JAVA在5.0之后就允许class和method definitionli有type形的参数

Type:所有类型的公共高级接口（所有类型的爹）

这样的class definition叫做generics

用type parameter的Generic编程让写出的代码可以应用到任意class上

是标准java库中的一个class

与array不同，array的长度是定死的，arraylist是在程序运行时，一个长度可以增大减小的object

通常来说arraylist和array在程序中是同一个定位，区别仅在于arraylist可以改变长度

当使用array作为一个private实例变量时，Arraylist 被implement

当隐藏的array满了，一个新的更大的隐藏array被创造，然后数据转移到新array

为什么不用arraylist完全替代array

1. arraylist效率低
2. 他没有方便的方括号
3. Arraylist 的base type必须是class type(或者其他reference type)：他不能是primitive type

最后一点其实不太重要，因为现在java提供了自动的boxing和unboxing

使用ArrayList Class

要使用arraylist 必须Import

一个arraylist创建和命名与object相同，除了你要声明basetype



括号里你可以指明初始arraylist长度

20个String为base type的arraylist

注意这个base type被认为是type parameter

add可以在arraylist中加入元素，加在空白的后面，

第一个参数是index，第二个是要add的数，这个操作完以后index1位置是c6，index2是原Index1

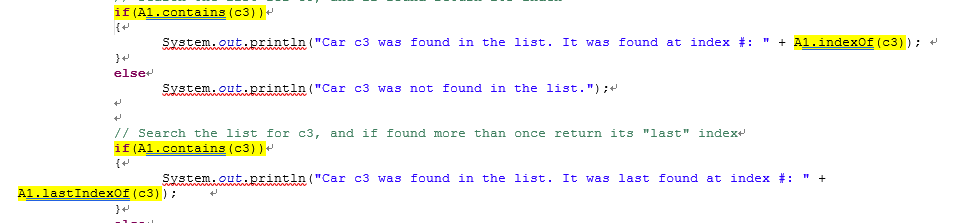
size，得到现在arraylist中总共有几个元素，哪怕length是5，而5个都是Null，那么就是size=0

set



Get

Contains 会返回一个boolean值



注意如果我们对同一个元素连续contains两次，第二个contain代表着这个array有两个以上的这个元素

indexOf返回一个Int值

会return最后一个c3所在

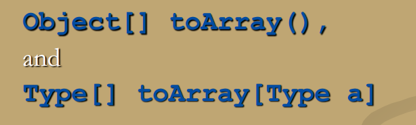
removeremove index2

remove以后size变小，如果remove了index2，原来Index2消除，index3变为index2

clear: 消除所有元素,size变为0

isEmpty: Empty，true

把arraylist转成array

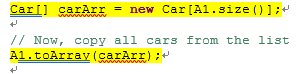


方法1这里创建了一个A1 arraylist



但这里1是Object类型，要使用typecast

方法2



这时候carArr就有着所有arraylist

Car[] carArr2 = **new** Car[A1.size()-2];

Car[] carArr3 = A1.toArray(carArr2);

