

Java Day 05

数组基础

概念：存放相同数据类型数据的集合

Java 中的数组是对象，有长度和数组成员

声明数组

1) 数据类型[] 数组名;

```
int[] arr;
```

2) 数据类型 数组名[];

```
int arr[];
```

创建数组

1) 数据类型[] 数组名 = new 数据类型[长度];

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

2) 数据类型[] 数组名 = new 数据类型[] {元素 1, 元素 2, 元

素 3...};

```
int[] arr = new int[] {1, 2, 3, 4, 5};
```

3) 数据类型[] 数组名 = {元素 1, 元素 2, 元素 3...};

```
int[] arr = {1, 2, 3, 4, 5};
```

可以存放基本数据类型，也可以存放类类型

数组分配好堆内存后, 会对成员进行默认初始化 (数值类型为 0, boolean 类型为 false, 引用类型为 null)

访问数组

索引: 每一个存储到数组的元素, 都会自动的拥有一个编号, 从 0 开始, 这个自动编号称为数组索引(index), 可以通过数组的索引访问到数组中的元素。

数组名[索引]

数组名[索引]=数值, 为数组中的元素赋值

变量=数组名[索引], 获取出数组中的元素

数组动态初始化: new 分配内存, [] 访问数组成员赋值, 声明、赋值可以分开写

数组静态初始化: {} 直接分配内存、赋值, 声明、赋值不能分开写

数组长度

每个数组都具有长度, 而且是固定的, Java 中赋予了数组的一个属性, 可以获取到数组的长度, 语句为: 数组名.length。属性 length 的执行结果是数组的长度, int 类型, 数组的最大索引值为数组名.length-1。

数组下标访问时，可以使用变量。

数组访问时，编译器不会检查下标范围，由程序员自己负责，容易犯错。

数组遍历

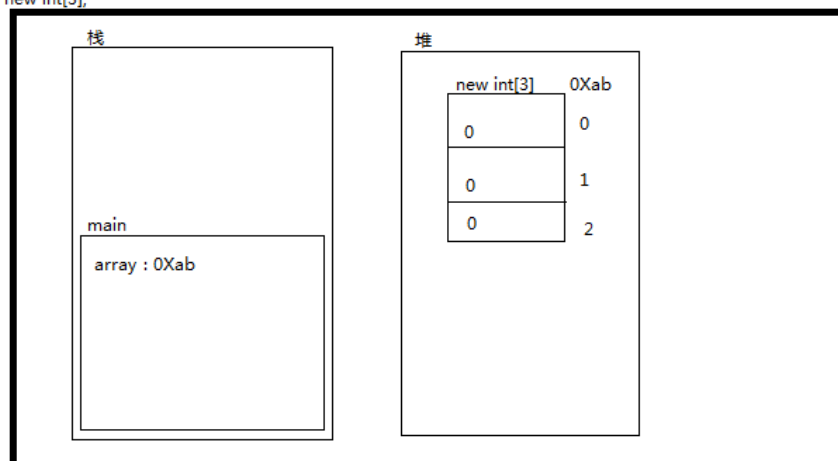
就是将数组中的每个元素分别获取出来

数组内存图

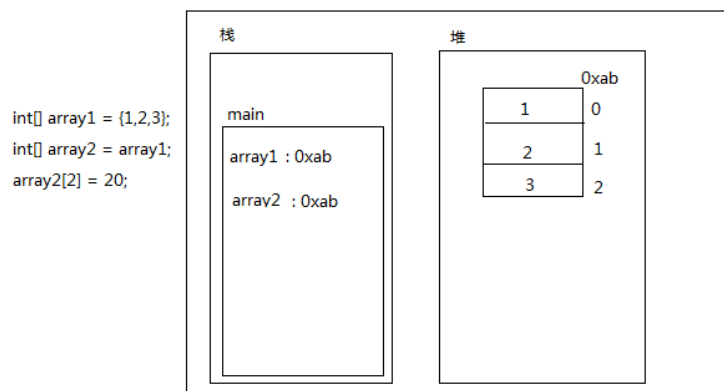
一块连续的内存空间，查找速度快

一个数组内存图

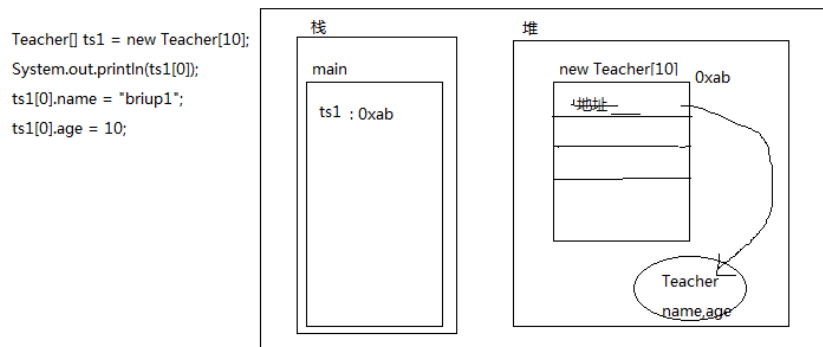
```
int[] array = new int[3];
```



两个变量指向一个数组



数组中存放类类型



数组常见异常

ArrayIndexOutOfBoundsException

数组下标越界

编译器不进行数组下标范围检查，需要程序员自己检查

NullPointerException

空指针

调试错误：

编译时错误

查看报错信息，定位错误所在

添加删除程序法（注释）

运行时错误

查看报错信息，定位错误所在

添加删除程序法（注释）

跟踪程序执行流程：

大范围跟踪：

```
System.out.println("1000");  
System.out.println("1001");  
System.out.println("2000");
```

小范围跟踪：

```
System.out.println(变量)
```

数组作为方法参数和返回值

作为参数：传递的参数是数组内存的地址

```
public static void main(String[] args) {  
    int[] arr = { 1, 3, 5, 7, 9 };  
  
    //调用方法，传递数组  
    printArray(arr);  
}
```

作为返回值：返回的是数组的内存地址

```
public static void main(String[] args) {  
    //调用方法，接收数组的返回值  
  
    //接收到的是数组的内存地址  
    int[] arr = getArray();  
    for (int i = 0; i < arr.length; i++) {  
        System.out.println(arr[i]);  
    }  
}
```