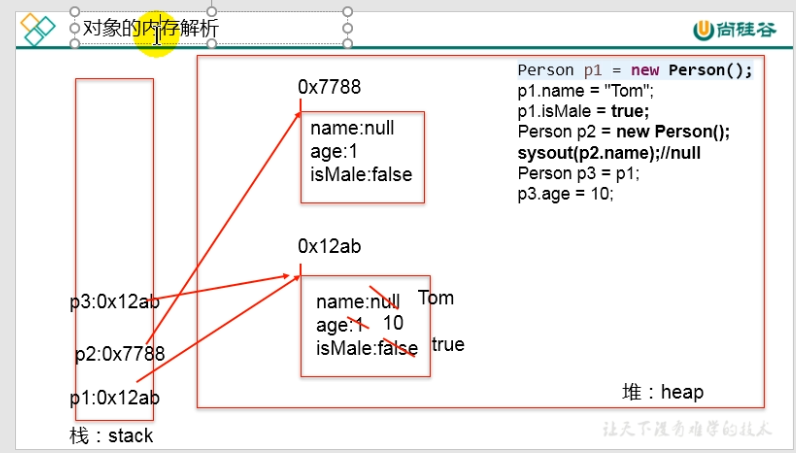
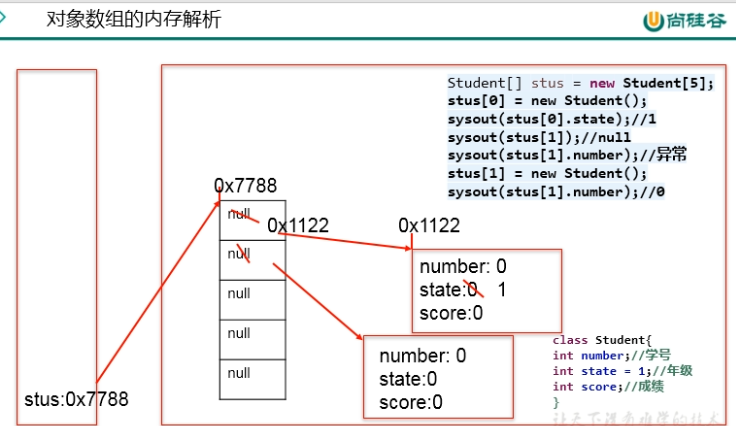
## 对象的内存解析







## 属性(非static)和局部变量的区别

1、定义位置不同

2、关于权限修饰符的不同

3、默认初始化值的不同

属性有初始值

局部变量没有初始值

4、内存中位置的不同

局部变量--栈

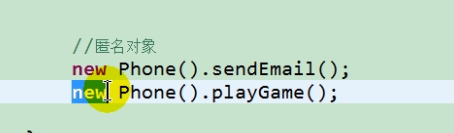
属性--堆空间中

## 理解万事万物皆对象

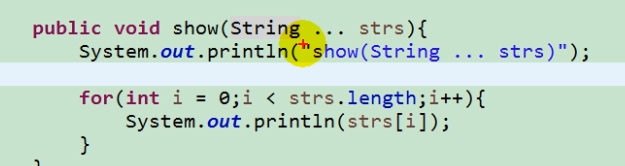
每一个抽象的东西都就可以具体为对象（文件，网络地址，数据库表，数据库连接）

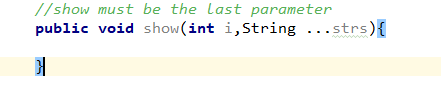
## 匿名对象

创建对象时没有显式的赋值给一个变量名，即为匿名对象

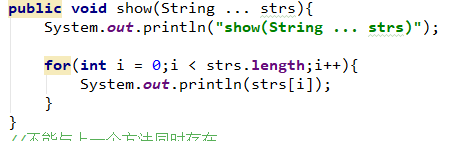


## 可变形参

1 可变个数形参的格式：数据类型 ... 变量名  
2 当调用可变个数形参的方法时，传入的参数个数可以是：0个，1个,2个，。。。  
3 可变个数形参的方法与本类中方法名相同，形参不同的方法之间构成重载  
4 可变个数形参的方法与本类中方法名相同，形参类型也相同的数组之间不构重载。换句话说，**二者不能共存**。  
5 可变个数形参在方法的形参中，**必须声明在末尾**

  
6 可变个数形参在方法的形参中,最多只能声明**一个**可变形参。

使用方法



## 变量的值传递

如果变量是**基本数据类型**，此时赋值的式变量所保存的**数据值**

如果变量是**引用数据类型**，此时复制的式变量所保存的**数据的地址值**

## JavaBean

一种java 语言编写的可重用组件

## 代码块 （初始化块）

1. 代码块的作用：用来初始化类、对象  
   2. 代码块如果有修饰的话，只能使用static.  
    3. 分类：静态代码块 vs 非静态代码块  
      
    4. 静态代码块  
    >内部可以有输出语句  
    >随着类的加载而执行,而且只执行一次  
    >作用：初始化类的信息  
    >如果一个类中定义了多个静态代码块，则按照声明的先后顺序执行  
    >静态代码块的执行要优先于非静态代码块的执行  
    >静态代码块内只能调用静态的属性、静态的方法，不能调用非静态的结构  
      
    5. 非静态代码块  
    >内部可以有输出语句  
    >随着对象的创建而执行  
    >每创建一个对象，就执行一次非静态代码块  
    >作用：可以在创建对象时，对对象的属性等进行初始化  
    >如果一个类中定义了多个非静态代码块，则按照声明的先后顺序执行  
    >非静态代码块内可以调用静态的属性、静态的方法，或非静态的属性、非静态的方法
2. 属性赋值顺序

对属性可以赋值的位置：  
 ①默认初始化  
 ②显式初始化/⑤在代码块中赋值 （谁在下面谁后执行）  
 ③构造器中初始化  
 ④有了对象以后，可以通过"对象.属性"或"对象.方法"的方式，进行赋值  
  
   
 执行的先后顺序：① - （② / ⑤） - ③ - ④