//如果这里carArr大小小于实际arraylist，而A1.toArray(chArr2)会return一个正确大小的array

然后carARR2是空的，carArr3是正确复制的

如果这个array与carArr2大小相等，那么就会成功复制给carArr2，然后再return，那么carArr2,carArr3都是正确的

为什么有的参数的type是base type，有的是

因为ArrayList implement了许多interface并且继承了许多method。

这些interface 与ancestor class 可以辨认出那些parameter的type是Object

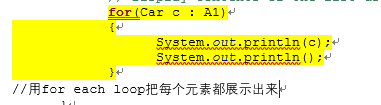
The ”For Each ” Loop

ArrayList 这个Class是collection class的一个例子

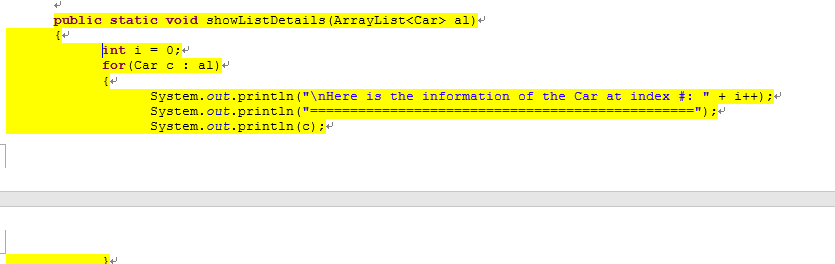
Java5.0开始，java加了一种新的loop，叫做或者 loop

这种loop会不断循环，直到彻底循环完arraylist这样的collection中的每一个元素





ArrayList也可以作为参数使用



一个ArrayList会自动增长，但是不会自动收缩（比如初始长度是5，但是实际size是0）

这时候就可以用这个方法，会自动搜索

当需要dep copy一个ArrayList object的时候，clone是不够用的，因为只是shallow copy

如果要deep copy，那么必须有可能把base type deep copy

这样arraylist 里面每个元素可以被deep copy然后放到新Object

标准库还有一个，和ArrayList几乎一模一样，但是我们更倾向于使用arraylist

ArrayList是一个parameterized class参数class

他有一个参数，，他可以用所有reference type来代替

5.0以后java可以用types来作为参数，这些class叫做parameterized class或者generic difinitions或者generics

我们这里讨论的Arraylist 或是Vector，可以用base type作为参数，也可以用Object作为参数

Generics



这个BaseType就是type parameter

Class或者method可以有type parameter参数T就是type parameter

一个type parameter可以有任意reference type插入，以下图插入了String和Integer



当一个特殊的type被输入，（比如String）就会产生一个专门的class type或method

传统上，一个单独大写字母被用来当做type parameter，但是任何没有关键字的identifier也有可能被使用

Type parameter在class heading的class name后面用尖括号包裹

**class** GenClass<T> { 注意T在这里只是名字，他可以是String也可以是Int

// Attributes

**private** T x; // Again, x is just an object name; it can be anything

// Constructor

**public** GenClass(T y)

{

x = y;

}

**public** **void** setX(T y)

{

x = y;

}

**public** T getX()

{

**return** x;

}

**public** **void** showInfo()

{

System.*out*.println("Here is the data of the object; it is: " + x);

}

} // end of class

GenClass<String> gc1 = **new** GenClass<String>(s1);

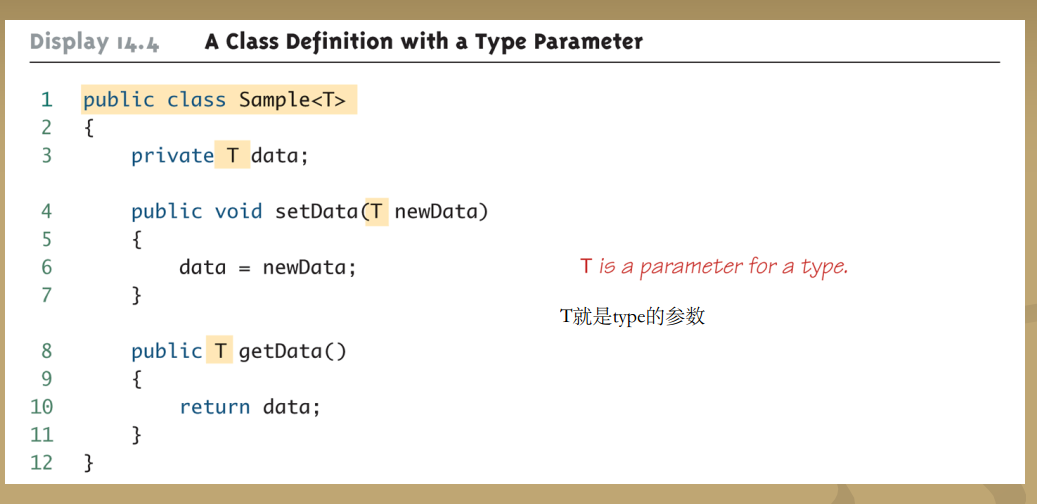
GenClass<Integer> gc2 = **new** GenClass<Integer>(i1);

