今天讲的东西是补充内容，有点散，请大家都要认真听，后面会用习题练习

讲作业的时候讲

方法的参数传递

形式参数与实际参数：

public Person(String n, int i); 形式参数

p1 = **new** Person("Tony",18)；实际参数

对象作为参数传递：

public static void changeAge(Person p, int i){

p.age = p.age + i;

}

类和方法的内存解析

方法的种类：

Accessor：返回一个值

Mutator：改变一个量

Constructor

Static

请大家把上次的classTest project打开，在其中的Person 类下继续写入方法

# [Java 中 静态方法与非静态方法的区别](http://www.cnblogs.com/OldZhao/p/5034888.html)

静态方法和实例方法的区别主要体现在两个方面：  
  
在外部调用静态方法时，可以使用"类名.方法名"的方式，也可以使用"对象名.方法名"的方式。而实例方法只有后面这种方式。也就是说，调用静态方法可以无需创建对象。

静态方法：  
**public** **static** **void** changeAge(Person p, **int** i){

p.age = p.age +i;

System.***out***.println(p.age);

}

非静态方法：

**public** **void** changeAge(**int** i){

age = age +i;

System.***out***.println(age);

}

方法的调用：

p1.changeAge(4);

Person.*changeAge*(p2,4);

**参数的传递与变量的存储**

**类中变量类型：**

****一、实例变量****

也叫对象变量、类成员变量；从属于类由类生成对象时，才分配存储空间，各对象间的实例变量互不干扰，能通过对象的引用来访问实例变量。但在Java多线程中，实例变量是多个线程共享资源，要注意同步访问时可能出现的问题。

****二、类变量****

也叫静态变量，是一种比较特殊的实例变量，用static关键字修饰；一个类的静态变量，所有由这类生成的对象都共用这个类变量，类装载时就分配存储空间。一个对象修改了变量，则所以对象中这个变量的值都会发生改变。

在Person 类中写入类变量 **public** **static** **int** *height* = 160;

在主类中执行如下：

p1.*height* = p1.*height* + 3;

System.***out***.println(p2.*height*);

方法中的变量：

1. ****局部变量****

**public** **void** changeAge(){

**int** i = 4;

age = age + i;

System.***out***.println(age);

}

方法中或者某局部块中声明定义的变量或方法的参数被称为局部变量，他们只存在于创建他们的block里（{}之间）无法在block外进行任何操作，如读取、赋值。在Java多线程中，每个线程都复制一份局部变量，可防止某些同步问题发生。

**内存分析：**

Person p1;

p1 = **new** Person("tony",17);

1. Java语言中的变量除基本类型之外都称之为引用类型

基本类型 int i;

i = 1;

其余所有变量都是引用类型：

Person p1；null的变量因为什么都没有给它，成为null引用

p1 = **new** Person("Tony",18); 时，我们就已经实例出了一个实实在在的人

那么现在在对内存中就有一个实实在在的人了，这个人的名字叫Tony，年龄是18岁

在另一个不知道什么的空间中，产生出了一个变量叫做p1， 这个变量指向那个人，因此这个变量被称为那个人的引用，因为当提到这个变量，就自然而然地把它看成那个人了。所以为什么说java中没有指针，因为所有的引用都是指针。

那个人会占用堆内存的空间，堆内存是动态存储空间，因为只有当我们创建了一个对象，才会在堆内存中开辟空间。方法不占用内存，除非被调用，它永远只是一段代码。

思考：当创建了两个对象，他们的引用不同，但是属性都相同时，这两个对象相同吗？

类变量的空间存储：在编译时就在堆内存中开辟一个空间存放类变量，每一个对象都共享静态变量

1. 内存开辟

栈内存存储基础类型

因此栈有一个很重要的特殊性，就是存在栈中的数据可以共享。假设我们同时定义：   
　　int a = 3;   
  
