从主方法传值以后，值会先跟着super到父类中，为父类赋值，此时

在子方法中用super 。 变量 可以打印输入的值

此时this也可以调用

不加this也可以

当子类变量名与父类变量名相同时

Super仍相同

但用this时，构造方法检测到子类有这个变量，会把变量给赋值为null

此时再用this和不加this都是null了

在大多数情况下this和super是可以省略的

但要记住，在无参构造方法后面会默认给this变量赋值

且在每个构造方法后面都会默认调用一个super调用父类，除非再写一个进行覆盖

大多数情况下不需要区分super和this

只有当子类和父类变量名相同时

才需要做出区分

常量

常量不会变，且只能被赋值一次

Static final COUNTRY = “中国”

静态变量是相对于实例变量的

Static country = “中国”

可以改变

抽象方法和构造方法

父类不写构造方法会默认有一个无参构造

子类也是

当写了有参构造会把无参给覆盖掉

再想要无参构造方法需要另写

子类构造方法会默认有个super调用父类无参构造

若父类用有参了，把无参覆盖

那就会报错

抽象方法

Abstract加在类前为抽象方法

还有抽象类是加在public后面

抽象方法只能出现在抽象方法中

若子类不是抽象类

则需要进行方法重写，把abstract去掉

若子类也是抽象方法则不用重写

接口

接口只允许出现常量和抽象方法

常量格式和抽象方法格式可省略一部分

Public abstract int sum（int a，int b）可省略前面两个

常量正常写

接口可以多继承

接口的抽象方法中不能出现方法体

接口被实例类（没有被abstract修饰）继承（实现implements）时

也要进行重写（实现）

不过要把之前省略的格式给补齐了

重写的方法，访问权限只会更高，所以用public前缀

11/11

泛型

创建类时，泛型填什么都可以

关键还是在main方法new对象时，你给类的泛型是啥

接下来你的方法和各种值的数据类型都要是new时所声明的

类中写·的·泛型只是个标记，意思是，我这里用了泛型

实际还得看调用时所声明的泛型是啥

Foreach 只要知道数组名即可遍历数组元素

Treeset’可以本身是无序列表，但可以根据元素大小进行排序

列表

Collection包含List和Set

List有序

Set无序（顺序与存入时不一样）