有着type parameter的class definition被存储在文件里并像其它标准class一样被编译（比如上面一串GenClass）

一旦parameterized class被编译完成，他可以像其它class一样被使用

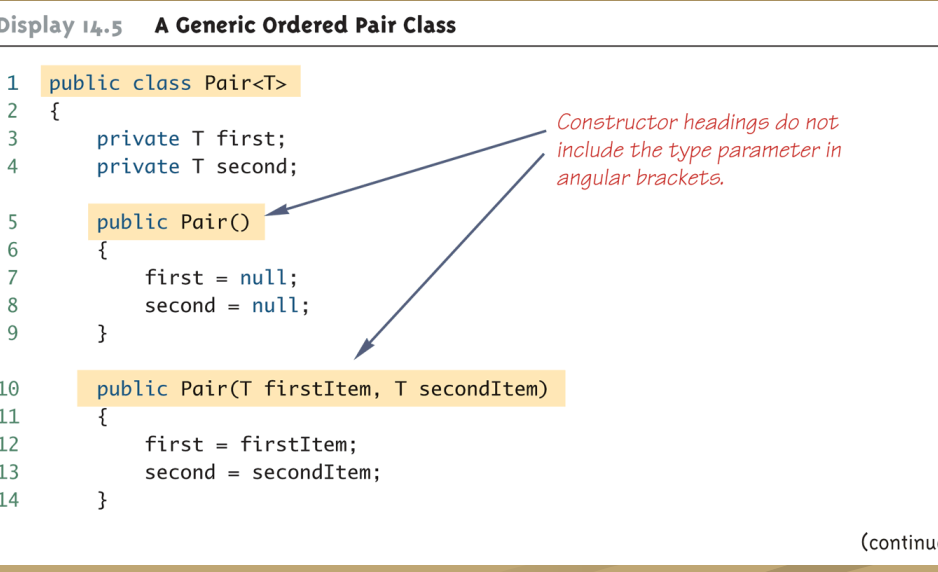
当然，在他被使用之前，先要确立好type parameter是啥（加入了String或者Integer后，才能用它里面的method）

