### 类:对某个事物的抽象，将这个事物的属性方法抽象出来，例如将一个人抽象的话就是--------------------------》》

class Person{

private String name;//属性

### private int old;

sleep(); // 方法

}

### private int old:

}



对象：将一个类实例化，例如 Person xiaoming = new Person(); xiaoming就是一个对象。

### 面向对象思想：万物皆对象！

**构造方法：**方法名与类名相同，可以有多个构造方法；没有得情况下，系统默认无参构造方法。 在子类构造方法中必须调用父类构造方法 super（）；

**重载与重写：**

重写：必须是在子类继承父类的情况下，在子类里对父类的方法进行重写；

重载：是在同一个类中的两个或两个以上的方法，拥有相同的方法名，但是参数却不相同，方法体也不相同，最常见的重载的例子就是类的构造函数。

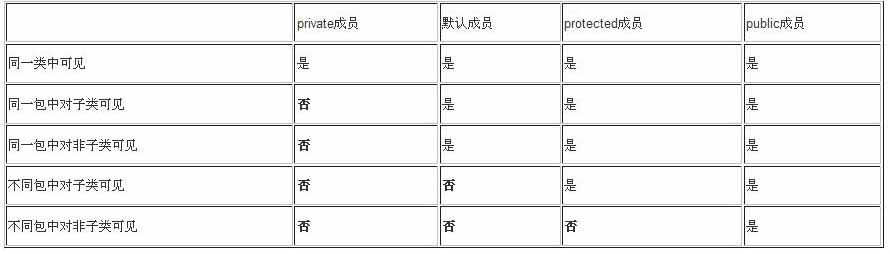
**This：**当一个对象创建以后，在JVM中就会为其分配一个指向自己的指针，就是this。

在Person类中xiaoming对象 this.name 就相当于 xiaooming.name。还有this不能用于静态函数中。

**Static：**在《Java编程思想》P86页有这样一段话：　　“static方法就是没有this的方法。在static方法内部不能调用非静态方法，反过来是可以的。而且可以在没有创建任何对象的前提下，仅仅通过类本身来调用static方法。这实际上正是static方法的主要用途。”

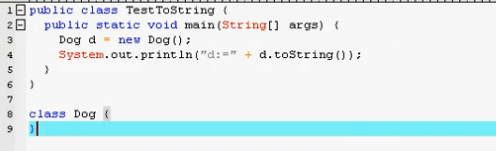
Static方法作为静态类，它是独立于对象而存在的。非静态类中可以直接调用该类的方法属性，而不用new出一个对象。在内存中，static类仅加载一次，只有一个存在于内存中，不可拷贝。对于众多对象来说，它是共用的（为什么我污了。。。）。 目前还不是太了解，未完待续！静态方法不能访问非静态变量

**修饰符：**public的类、类属变量及方法，包内及包外的任何类均可以访问；  
 protected的类、类属变量及方法，包内的任何类，及包外的那些继承了此类 的子类才能访问；  
 private的类、类属变量及方法，包内包外的任何类均不能访问；



**Extends：**继承：java特性之一，一个子类只有一个父类，一个父类可以有多个子类。子类可以调用父类的方法。在子类中可以用super来调用父类的方法，同时对父类的方法进行重写。

**Final: final 修饰变量不可修改！**

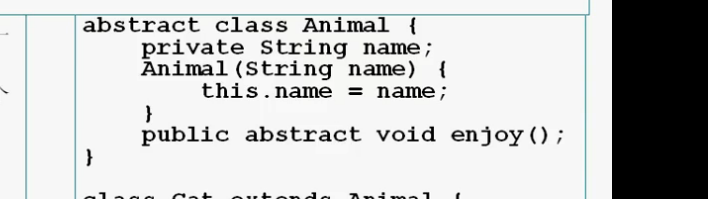
**Tostring：**Object类中有定义有public String toString()的方法，其返回值是String类型，描述当前对象的有关信息



