JAVA内存分析

1. Java内存区域

Java的内存区域主要分四块，分别如下：

1. heap：堆区域，主要存放的是new出来的数据（对象）。
2. stack：栈区域，存放的是局部变量。
3. code segment：主要存放代码。
4. 实例分析java程序内存分配过程

下面以一个小程序为例说明java程序内存分配过程，代码如下：

public class TestCirclePoint {

public static void main(String[] args){

Point2 point=new Point2(2.0,3.0);

Circle circle=new Circle(new Point2(0.0,0.0),5.0);

System.out.println("点point在圆o里面："+circle.contains(point));

}

}

class Point2{

private double x;

private double y;

Point2(){};

Point2(double x,double y){

this.x=x;

this.y=y;

}

public void setX(double \_x){

x=\_x;

}

public void setY(double \_y){

y=\_y;

}

public double getX(){

return x;

}

public double getY(){

return y;

}

double computeDistanceSquares(Point2 p){

return (this.x-p.x)\*(this.x-p.x)+(this.y-p.y)\*(this.y-p.y);

}

}

class Circle{

private Point2 o;

private double r;

Circle(){};

Circle(Point2 p,double r){

o=p;

this.r=r;

}

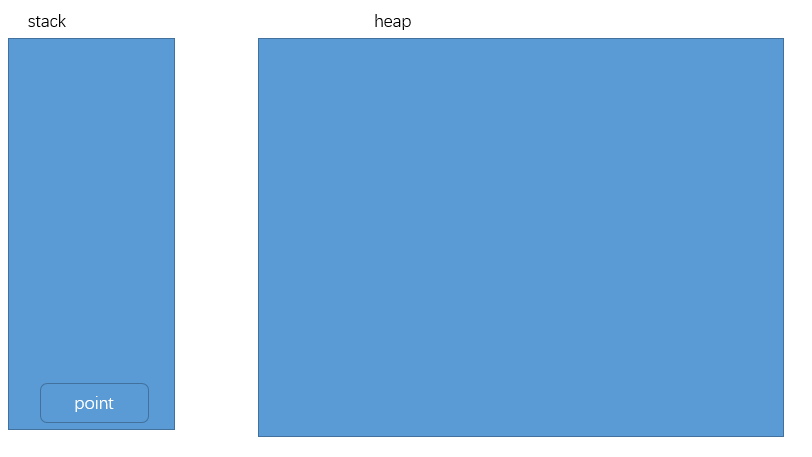
public boolean contains(Point2 p){

return o.computeDistanceSquares(p)<=(r\*r);

