传智播客 www.itcast.cn —高级软件人才实作培训专家！

Java语言基础

毕向东

2.7 函数

- 函数的定义
- 函数的特点
- 函数的应用
- 函数的重载

2.7.1 函数的定义

- 什么是函数？
  - 函数就是定义在类中的具有特定功能的一段独立小程序。
  - 函数也称为方法。

Java中最小的功能单位是函数

- 函数的格式：
  - 修饰符 返回值类型 函数名(参数类型 形式参数1，参数类型 形式参数2，)

```
{
        执行语句;
        return 返回值;
}
```

返回值类型：函数运行后的结果的数据类型。
参数类型：是形式参数的数据类型。
形式参数：是一个变量，用于存储调用函数时传递给函数的实际参数。
实际参数：传递给形式参数的具体数值。
return：用于结束函数。
返回值：该值会返回给调用者。

2.7.2 函数的特点

- 定义函数可以将功能代码进行封装
- 便于对该功能进行复用
- 函数只有被调用才会被执行
- 函数的出现提高了代码的复用性
- 对于函数没有具体返回值的情况，返回值类型用关键字void表示，那么该函数中的return语句如果在最后一行可以省略不写。
- 注意：
  - 函数中只能调用函数，不可以在函数内部定义函数。
  - 定义函数时，函数的结果应该返回给调用者，交由调用者处理。

2.7.3 函数的应用

- 两个明确
  - 明确要定义的功能最后的结果是什么？
  - 明确在定义该功能的过程中，是否需要未知内容参与运算
- 示例：
  - 需求：定义一个功能，可以实现两个整数的加法运算。
  - 分析：
    - 该功能的运算结果是什么？两个数的和，也是一个整数(int)
    - 在实现该功能的过程中是否有未知内容参与运算？加数和被加数是不确定的。(两个参数int，int)
    - 代码：

```
int  getSum(int x,int y)
{
        return x+y;
}
```

2.7.4 函数的重载(overload)

重载的概念
在同一个类中，允许存在一个以上的同名函数，只要它们的参数个数或者参数类型不同即可。
重载的特点：
与返回值类型无关，只看参数列表。
重载的好处：
方便于阅读，优化了程序设计。
重载示例：

```
//返回两个整数的和
  int add(int x,int y){return x+y;}
//返回三个整数的和
  int add(int x,int y,int z){return x+y+z;}
//返回两个小数的和
  double add(double x,double y){return x+y;}
```