这个叫做instantiate（举例说明/实例化）一个generic class（就是确定T是String的过程）

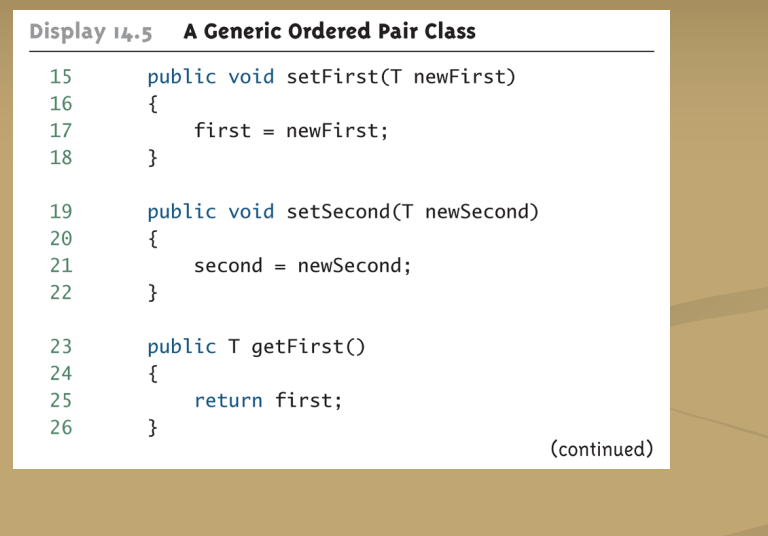


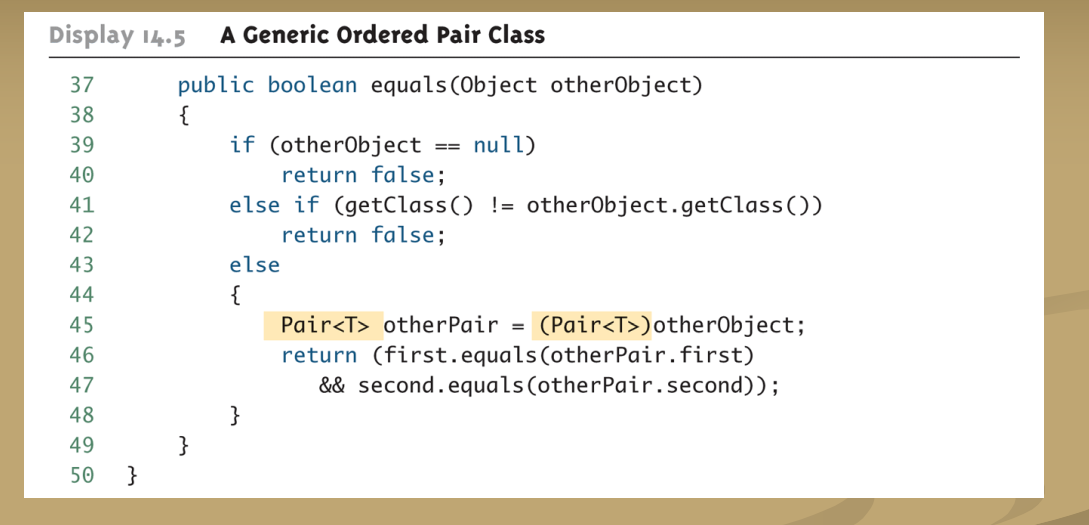
插入这个以后，可以对代码的潜在问题或问题产生有益的诊断，

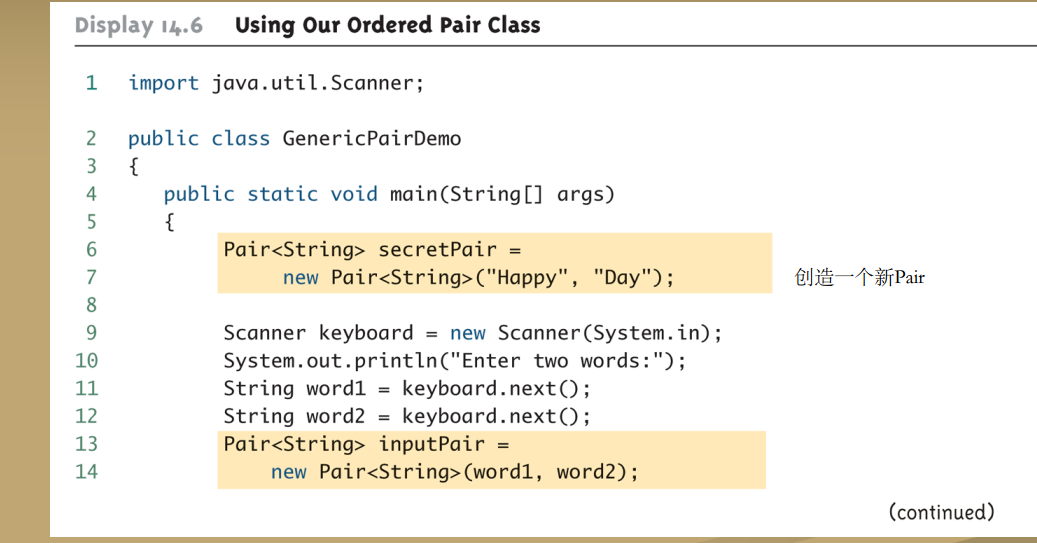
A generic ordered pair class

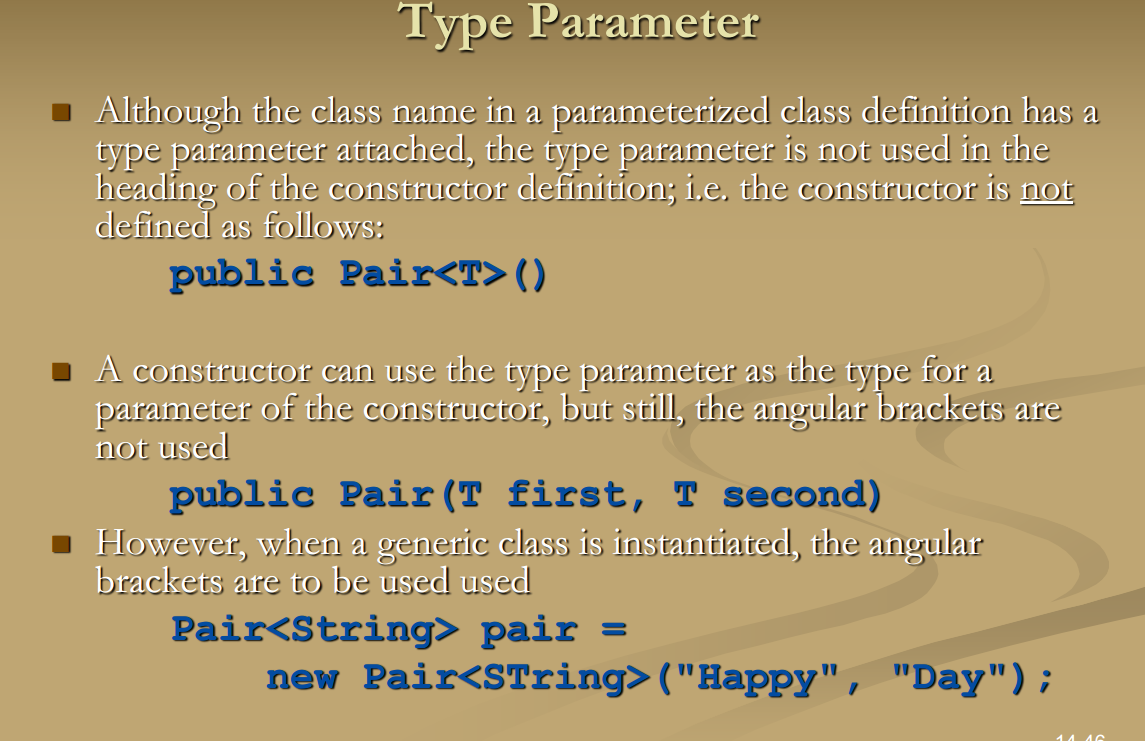


