基础[3]

- 【一】数组
- 【二】方法

一、数组

目录

- 【1】概念
- 【2】声明与赋值
- 【3】组成
- 【4】遍历
- 【5】排序
- 【6】注意

1、概念

概念

> 数组就是用来存储一批同种类型数据的内存区域（数据长度固定的容器）

2、声明与赋值

声明与赋值

2.1、静态初始化

2.1.1、无省略格式

- 格式

```
数据类型[] 数组名 = new 数据类型[]{arr1,arr2,arr3....};
```

- 举例

```
int[] arr = new int[]{1,2,3};
//or
int[] arr;
arr = new int[]{1,2,3};
```

2.1.2、省略格式

- 格式

```
数组类型 [] 数组名 = {arr1,arr2...}
```

- 举例

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

2.2、动态初始化

- 格式

```
数组类型[]  名字 =new 数组类型[数组长度];
```

- 举例

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

3、组成

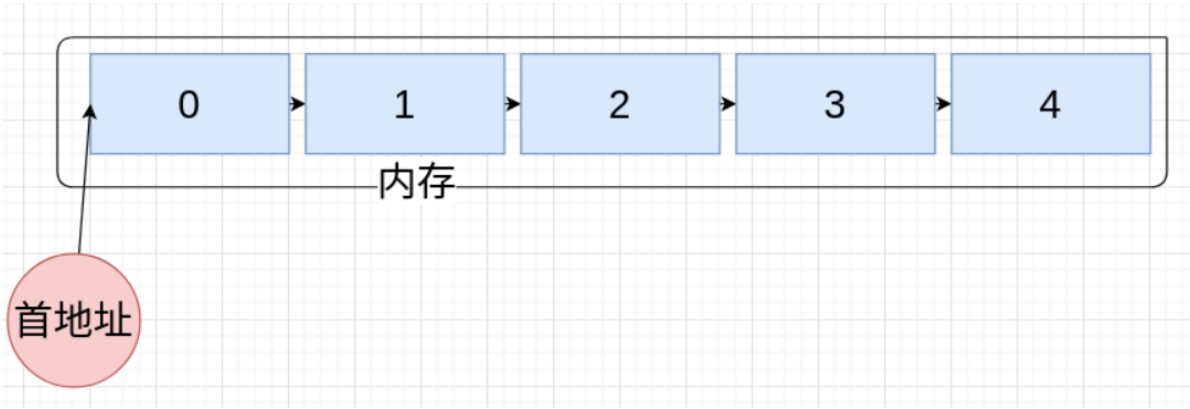
组成

- 数组中的每个数据被称为数组元素
- 通过下标访问值
- 对每个元素进行赋值或取值的操作被称为“元素的访问“
- 数组中的每个数据被称为

4、遍历

遍历

```
String[] array;
array=new String[]{"h","e","l","l","o"};
for (String string : array) {
    System.out.println(string);
}
```



5、排序

排序

- 选择排序

  > 直接选择排序、堆排序

- 交换排序

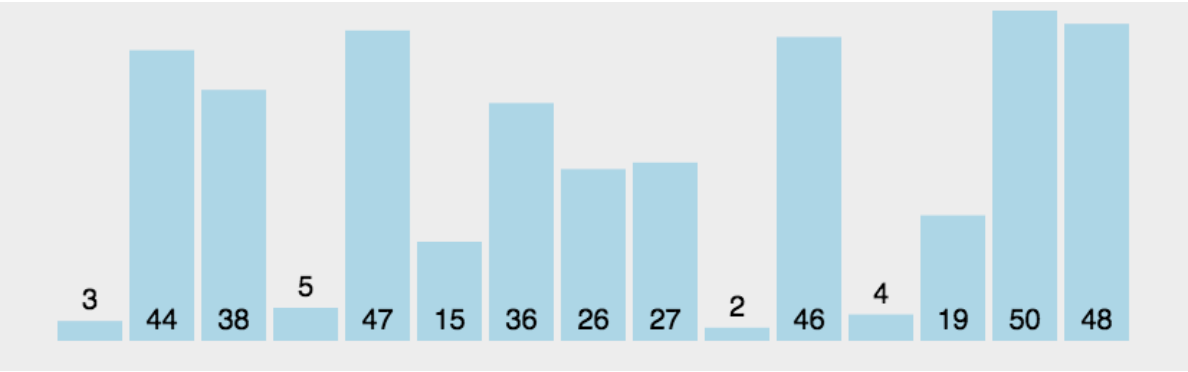
  > 冒泡排序、快速排序

- 插入排序

  > 直接插入排序、折半插入排序、Shell排序

- 归并排序
- 桶式排序
- 基数排序

冒泡排序




```
for (int i = 0; i < arr.length; ++i) {
    for (int j = 0; j < arr.length - 1 - i; ++j) {
        if (arr[j] > arr[j + 1]) {
            int temp = arr[j];
            arr[j] = arr[j + 1];
            arr[j + 1] = temp;
        }
    }
}
```