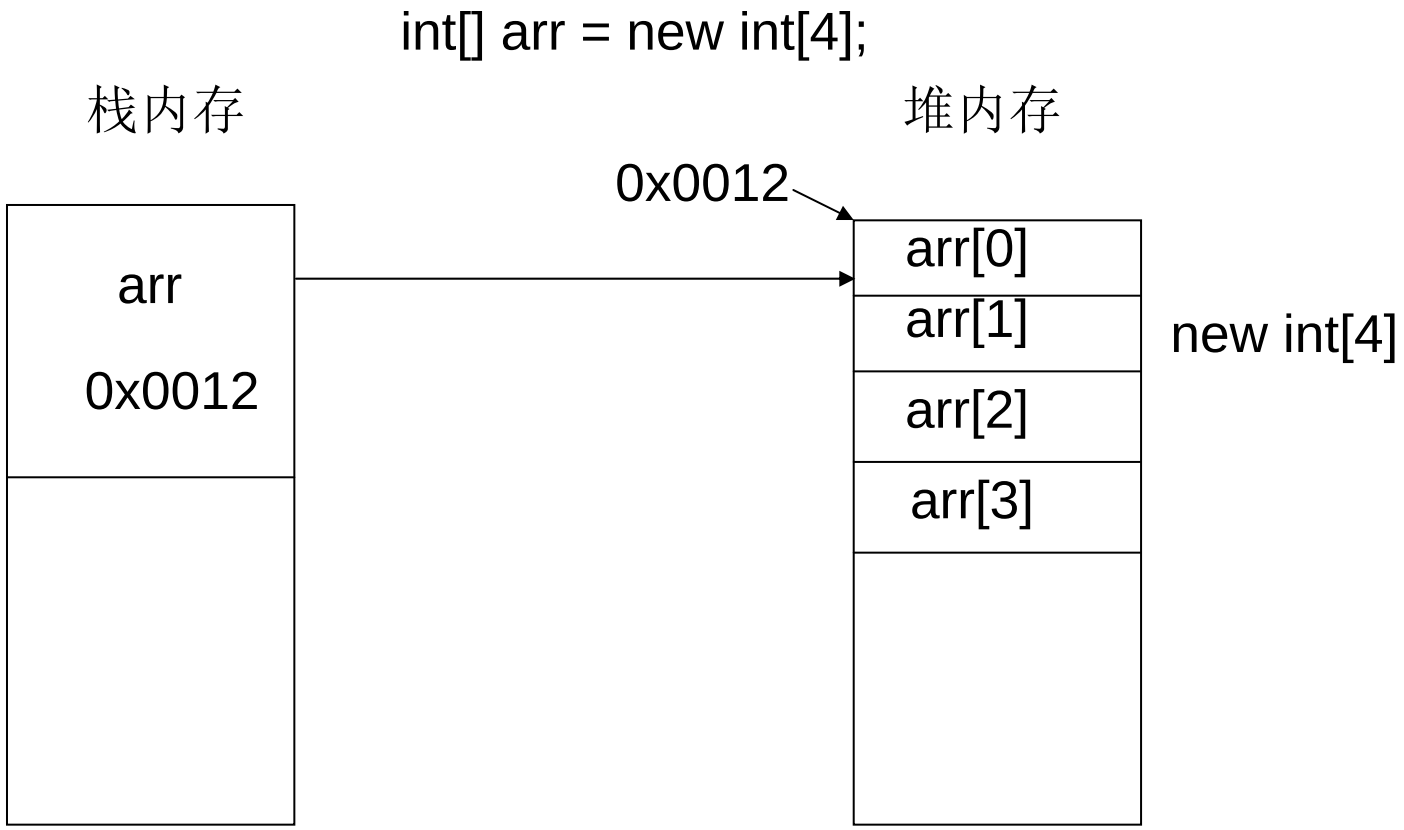
2.8 数组

- 数组的定义
- 数组的内存分配及特点
- 数组操作常见问题
- 数组常见操作
- 数组中的数组

2.8.1 数组的定义

概念
同一种类型数据的集合。其实数组就是一个容器。
数组的好处
可以自动给数组中的元素从0开始编号，方便操作这些元素。
格式1：
元素类型[] 数组名 = new 元素类型[元素个数或数组长度]; 示例：int[] arr = new int[5];
格式2：
元素类型[] 数组名 = new 元素类型[]{元素，元素，……}; int[] arr = new int[]{3,5,1,7}; int[] arr = {3,5,1,7};

2.8.2 数组内存结构


int[] arr = new int[4];
栈内存
堆内存
0x0012
arr
0x0012
arr[0]
arr[1]
arr[2]
arr[3]
new int[4]

2.8.2 内存结构

Java程序在运行时，需要在内存中的分配空间。为了提高运算效率，有对空间进行了不同区域的划分，因为每一片区域都有特定的处理数据方式和内存管理方式。

栈内存

- 用于存储局部变量，当数据使用完，所占空间会自动释放。

堆内存

- 数组和对象，通过new建立的实例都存放在堆内存中。
- 每一个实体都有内存地址值
- 实体中的变量都有默认初始化值
- 实体不在被使用，会在不确定的时间内被垃圾回收器回收

方法区，本地方法区，寄存器

2.8.3 数组操作常见问题

数组脚标越界异常(ArrayIndexOutOfBoundsException)

int[] arr = new int[2];
System.out.println(arr[3]); //数组角标越界
访问到了数组中的不存在的脚标时发生。

空指针异常(NullPointerException)
int[] arr = null; System.out.println(arr[0]); **arr**引用没有指向实体，却在操作实体中的元素时。

2.8.4 数组常见操作

- 获取最值(最大值，最小值)
- 排序（选择排序，冒泡排序）
- 折半查找(二分查找)

2.8.5 数组中的数组

二维数组[][]

格式1：int[][] arr = new int[3][2];

- 定义了名称为arr的二维数组
- 二维数组中有3个一维数组
- 每一个一维数组中有2个元素
- 一维数组的名称分别为arr[0], arr[1], arr[2]
- 给第一个一维数组1脚标位赋值为78写法是：arr[0][1] = 78;

格式2：int[][] arr = new int[3][];

- 二维数组中有3个一维数组
- 每个一维数组都是默认初始化值null
- 可以对这个三个一维数组分别进行初始化

```
arr[0] = new int[3];
 arr[1] = new int[1];
 arr[2] = new int[2];
```

2.8.5 数组中的数组

格式3：int[][] arr = {{3,8,2},{2,7},{9,0,1,6}};

- 定义一个名称为arr的二维数组
- 二维数组中的有三个一维数组
- 每一个一维数组中具体元素也都已初始化
- 第一个一维数组 arr[0] = {3,8,2};
- 第二个一维数组 arr[1] = {2,7};
- 第三个一维数组 arr[2] = {9,0,1,6};
- 第三个一维数组的长度表示方式：arr[2].length;

练习：获取arr数组中所有元素的和。使用for的嵌套循环即可。

注意特殊写法情况：int[] x,y[]; x是一维数组，y是二维数组。

练习

- 基础练习题
- 进制转换
- 幸运儿