# 类和对象

基本类型 int long 不够用，存是数据比较单一

自定义一个类型

类

Class 类型的名字{

Int

Long

String

Byte

Void show1(){

}

Void show2(){

}

}

对象就是变量名字

Int a=100

Int 类型 a 是变量 = 赋值

Dog b=new Dog()

Dog 是类型 b 对象 new 构造对象 简单的赋值

New 就是在堆里开辟空间，存放这个类型中的所有的数据

基本类型赋值 ，直接在栈内存中赋值

所有的文件，都是类

从类出发

# 特别的函数

构造函数

Main函数 程序入口函数

构造对象的一个函数，开辟空间的一个函数

定义一个变量，必须有类型

name=”张三”

人？狗？猫？

人类 name=”张三”;

狗类 name=”李四”;

构造函数是必须的，但是你不写，编译器会自动生成一个无参的构造函数

构造函数名字必须和类名一致

开辟空间，初始化类中的属性的值

每次new 就会开辟空间。但是发现，前面的对象已经不再被使用了。那么内存中对象还在吗？

因为jvm程序一次编译，导出运行

通过调试，程序慢镜头演示

类加载顺序

第一个执行的方法

前提是类被加载之后调用

静态的变量 ，类加载之后就赋值了

参数传递都是值传递

复制的是地址的副本

基本类型，包装类型

每一个基本都都有一个对应的包装类型

New 实例化，而且里面提供能了很多方法

和基本一样，但是比基本类型多了一些功能

存在堆中。

基本类型和包装类型

装箱和拆箱

1.5之后

this就是当前对象的引用

静态方法不能调用非静态的资源

非静态方法可以调用静态资源

# ==运算符

判断是否相等

分基本类型还是引用类型

基本类型，判断值

引用类型 地址引用是否相等