例如上图，系统默认输出它的hashcode

这个就是对 toString的重写。

抽象类：

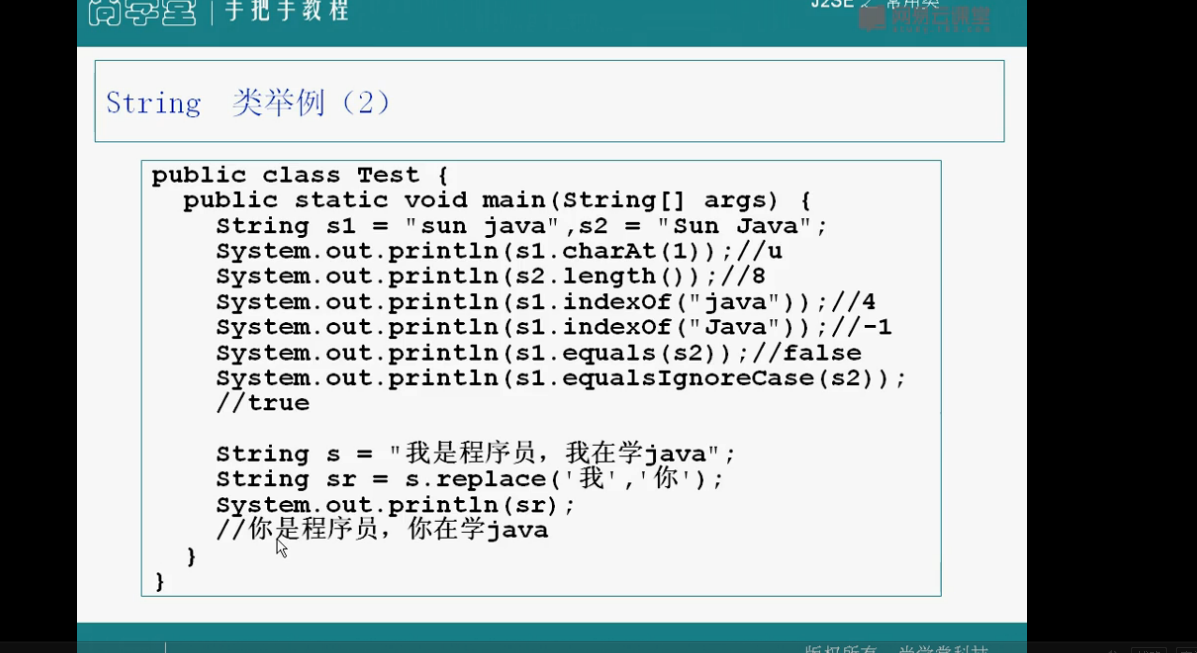


Interface：接口中定义了一些抽象类。使程序更有扩展性。变量前默认public static final修饰，所有的方法默认abstract方法，接口可以继承接口