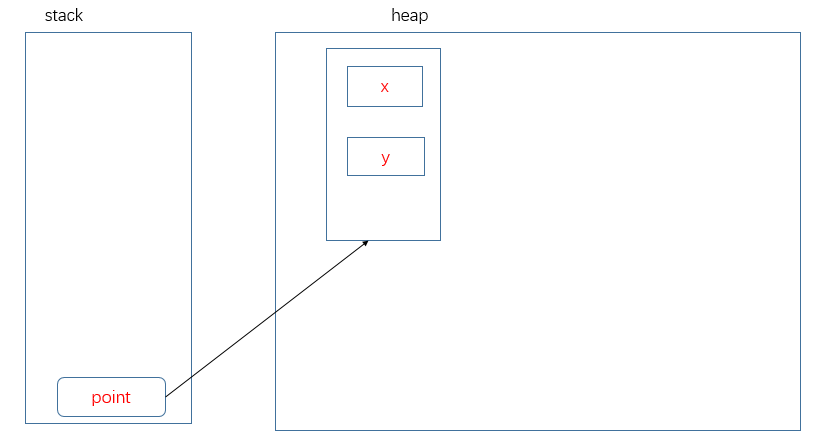
}

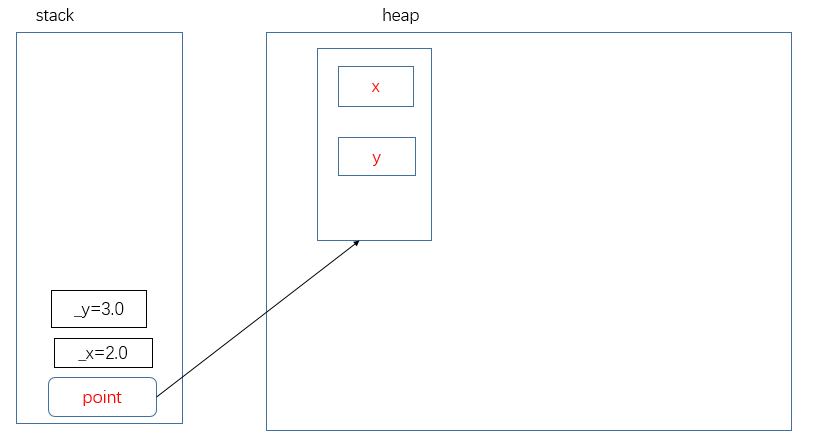
}

首先从main函数开始，栈空间先分配一个局部变量point（引用类型），如下图：

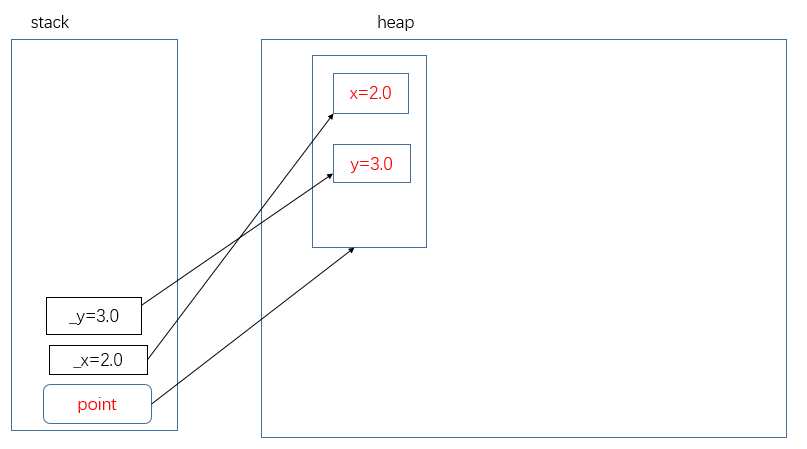


然后程序会在堆内存里new出一个Point2对象来, point引用变量指向这个new出来的对象，如图：

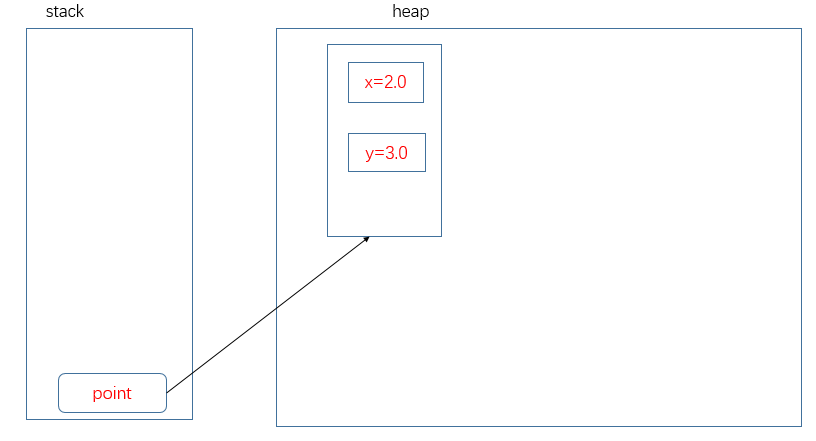
调用构造函数point2，此时栈空间会会生成两个参数变量\_x和\_y，给形参传过去的值分别为2.0和3.0如下图：



