



每日一讲

史肖敏



JAVA基础

- 1.变量的作用域:

从变量的声明开始到包含它最近的大括号结束

- 2.变量的重命名问题: 作用域同名的变量不能同名

```
int a = 1;
```

```
if(a>2){
```

```
    int b = 1;
```

```
    int a = 5;//此处编译出错，作用域重叠
```

```
}else{
```

```
    int b = 2;
```

```
}
```

1.while(循环条件){

}

3.循环三要素:

1.循环变量的初始化

2.循环条件（以循环变量为基础）

3.循环变量的改变（向循环结束变）

循环变量：在循环过程中所改变的那个量

特点：（适用情况）?????????

2.do{

}while(循环条件);

特点：（适用情况）?????????

3.for:

特点：（适用情况）?????????

4.for-each:

特点：（适用情况）?????????

PS：关键找循环三要素！！！！

4.生成随机数:

```
Math.random();//0.0-0.9999.....  
Random rand = new Random();  
变量 = rand.nextInt(20);//0-19
```

5.break:

用于循环，常常和条件配合使用，通常只能跳出一层循环

continue:

跳出循环题中剩余语句而进入下一次循环

6.嵌套循环的执行原理: ? ? ? ? ? ? ? ?

7.程序=算法+数据结构

算法=? -----解决问题的流程/步骤-----顺序、分支、循环

数据结构=? --将数据按某种特定的结构来保存

设计良好的数据结构会导致良好的算法

8.数组： 即一种基本的数据结构

1.相同的数据类型元素的集合

2.也是一种数据类型，但不是基本的数据类型，而是一种引用类型

3.按现行顺序排列，第一个元素除外，每个元素都有一个唯一的前驱
最后一个元素除外，没个元素都有一个唯一后继

数组的访问? ? ? ? ?

9.数组的复制:

1.System.arraycopy(a,1,a1,0,4)

a-----源数组

1-----源数组起始下表

a1-----目标数组

0-----目标数组起始下表

4-----复制数据个数

PS: 数组的扩容

int[] a = {};//若不写, 则默认为0个元素

a = Arrays.copyOf(a,a.length+1);

a--目标数组 a--源数组 a.length--目标数组个数

给数组中最后一个元素赋值? ? ? ? ? ? ?

10.数组的排序

常用的有? ? ? ? ? ? ?

冒泡工作原理:

- 1.四个数冒三轮
- 2.每一轮都是从第一个元素开始冒（比）
每次都是和它的下一个元素比
- 3.冒出来的就不待它玩了（不再参与比较了）

质数：又称素数，只能被1和它本身整除的数

- 1.是质数：区域所有都不得0
- 2.不是质数：只要有1个得0，一定不是质数

开关的应用条件：并非一次判断就能的到最终结果

一、方法的5要素:

- 1.修饰词
- 2.返回类型
- 3.方法名
- 4.参数列表
- 5.方法体

修饰符	本类	同包类	子类	其他类
public	√	√	√	√
protected	√	√	√	×
默认	√	√	×	×
private	√	×	×	×

二、参数列表

- 1.方法可以有参、也可以无参，小括号不能生
- 2.有参可以使方法更灵活
- 3.多个参数用逗号隔开

三、方法体

里面封装着特定的功能实现

return:

- 1.终止方法的执行
- 2.返回结果给调用方（声明变量保存调用结果并输出）

面向对象

面向过程的缺陷：

1. 缺乏对数据的封装
2. 数据和方法的分离（操作数据）

数据结构：

1. 基本数据类型-----1个数
2. 数组-----多个数
3. 自己造一个数据类型-----多个数（类型不同）

1.什么是类？ 什么是对象？

1>现实世界是由很多很多对象组成的||基于对象抽出来类

2>对象：真实存在的单个个体

类：代表一类个体，类型、类别

3>类中可包含：

a.所有对象所共有的特征/属性-----变量

b.-----行为-----方法

4>1个类可以创建多个对象

同一个类的对象，结构相同，数据不同

（对象能点出来的东西一样）

5>类是对象的模板，对象是类的具体实例

基本类型与引用类型之间画等号

基本类型之间画等号

1.赋值

2.对其中一个变量的修改不会影响另外一个

引用类型之间画等号

1.指向同一个对象

2.对其中一个引用类型的修改会影响另外一个

null与NullPointerException

null:空，意为不再指向任何对象

若引用的值为null,则不再进行任何操作了

若引用后运行，则会上报NullPointerException异常

PS: 引用中装的都是地址

方法的签名=方法名+参数列表

方法的重载:

1.在同一个类中，不可以有两个方法的签名完全相同，即一个类中不可有2个方法名和参数列表完全一样

2.JAVA中允许多个方法名相同，但参数列表不同-----方法的重载

1.在同一个类中，方法名相同，参数列表不同

2.编译器在编译时会自动根据签名绑定不同的方法

构造方法:

1.常常用于给成员变量赋初值

2.与类同名，没有返回值

3.在创建new对象时，自动调用构造方法

4.若用户不写构造，则编译器默认提供一个无参的构造方法

若自己写了构造，则编译器不再默认提供构造方法

5.构造方法可以重载

This: 只带当前对象，哪个对象调用指的就是哪个对象
只能写在方法中，方法中访问成员变量会默认有个this.

Thi