

this:

- 1) 可以用来修饰属性，方法和构造器
- 2) this理解为当前对象或当前正在创建的对象（谁调用当前对象谁就是this）如this.name
- 3) 可以在构造器中通过“this（形参）”的方式显示的调用当前类的重载的指定的构造器，

要求：①在构造器内部，只能写在该构造器的首行；

②若一个类中有n个构造器，那么最多有n-1个构造器指定this（）；

```
public void setName(String name)
    this.name = name;
```

this.name: 表示当前正在创建的对象

name是形参

修饰构造方法: this（）: 可以用来显示的调用当前类的重载的指定的构造器

```
//name:是形参
public Person(String name,int age){
//    this.name = name;
    this(name);
    this.age = age;
}
```

## 二，javaBean

所谓的javaBean，则需要满足下面的要求：

- ①类是公共的
- ②有一个无参的公共的构造器
- ③有属性，且为对应的set和get方法

## 三，package和import关键字

1，package: 声明源文件所在的包，， 写在程序的第一行

每. 一次，表示一层文件目录

包名都要小写

2，import:

- ①显示的导入指定包下的类或接口
- ②写在包的声明和源文件之间
- ③如果需要引入多个类和接口，则需要并列引入
- ④如果导入的类是java.lang包下的，如system stirng math等，就不需要显示的声明

⑤.\*的概念，如java.util.\*;

⑥如何处理同名类的引入，如在util包下和sql包下的date类

```
Date d = new Date();  
List list = new ArrayList();  
  
java.sql.Date d1 = new java.sql.Date(522535114234L);
```

⑦import static表示导入指定类的static的属性和方法

⑧导入java.lang.\*;只能导入lang包下的所有类或接口，不能导入lang的子包下面的类和接口

#### 四，object类

object类是所有类的子类

1, ==

①基本数据类型，根据基本数据类型的值判断是否相等，相等返回true反之返回false，比较的时候俩端的数据类型可以不同，在不同的情况下，也可以返回true

②引用数据类型，比较引用类型变量的地址值是否相等

2, equals () ; (java.lang.Object类，是所有类的父类)

①只能处理引用数据类型变量

②在Object类中，equals () 比较的仍然是俩个引用变量地址值是否相等

(string类重写了Object中的equal方法)

③像string包装类file类date类这些重写了Object类的equals方法，比较来个对象的实体是否完全相同

3, toString (在Object类中，返回一个string类型的数据) 打印类和实体在堆空间的首地址值

toString () 方法:

```
public String toString() {  
    return getClass().getName() + "@" + Integer.toHexString(hashCode());  
}
```

①当我们打印一个对象的引用时，实际上默认调用的就是这个对象的toString () 方法

②当我们打印的对象所在的类没有重写Object中的toString () 方法时，那么调用的Object对象默认的toString方法，返回此对象所在的类以及对应的堆空间实体的首地址值;

③当我们打印的对象所在的类重写了Object类中的toString（）方法时，调用的就是我们自己重写的toString方法，常常将对象的属性信息返回（手动实现）

④在没有重写toString方法时，返回return getClass().getName() + "@" + Integer.toHexString(hashCode());

⑤像string类，file类，包装类，date类已经实现了Object类中toString方法