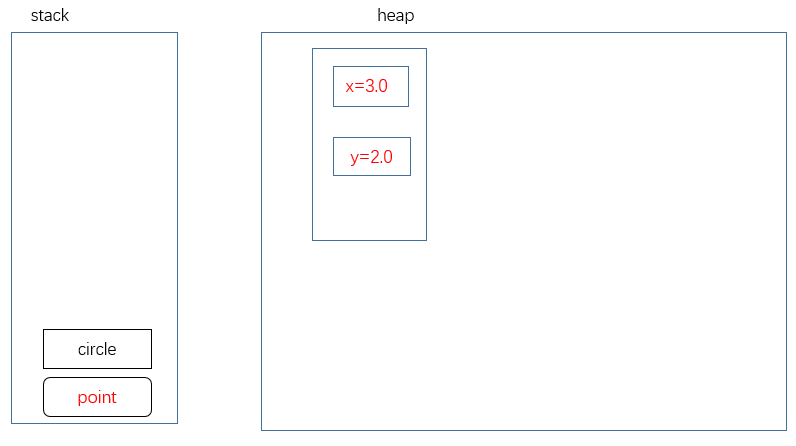
构造函数两个值传给对象的两个成员变量x和y，如图：



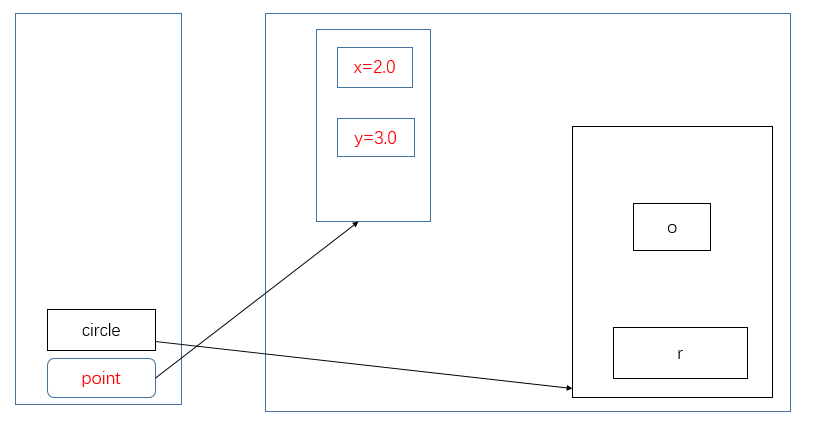
构造函数运行完毕，局部变量被释放，如图



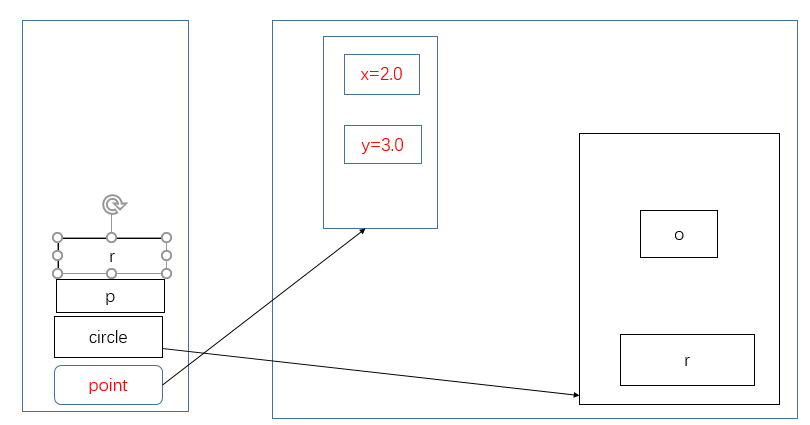
接下来栈空间会生成一个局部变量circle（引用变量），如图：



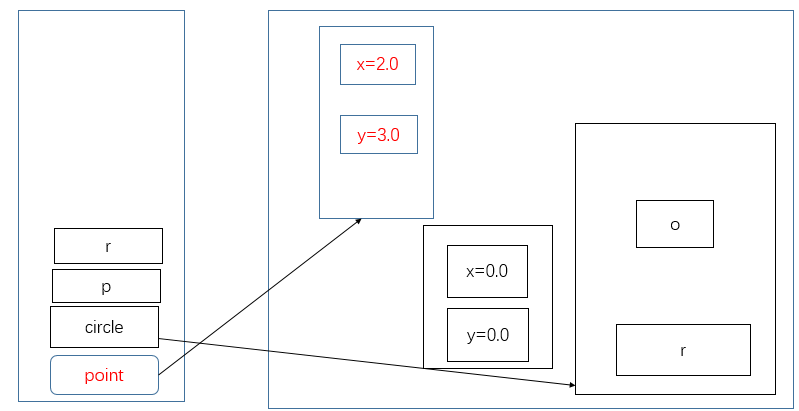
然后会在堆空间new出一个Circle对象来，circle引用变量指向这个new出来的对象，如图：



