this关键字必须放在非静态方法里面

this关键字主要有以下几个用途：

1. **引用成员变量**

在一个类的方法或构造方法内部，可以使用“this.成员变量名”这样的格式来引用成员变量名，有些时候可以省略，有些时候不能省略。

参数a的作用范围为构造方法或方法内部，成员变量a的作用范围是类的内部，这样在构造方法和setA方法内部就存在了变量a的冲突，Java语言规定当变量作用范围重叠时，作用域小的变量覆盖作用域大的变量。所以在构造方法和setA方法内部，参数a起作用。

/\*\*

\* 使用this引用成员变量

\*/

public class ReferenceVariable{

private int a;

public ReferenceVariable(int a){ //定义一个参数（局部变量）a

this.a=a; //将局部变量的值传递给成员变量

}

public int getA(){

return a;

}

public void setA(int a){

this.a=a;

}

}

1. **在自身构造方法内部引用其他构造方法**

当一个类内部的构造方法比较多时，可以只书写一个构造方法的内部功能代码，然后其它的构造方法都通过调用该构造方法实现，这样既保证了所有的构造是统一的，也降低了代码的重复。

在实际使用时，需要注意的是，在构造方法内部使用this关键字调用其它的构造方法时，调用的代码只能出现在构造方法内部的第一行可执行代码。这样，在构造方法内部使用this关键字调用构造方法最多会出现一次。

/\*\*

\* 使用this关键字引用构造方法

\*/

public class ReferenceConstructor{

int a;

public ReferenceConstructor(){ //不带参数的构造方法

this(0); //使用this调用另一个构造方法，0是需要传递的参数

}

public ReferenceConstructor(int a){

this.a=a;

}

}

1. **代表自身类的对象**

在一个类的内部，也可以使用this代表自身类的对象，或者换句话说，每个类内部都有一个隐含的成员变量，该成员变量的类型是该类的类型，该成员变量的名称是this。

/\*\*

\* 使用this代表自身类的对象

\*/

public class ReferenceObject{

ReferenceObject instance;

public ReferenceObject(){ //构造方法

instance=this; //将对象this的值赋值给instance

}

@Test

public void test(){

System.out.println(this);

}

}

在test方法内部，输出对象this的内容，这里的this都代表自身类型的对象。

1. **引用成员方法**

在一个类的内部，成员方法之间的互相调用时也可以使用“this.方法名(参数)”来进行引用，只是所有这样的引用中this都可以省略。