Implements：调用借口 A extends b(父类) implements c（接口），调用后，需要对接口中的类进行重写。

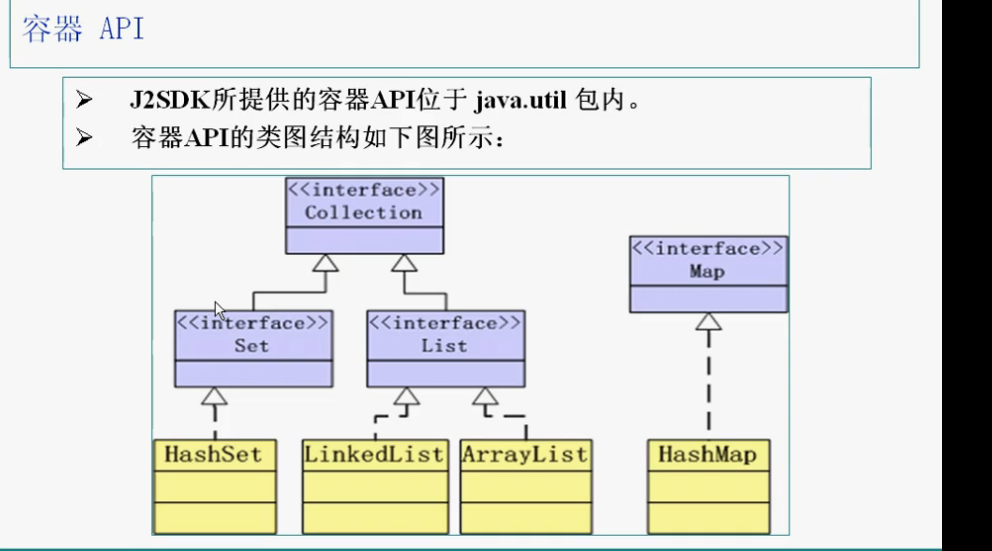