有两个item









注意了generic paired 的constructor没有Parameter参数

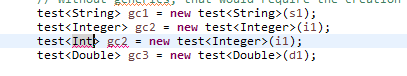
，他可以吧parameter type作为参数但是没有尖括号

但是instantiate的时候，有尖括号

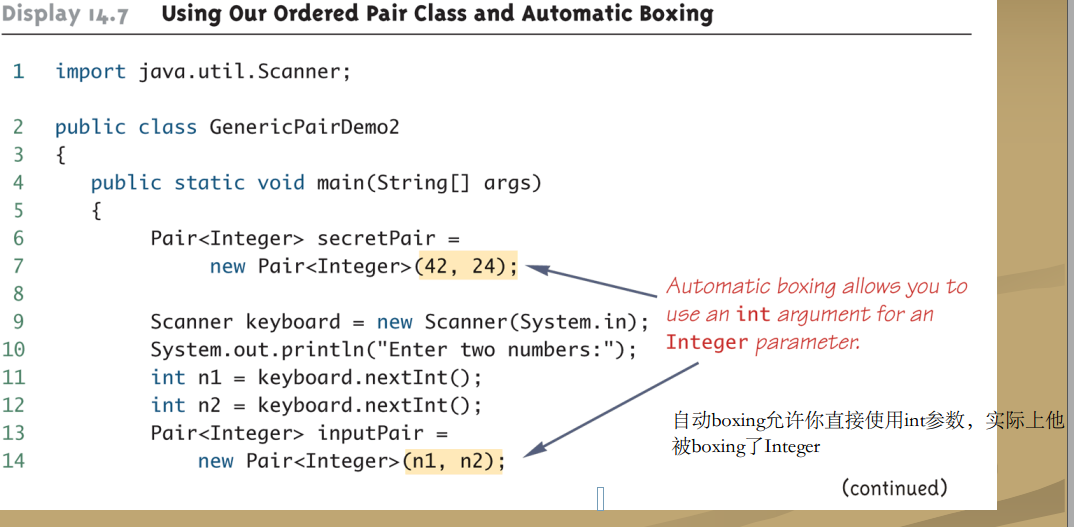


注意了plug 的parameter type一定要是reference type，换句话说，不能用int,double或者char这种Primitive type