其他排序方法及图片请看这里

6、注意

注意

- 防止数组空指针异常
- 防止数组越界

二、方法

目录

- 【1】概念
- 【2】定义
- 【3】组成
- 【4】内存
- 【5】调用
- 【6】方法重载

1、概念

概念

> 方法是一种语法格式，他可以把一段代码封装成一个功能，以方便重复调用。
>
> - 提高代码复用性
> - 程序逻辑更清晰

2、定义

定义

> 方法定义后需要去调用使用。`main`由`JVM`自己调用

`格式`

```
[修饰符]  返回值类型  方法名(形式参数列表){
    语句体;
}
```

`示例`

```
public static int Add(int a,int b){
    int temp=a+b;
    return temp;
}

int num = Add(20,20);
```

3、组成

组成

- 参数

  > `形式参数列表`，涉及方法重载

  - 形似参数全部都是**局部变量**，进入栈区，方法结束自动释放。
  - 形参的个数和类型很重要，**关于方法重载**。
- 返回值
  - 无返回值 `void`

    ```
    return;
    //停止函数，无返回值
    ```
  - 有返回值

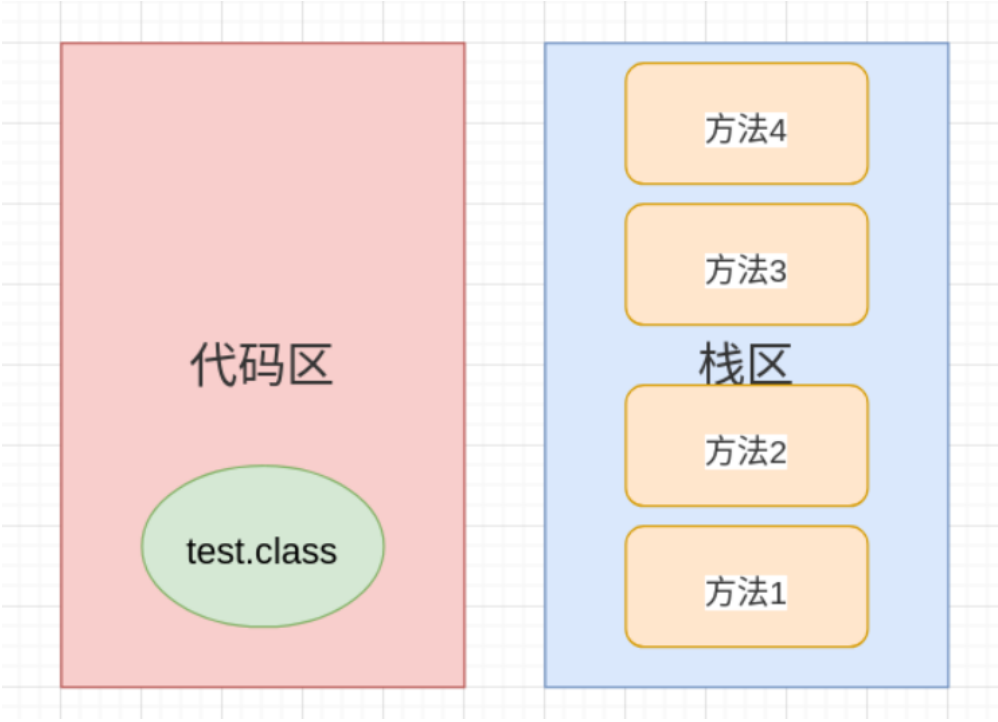
    ```
    return [type];
    //停止函数，返回值
    ```
- 方法名

  > 驼峰命名规则

4、内存

内存

> - 方法没有调用前，在方法区的字节码中
> - 调用后，进入到栈内存



5、调用

调用

- 静态方法：`static`

```
类名.方法名(形参....);
//or
类名  xxx  = new 类名();
xxx.方法名(形参...);
```

- 普通方法

```
类名  xxx  = new 类名();
xxx.方法名(形参...);
```

6、方法重载

方法重载

6.1、方法重载概念

方法重载指同一个类中定义的多个方法之间的关系，满足下列条件的多个方法相互构成重载

- 多个方法在同一个类中
- 多个方法具有相同的方法名
- 多个方法的参数不相同，类型不同或者数量不同

6.2、注意：

- 重载仅对应方法的定义，与方法的调用无关，调用方式参照标准格式
- **重载仅针对同一个类中方法的名称与参数进行识别，与返回值无关**，换句话说不能通过返回值来判定两个方法是否相互构成重载

6.3、正确范例：

```
public class MethodDemo {
    public static void fn(int a) {
        //方法体
    }
    public static int fn(double a) {
        //方法体
    }
}

public class MethodDemo {
    public static float fn(int a) {
        //方法体
    }
    public static int fn(int a , int b) {
        //方法体
    }
}
```

6.4、错误范例：

```
public class MethodDemo {
    public static void fn(int a) {
        //方法体
    }
    public static int fn(int a) {   /*错误原因：重载与返回值无关*/
        //方法体
    }
}

public class MethodDemo01 {
    public static void fn(int a) {
        //方法体
    }
}
public class MethodDemo02 {
    public static int fn(double a) { /*错误原因：这是两个类的两个fn方法*/
        //方法体
    }
}
```