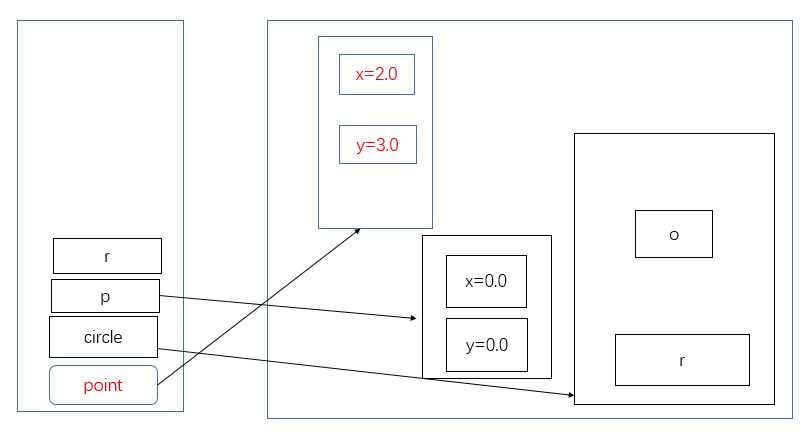
然后调用构造函数Circle，会在栈空间生成两个局部变量，如图：



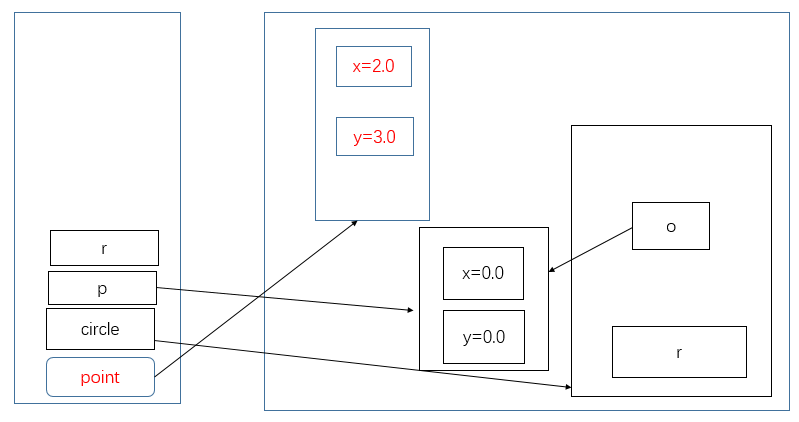
然后会new出一个point2对象，成员变量值分别为0.0和0.0，如图：



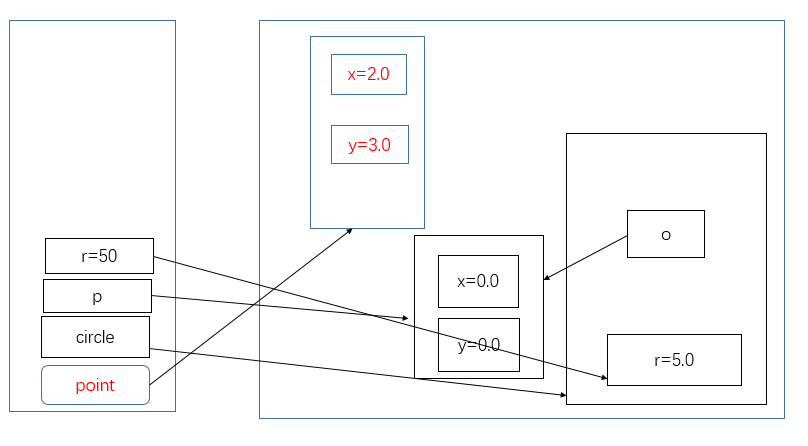
