**this关键字只能在方法内部使用。**

**除构造器之外，编译器禁止在其他任何方法中调用构造器。可以用this调用一个构造器，但却不能调用两个。this(参数1，参数2）**

**必须将构造器调用置于最起始处，否则编译器会报错。**

构造器的作用：初始化对象的成员变量。

构造器不能被子类继承，普通方法和属性可以被子类继承。

默认情况下，子类的构造器中的第一行隐式地调用父类默认的无参构造器，以初始化父类的成员变量。

因为子类从父类继承过来的方法，可能要用到父类的成员变量，所以需要在子类的构造器中调用父类的构造器以初始化父类的成员变量。

**Java会自动在导出类的构造器中插入对基类构造器的调用**

在子类中不管是有参还是无参构造器，默认情况下均调用父类的无参构造器（即默认构造器）

等价于：

**public** Student(){

System.out.println("初始化Student的实例变量--无参构造器");

}

**public** Student(**int** sno){

System.out.println("初始化Student的实例变量--有参构造器");

**this**.sno = sno;

}

**public** Student(){

**super**();//super语句只能位于构造器的第一条语句

System.out.println("初始化Student的实例变量--无参构造器");

}

**public** Student(**int** sno){

**super**();

System.out.println("初始化Student的实例变量--有参构造器");

**this**.sno = sno;

}

super()的作用是：调用父类的默认构造器。super.Method():调用父类方法

在构造函数中，super语句必须放在第一句。

任何类的构造函数中，若是构造函数的第一行代码没有显式的调用super(...);那么Java默认都会调用super();作为父类的初始化函数。

如果此时将父类的无参构造器注释掉，而子类的构造器隐式或显式调用了父类的默认构造器，程序编译时将会报错，因为只要在父类中自定义了构造器，系统将不会在其中添加默认的构造器。

若是父类构造器有参数，子类必须显式调用super，调用父类的有参构造器。

当Java new出一个子类对象时（是一个包含**父类的子对象**的子类对象，父类的子对象与单独new一个父类对象一样，只不过这个子对象位于子类对象内部），在执行子类构造器的步骤中，Java默认首先调用父类不带参数的构造器（若子类中没有使用super()调用父类的构造器），然后回来执行子类构造器。

构造方法可以调用本类的其他方法。既然可以调用本类的属性,那么也就可以调用属于本类的其他方法。