demo7 {
    public static void main(String[] args) {
        int[][] arr = new int[3][2];
        System.out.println(arr); // [[I@10f87f48
        /*
         * [[I@10f87f48
         * [[ : 2 维
         * I : int 类型
         * @: @
         * 10f87f48 : 十六进制的内存地址
         */
        // 下面打印出来是三个一维数组的内存地址
        System.out.println(arr[0]);
        System.out.println(arr[1]);
        System.out.println(arr[2]);
        // 输出
        System.out.println(arr[0][0]);
        System.out.println(arr[0][1]);
        System.out.println(arr[1][0]);
        System.out.println(arr[1][1]);
        System.out.println(arr[2][0]);
        System.out.println(arr[2][1]);
    }
}
```

#### 4. 注意事项

一下格式也可以表示二维数组

- 1) 数据类型 数组名[][] = new 数据类型[m][n]; `int[][] arr = new int[3][2]`
- 2) 数据类型[][] 数组名 = new 数据类型[m][n]; `int arr[][] = new int[3][2]`
- 3) 数据类型[] 数组名[] = new 数据类型[m][n];

注意下面定义的区别

```
int x;
int y;
int x, y; //连续在同一行定义变量(开发中避免这么写)
```

```
int [] y[]; //二维数组
```

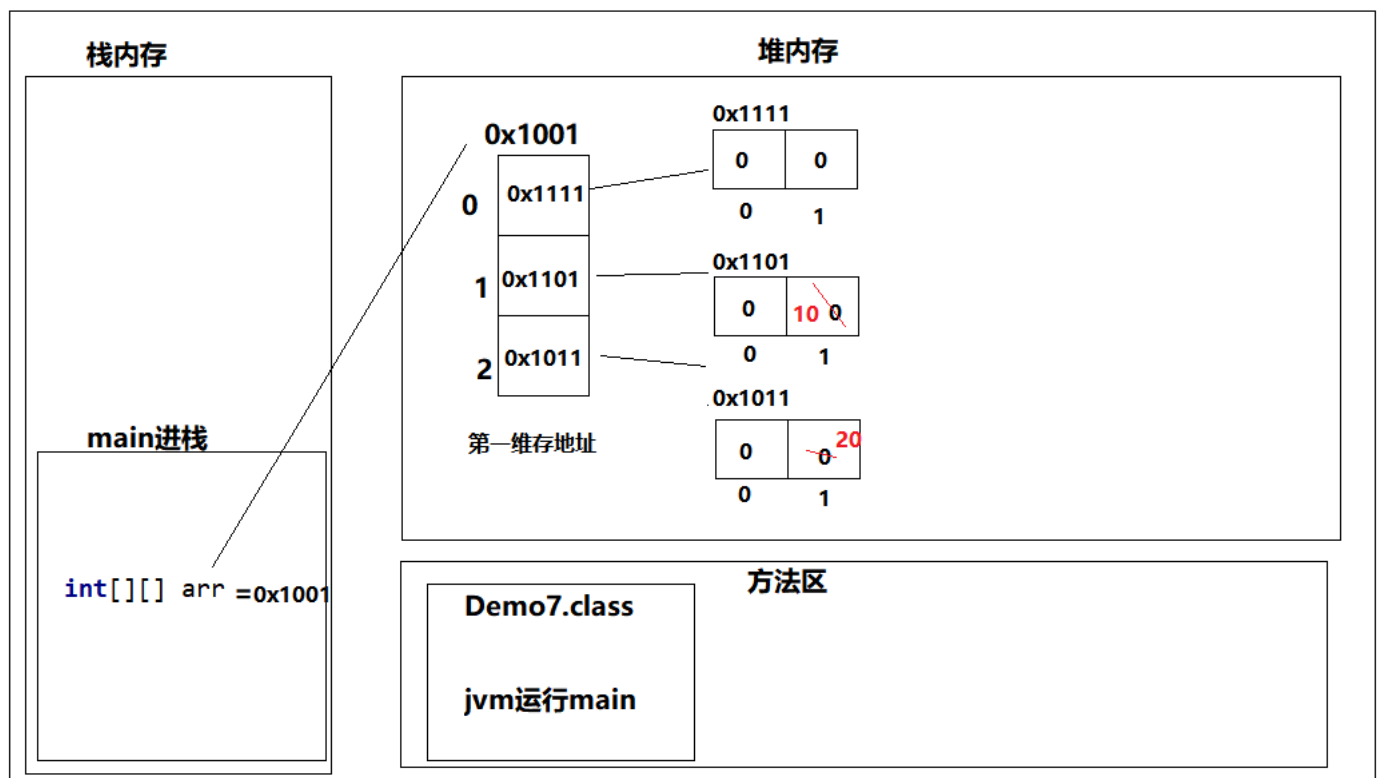
int []x, y[]; 等同于 int[]x; int[][]y

定义二维数组，输出二维数组名称，一维数组名称，一个元素