当然现在java有了自动的boxing，所以不是特别严格·



可以用Integer代替int



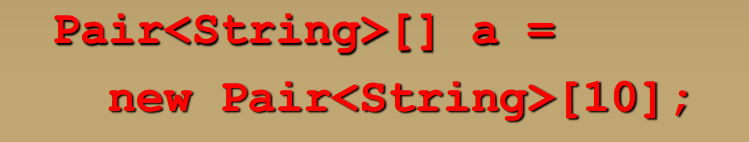
Type parameter 无法被使用，在type name可以被使用的地方

在parameterized class的definition中，普通的class name是可以被使用的，但type parameter是不可以被使用的。

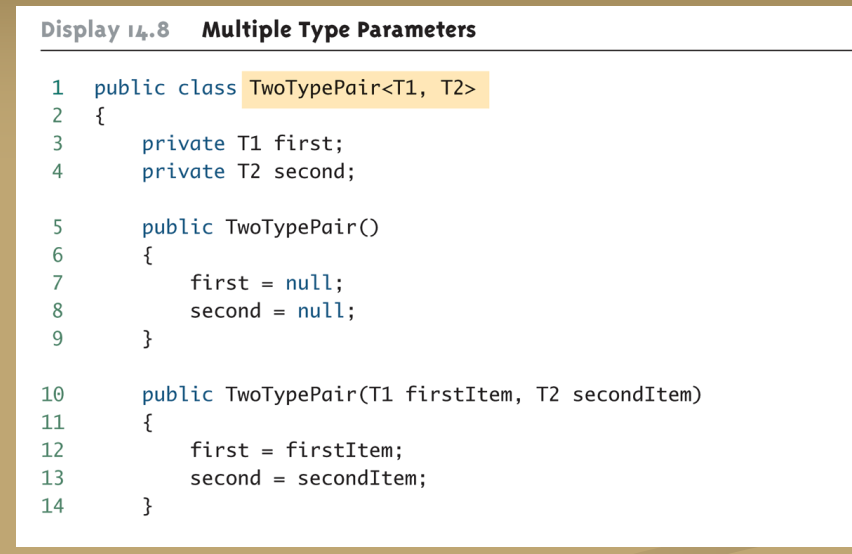
另外，不可以用type parameter 来new object

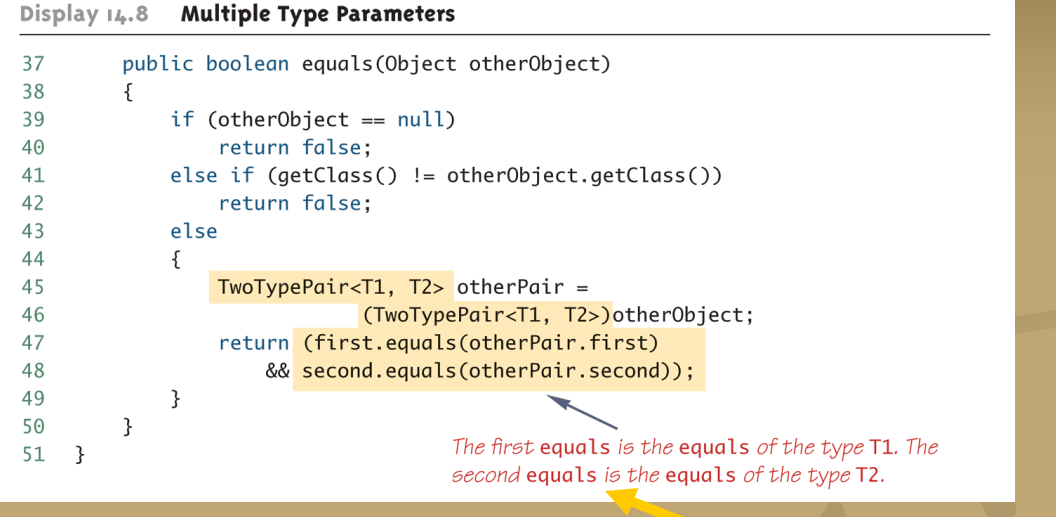


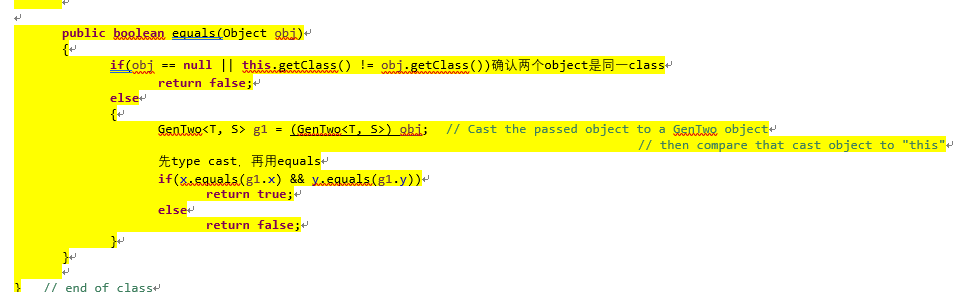
你不可以使用array base type 来instantiate



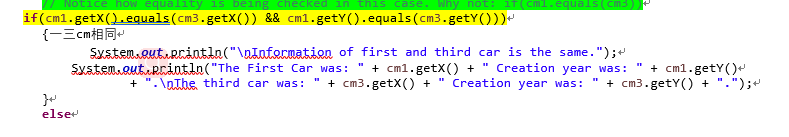
一个class可以拥有多个type parameter



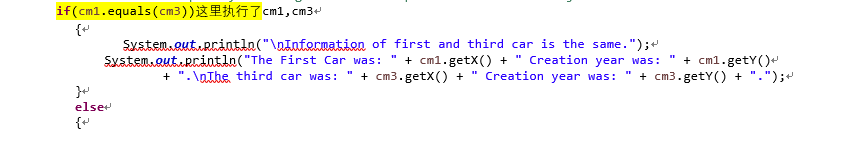




Doc2 和doc3的区别  
doc2



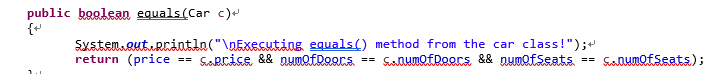
Doc3



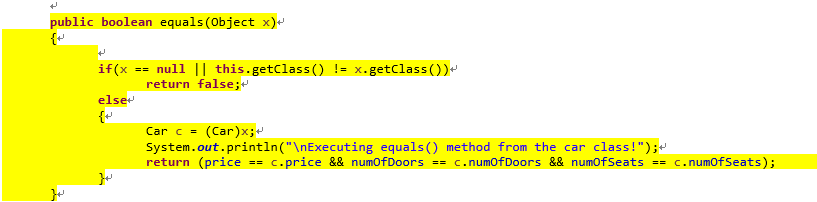
Dpc2使用了各种getX，get出来的是car，所以用的是car的equal

Doc3用的就是cm的equal

34区别，3的car equal



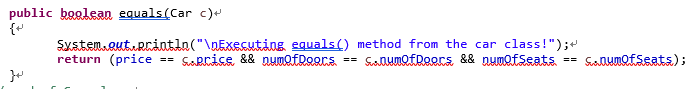
4的car equal



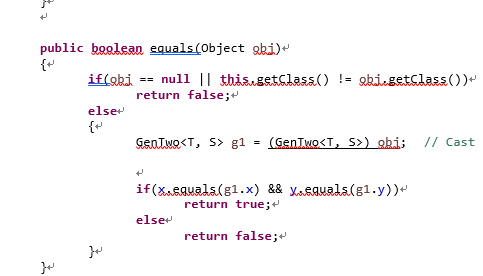
而三比较cm1的时候没用car equal，四比较的时候用了car equal

为什么3没用car equal

再看一遍3



他的参数是car.



而这里的x类型就是T，就不能激活3的car equals

Generic class不能作为exception class

不可以创造一个generic class 有着

，或是throwable的孙类

永远不行

Parameter type的限制

有时候限制可以plug in 的possible type

例如，只有implement interface的class可以plug in T



又比如，Vehicle的孙类

不是孙类的强行插入，会产生compiler error

这里的extends不是继承，而是限制的意思



如果同时Implement多个interface就extends+&