然后将形参引用类型变量p指向这个new出来的point2对象，如图：



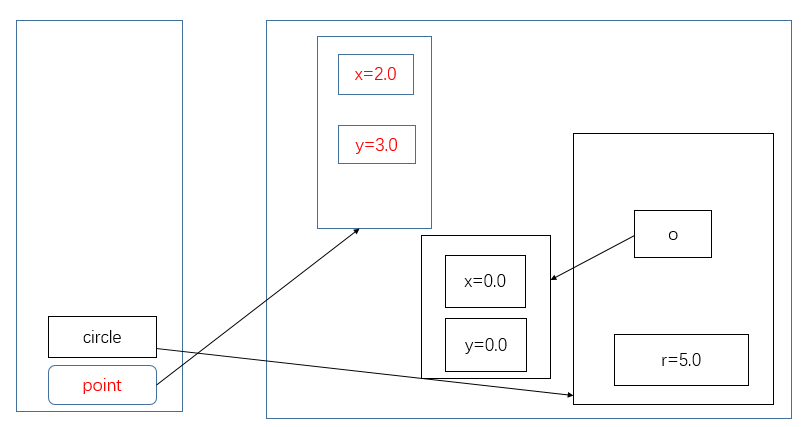
构造函数将p的值传给o，也就是o指向p指向的对象，如图：



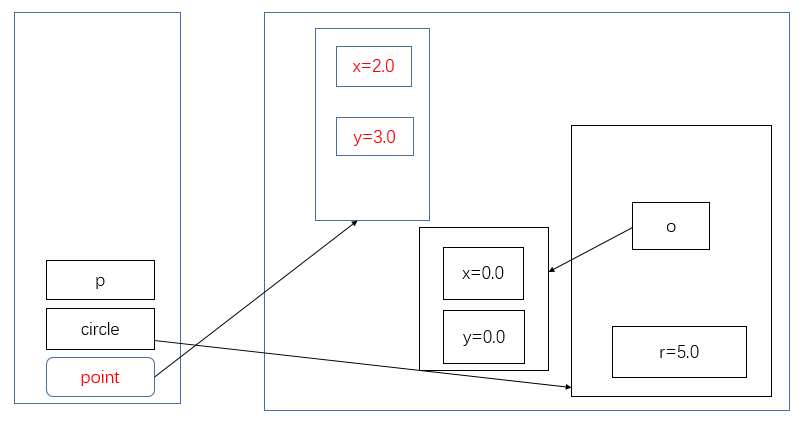
接下来，给形参r赋值5.0，构造函数将r的值传给堆内存中Circle对象的成员变量r，如图：