```
public class Demo7 {
    public static void main(String[] args) {
        int[][] arr = new int[3][2];
        System.out.println(arr);

        // 输出
        System.out.println(arr[0][0]);
        System.out.println(arr[0][1]);
        System.out.println(arr[1][0]);
        System.out.println(arr[1][1]);
        System.out.println(arr[2][0]);
        System.out.println(arr[2][1]);

        arr[1][1] = 10;
        System.out.println(arr[1][1]);
        arr[2][1] = 20;
    }
}
```



二维数组格式 2:

```
int[][] arr = new int[3][];
```

二维数组格式 2 的解释

案例演示

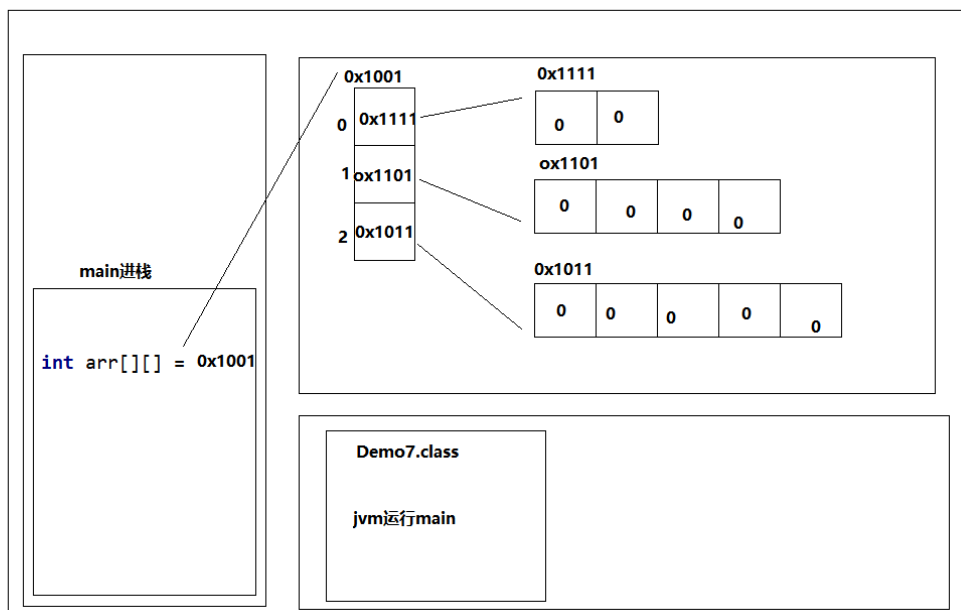
讲解格式，输出数据，并画内存图

案例:

画出二维数组的内存图

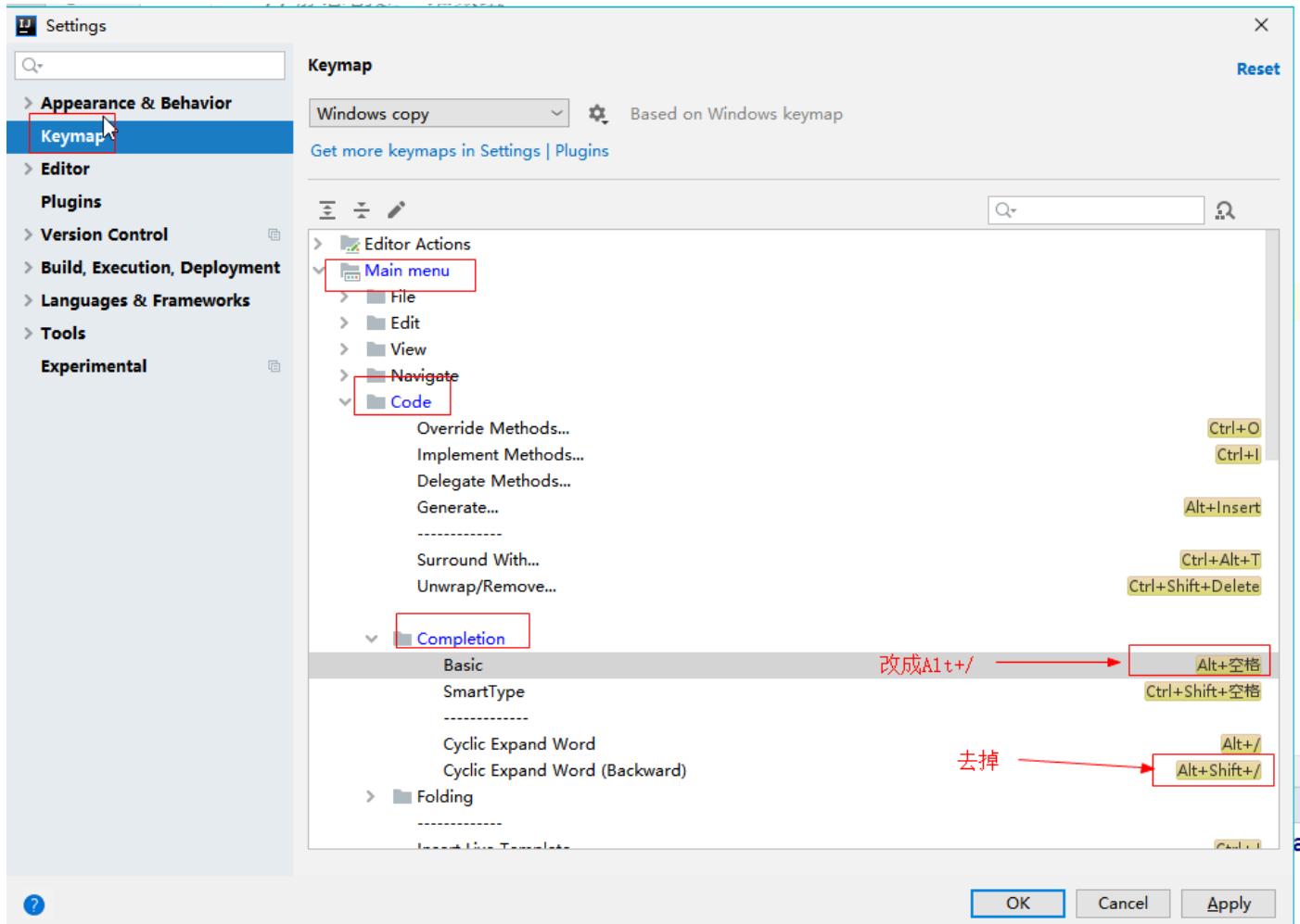
```
int[][] arr = {{2,3,4},{5,6,7,8},{11,22,33,44,55}};
```

```
public class Demo7 {  
    public static void main(String[] args) {  
        int arr[][] = new int [3][]; // 在创建二维数组的时候指定了第一维数组的长度  
        arr[0] = new int[2];  
        arr[1] = new int[4];  
        arr[2] = new int[5];  
    }  
}
```



`int arr2 = new int[][3];` // 这么写是错误的写法





案例:

二维数组的遍历