# **异常：**

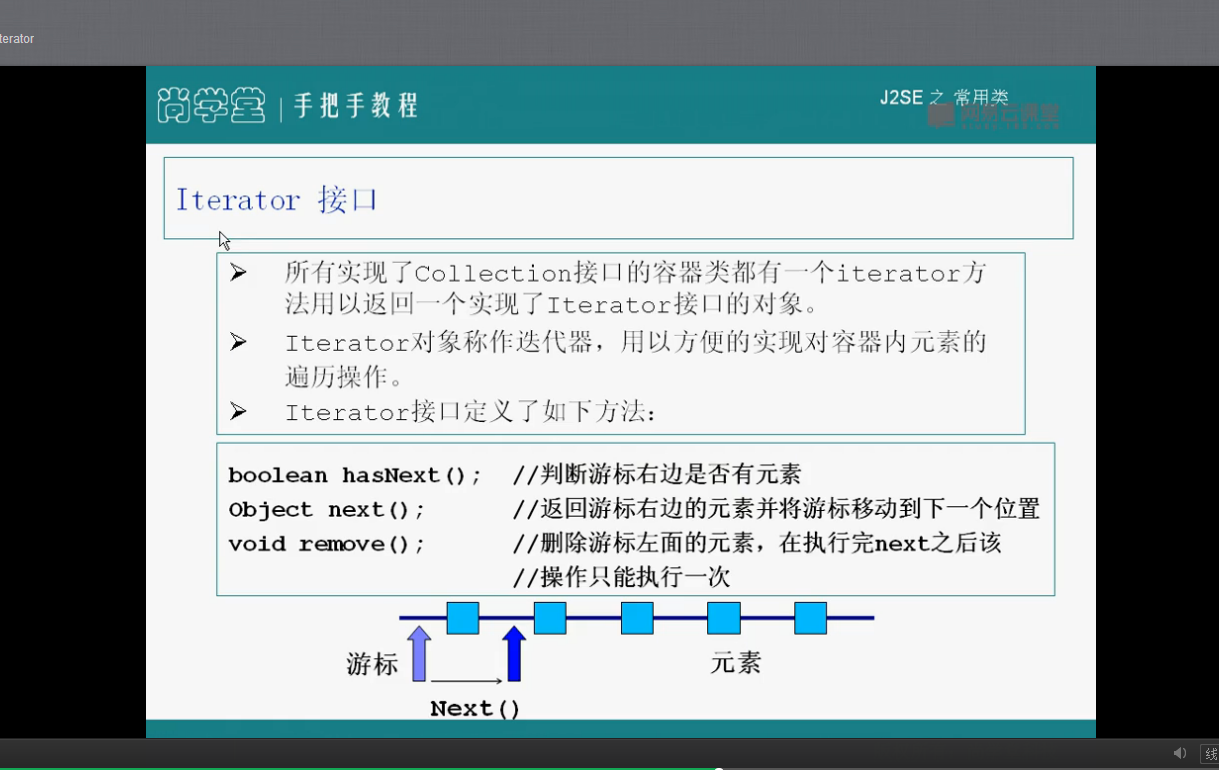
# String类

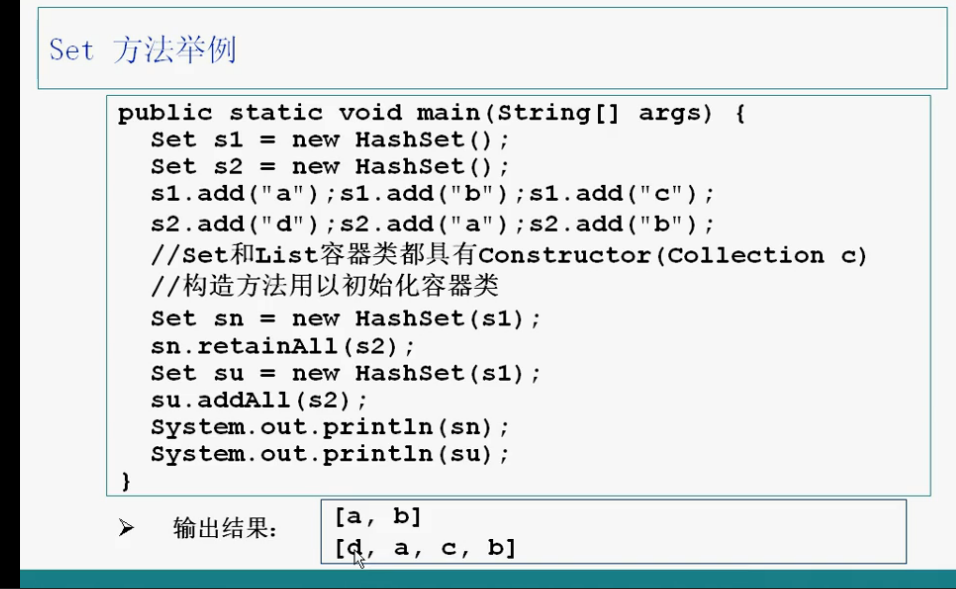




# 容器：





Set：相当于集合

Set 就是集合