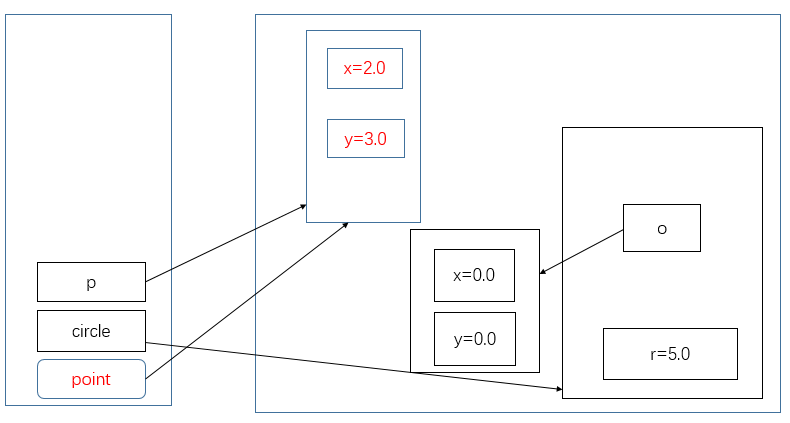
构造函数运行结束，函数中的局部变量都释放掉，如图：



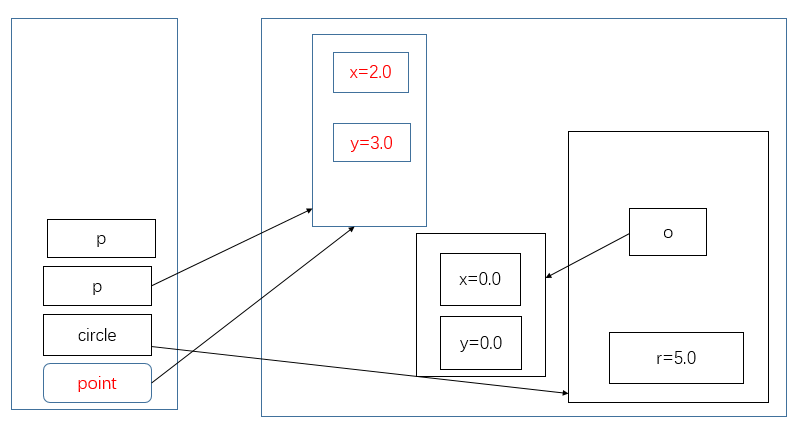
接下类，调用circle对象的contains方法，首先会在栈空间生成形参局部变量p，如图：



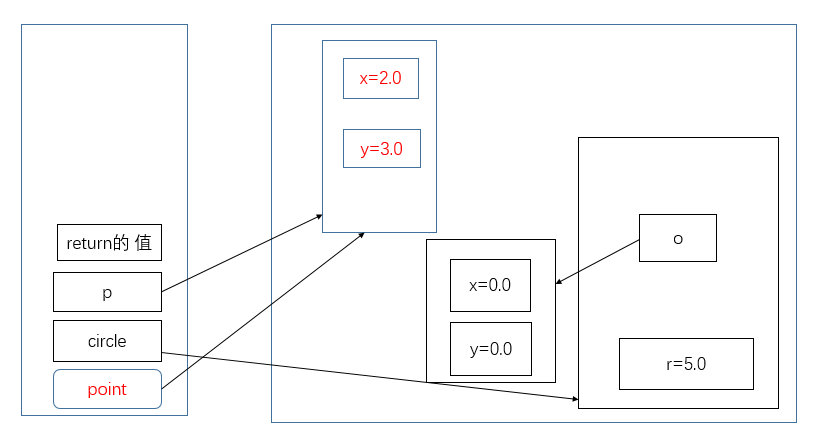
将point的值传给p，也就是p指向point指向的对象，如图：



在contains方法里有调用了对象o的computeDistanceSquares方法，会先在栈里生成形参p，如图：



方法计算结束Return一个double类型的值，存在栈里，局部变量p释放，如图：



这个返回的值返回给contains方法，原来return的值释放掉，而contains方法将接收到的值进行表达式计算，返回一个布尔类型的值，跟上图是一样的，只是这个return是contains返回的布尔类型的值。再打印出来。