```
public class Demo7 {
    public static void main(String[] args) {
        // 静态创建二维数组
        int[][] a = {{2,3,4},
                    {5,6,7},
                    {8,9}};
        System.out.println(a[1][2]);
        System.out.println(a[2][0]);
        System.out.println("-----");
        // 遍历
        for (int i = 0; i < a.length; i++) { // 第一维获取的是第二维的内存地址
            // System.out.println(a[i].length);
            for(int j=0;j<a[i].length;j++){ // 遍历二维数组中的值
                System.out.println(a[i][j]);
            }
        }
    }
}
```

案例演示：

稀有金属冶炼厂生产车间年产量的求和

按照季度和月份的统计数据：

一季度：20,50,40

二季度：56,78,43

三季度：25,65,54

四季度：43,50,30

```
public class Demo7 {
    public static void main(String[] args) {
        /*
        一季度：20,50,40
        二季度：56,78,43
        三季度：25,65,54
        四季度：43,50,30
        */
        int [][]a = {{20,50,40},
                     {56,78,43},
                     {25,65,54},
                     {43,50,30}};
        //求年产量的总和
        int sum=0;//申明一个接收年总产量的变量
        for (int i = 0; i <a.length ; i++) {
            for (int j = 0; j <a[i].length ; j++) {
                //获取到每个月的产量
                int t = a[i][j];
                sum += t;
            }
        }
        System.out.println("年总产量是: "+sum);

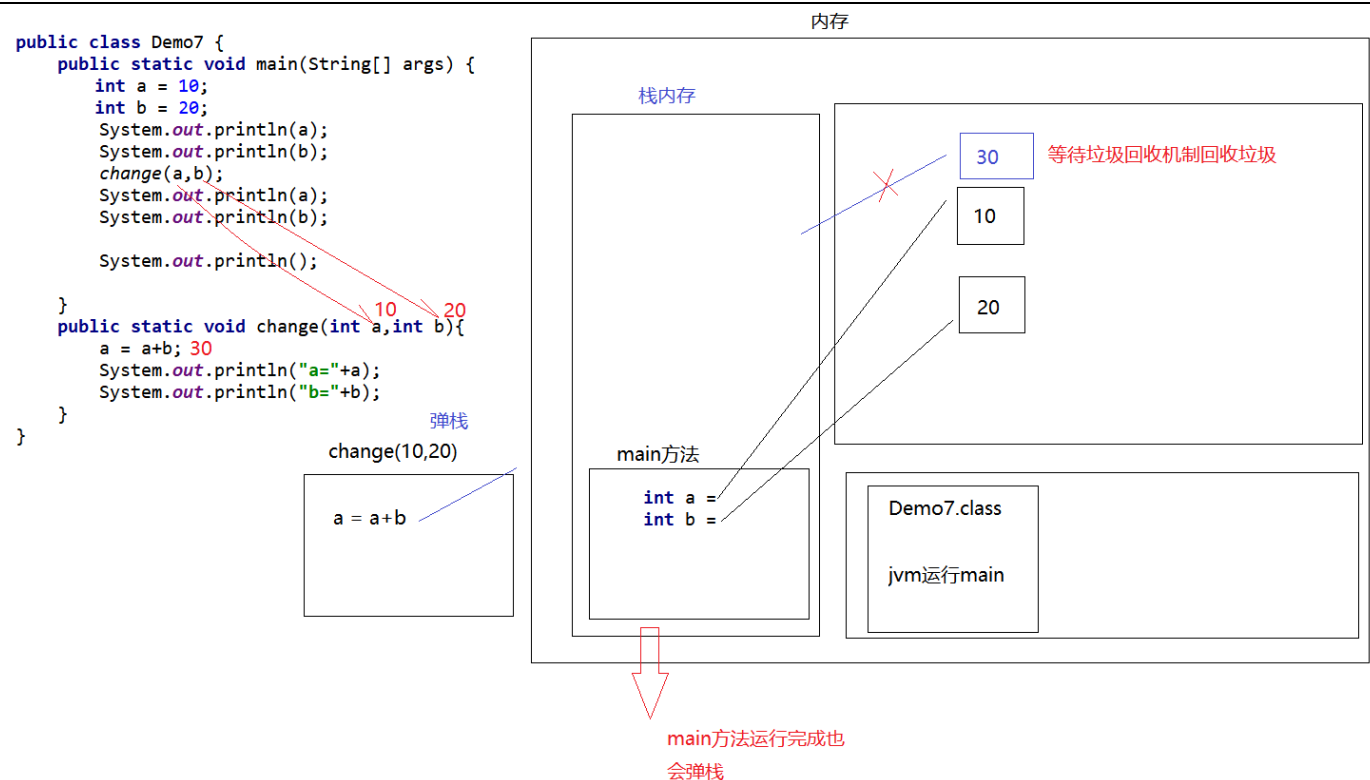
        //如果要求第三季度的产量总和
        int sum2 = 0;
        for (int j = 0; j <a[2].length ; j++) {
            //获取到每个月的产量
            int t = a[2][j];//第三个季度的第一维在a[2][]
            sum2 += t;
        }
        System.out.println("第三季度的产量和: "+sum2);
    }
}
```

二维数组在开发中用的很少，了解

案例：

问：a 和 b 的值被改变了没有？