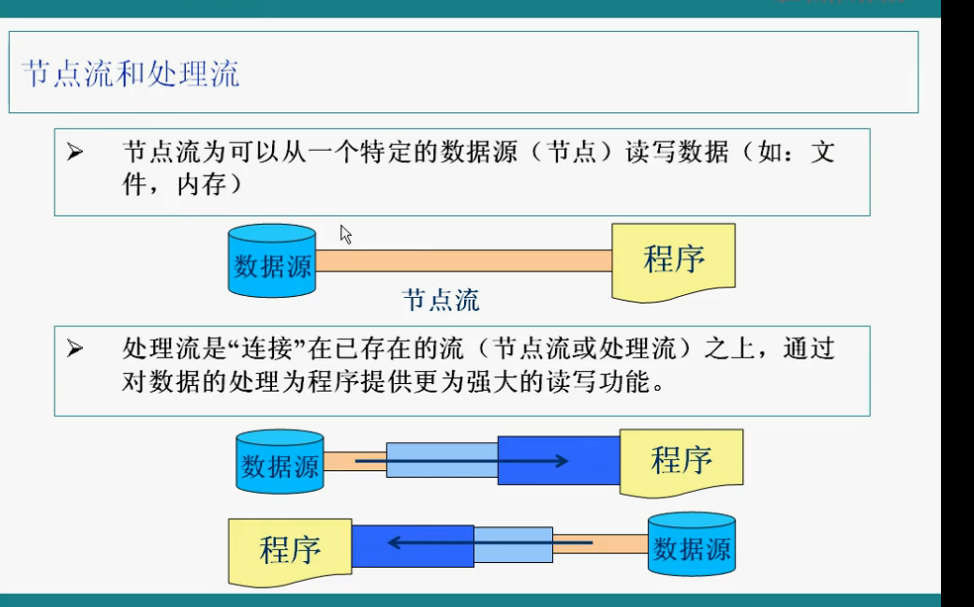
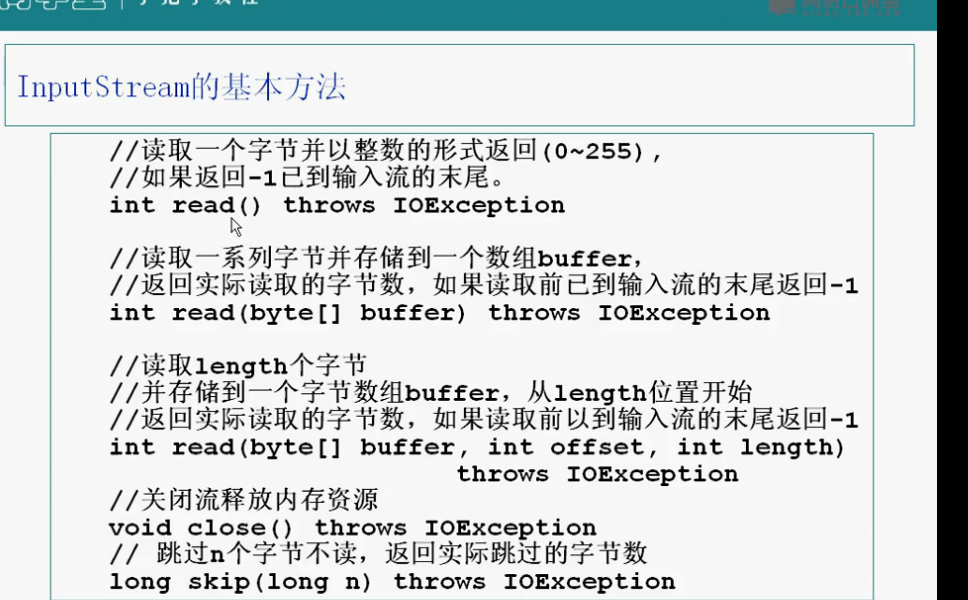
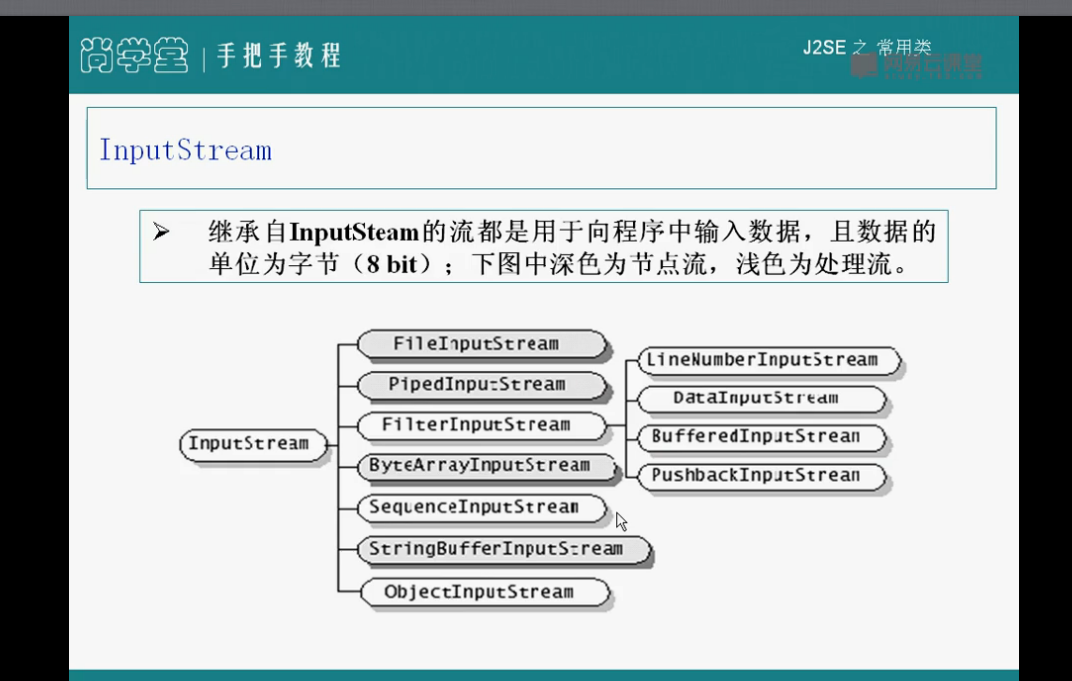
List 链表