　　int b = 3;   
　　编译器先处理int a = 3;首先它会在栈中创建一个变量为a的引用，然后查找栈中是否有3这个值，如果没找到，就将3存放进来，然后将a指向3。接着处理int b = 3;在创建完b的引用变量后，因为在栈中已经有3这个值，便将b直接指向3。这样，就出现了a与b同时均指向3的情况。这时，如果再令a=4;那么编译器会重新搜索栈中是否有4值，如果没有，则将4存放进来，并令a指向4;如果已经有了，则直接将a指向这个地址。因此a值的改变不会影响到b的值。要注意这种数据的共享与两个对象的引用同时指向一个对象的这种共享是不同的，因为这种情况a的修改并不会影响到b, 它是由编译器完成的，它有利于节省空间。而一个对象引用变量修改了这个对象的内部状态，会影响到另一个对象引用变量

栈内存存储引用类型变量，创建出的引用变量例如：p1，p2就是存储在栈内存中

栈内存作用域：

形参赋值与局部变量的存储：

**public** **void** changeAge(**int** i){

age = age +i;

System.***out***.println(age);

P1.changeAge(4);

**public** **void** changeAge(**int** i){

i = i+1;

}

p1.changeAge(4);

System.out.println(p1.age);

局部变量在栈内存中分配

**public** **void** changeAge(){

**int** i = 4;

age = age + i;

System.***out***.println(age);

}

**大习题：**

**1.What output will be produced by this program?**

public class Mystery

{

public static void strangeMethod(int x, int y)

{

x += y;

y \*= x;

System.out.println(x + " " + y);

}

public static void main(String[] args)

{

int a = 6, b = 3;

strangeMethod(a, b);

System.out.println(a + " " + b);

}

}

(A) 36

9

(B) 3 6

9

(C) 9 27

9 27

(D) 6 3

9 27

(E) 9 27

6 3

**2.**

Public class Time{

private int hrs;

private int mins;

private int secs;

public Time(){}

public Time(int h, int m, int s){}

public void resetTime(int h, int m, int s){}//resets Time to hrs = h,mins = m, secs = s

public void increment(){}//advances time by one second

public boolean equals(Time t){}//return true if this time equals t, false otherwise

public boolean lessThan(){}//return true if this time is earlier than t, false otherwise

public String toString(){}//return a String with the time in the form hrs:mins:secs

}

1. Which of the following is a false statement about the methods?
2. equals, lessThan, and toString are all accessor methods.
3. Increment is a mutator method.
4. Time() is the default constructor.
5. The Time class has three constructor.
6. There are no static methods in this class.
7. Which of the following represents correct implement code for the constructor with parameters?
8. hrs = 0;

mins = 0;

secs = 0;

B. hrs = h;

mins = m;

secs = s;

C. resetTime(hrs, mins, secs);

D. h = hrs;

m = mins;

s = secs;

E. Time = new Time(h, m, s);

1. A client class has a display method that writes the time represented by its parameter:

/\*\* Outputs time t in the form hrs:mins:secs.

\* param t the time

\*/

Public void display(Time t)

{

/\* method body \*/

}

Which of the following are correct replacement for /\* method body \*/?

I Time T = new Time(h,m,s);

System.out.println(T);

II System.out.println(t.hrs + “:” + t.mins + “:” + t.secs)

III System.out.println(t);

注：方法的调用都在另一主类中执行

1. 在classTest中创建Dog 类，将原来主方法中与Person类相关的都注释掉

要求：1.Dog类拥有public属性：性别，年龄，品种，和一个静态属性tail，值是“long”

2.Dog类拥有一个默认构造方法和一个可以传递性别，年龄，品种的构造方法

3.Dog类拥有方法：1.返回一个Dog对象的年龄

2.直接打印一个Dog对象的姓名+品种，例如：Tony Husky

1. 静态方法，使Dog对象的年龄+3，并直接打印出新的年龄

4.1）在主方法中创建两只狗，名字分别是Tony，Bruce，品种分别是Husky和 Samoyed，年龄均为3

2）在控制台分别打印出这两只狗的姓名+品种

3）在控制台分别打印出这两只狗改变后的年龄