this用在构造方法第一行，调用其他匹配的构造方法。

**没有引用指向的对象被视为垃圾回收。**

**属性的初始化**

1. 在定义时直接初始化

class｛

A ref2=new A();

｝

1. 在构造方法中初始化
2. 在构造代码块中初始化

classA{}

classB{

A ref2;

{

ref2= new A();

}

}

如果定义多个构造代码块，java编译器会保证按照定义顺序执行。

可以视为把构造代码块的代码放到

静态属性的初始化（**发生在类被加载的时候**）

1. 定义时初始化

Static A a=new A（）；

1. 静态代码块

static ｛

a = new A（）；

｝

**数据结构-复杂度**

1. 衡量算法好坏的时候，两个主要维度（时间、空间）

**时间复杂度**

**大O阶方法：**

1、用常数1取代运行时间中的所有加法常数。

2、在修改后的运行次数函数中，只保留最高阶项。

3、如果最高阶项存在且不是1，则去除与这个项目相乘的常数。得到的结果就是大O阶。

二分查找 O（log（n））

**空间复杂度**

算法执行过程中，占用的额外空间和数据规模n的关系的大o渐进表示法

开辟空间：

1. 栈上的局部变量、形参
2. 堆上的对象（数组、其他对象）
3. 如果是递归方法，考虑调用栈的占用情况

**顺序表**