```
public class Demo7 {  
    public static void main(String[] args) {  
        int a = 10;  
        int b = 20;  
        System.out.println(a);  
        System.out.println(b);  
        change(a,b);  
        System.out.println(a);  
        System.out.println(b);  
    }  
    public static void change(int a,int b){  
        a = a+b;  
    }  
}
```



arr 里面的元素会改变吗？

```
public class Demo {  
    public static void main(String[] args) {  
        int[] arr = {5,7,8,9,11};  
        change(arr);  
        for (int i = 0; i < arr.length; i++) {  
            System.out.print(arr[i]+" ");  
        }  
    }  
}
```

```

public static void change(int[] a) {
    a[0]*=3;
    a[1]+=3;
}
}

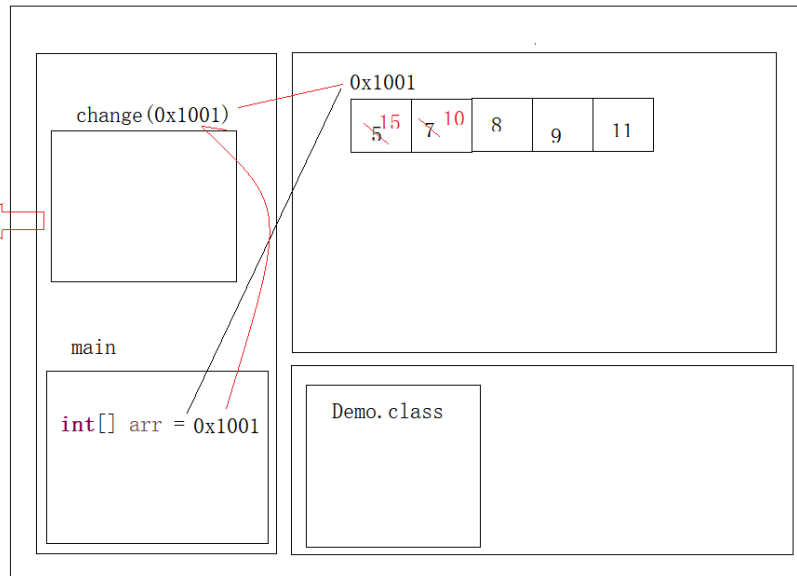
```

```

public class Demo {
    public static void main(String[] args) {
        int[] arr = {5, 7, 8, 9, 11};
        change(arr);
        for (int i = 0; i < arr.length; i++) {
            System.out.print(arr[i]+" ");
        }
    }
    public static void change(int[] a) {
        a[0]*=3;
        a[1]+=3;
    }
}

```

弹栈



结论:

基本数据类型传递的是值

应用数据类型传递的是内存地址

面试问题: java 中到底是传值还是地址呢?

java 之父“急死你, 搞死你”说: 基本数据类型的值和内存地址, 都是值, 所以 java 传的都是值