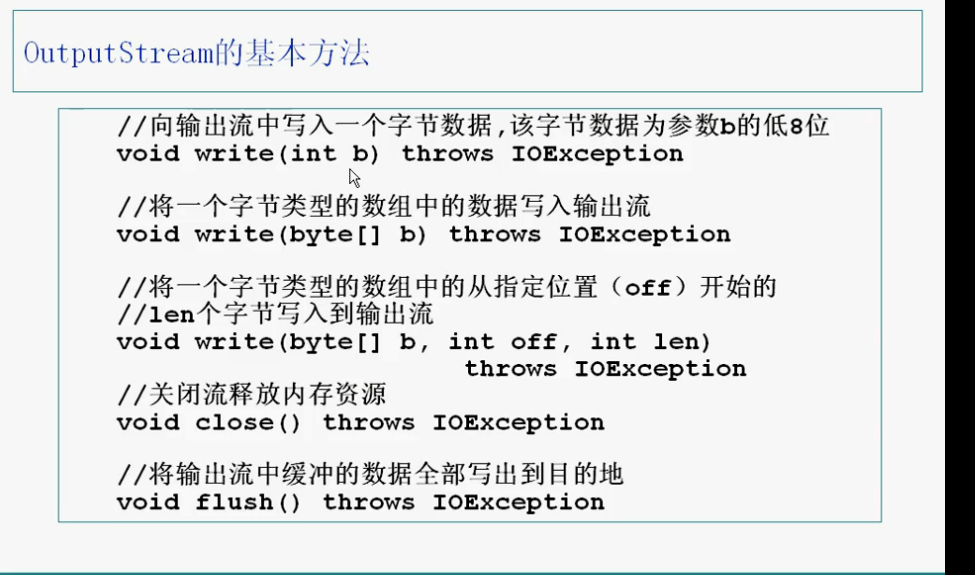
Arraylist 是 数组链表

Lindlist 是 动态链表



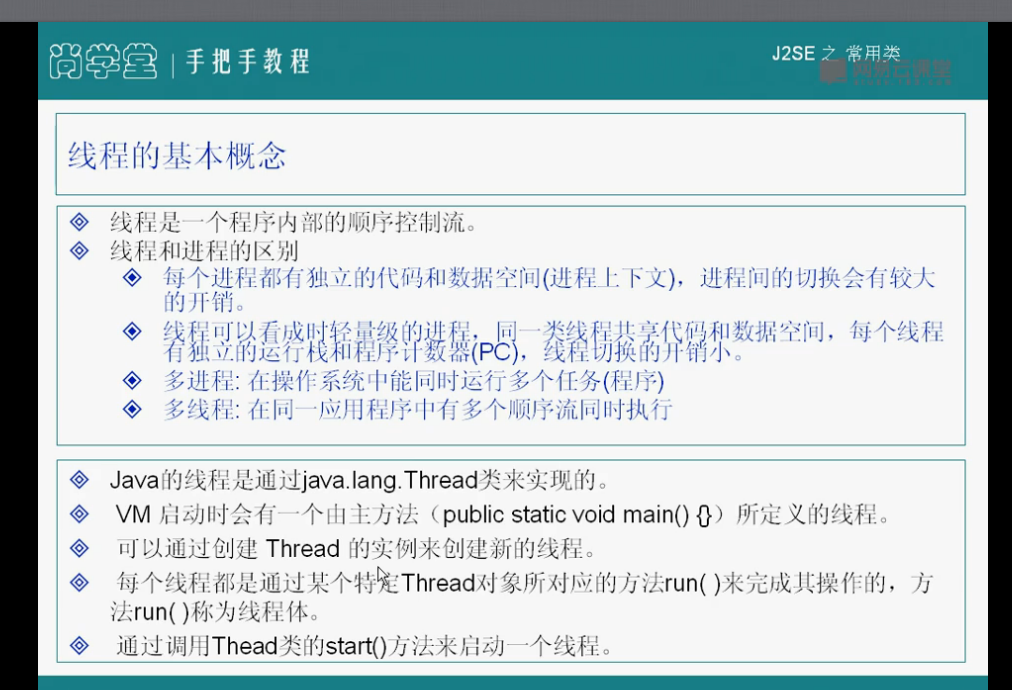
# 流：



先写 flush 在写close

# 多线程



使用多线程的两种方法：

1. 用接口runnable 并且在main函数中线生成这个类的对象，并把这个对象传输Thread XX = new Thead（该对象）； 然后就可以使用了；
2. 继承 Thead 然后在main函数中，new出对象，然后直接使用。

# 内部类：将一个类写在另一个类内部，那么该类就可以直接调用另一个类的属性。