Interface也可以有一个或多个type parameter

细节就和class一样

当一个generic class被描述的时候，type parameter可以被用在method的描述里

另外generic method可以有自己的parameter，这个type parameter不能与class的type parameter重合

一个generic method可以是普通class的member

或有着其他type parameter的generic class的member

Generic method的type method是local的，对class没用

普通class里面带T的方法

**class** Cls1

{

**public** Cls1()

{

// Nothing for now

}

// A generic method that takes an array of some type and a value (actually an object) of that type.

// The method searches the array for that value and returns that passed value if found.

// The method returns null if the value is not found.

**public** <T> T searchArr(T[] A, T val)

{

**int** i;

**for**(i = 0; i < A.length; i++)

{

**if** (A[i].equals(val))

**return** val;如果equal,return这个val，这就可以搜索到了

}

**return** **null**;不然就return null

}

}

尖括号必须在所有前缀后，return type前



使用的格式



**public** <T> T searchArr(T[] A, T val)

{

**int** i;

**for**(i = 0; i < A.length; i++)

{

**if** (A[i].equals(val))

**return** val;如果equal,return这个val，这就可以搜索到了

}

**return** **null**;不然就return null

}

}

一个generic class可以是普通class的子类或是另外一个generic class的子类

如果是普通class，subclass type子集的object也是superclass type全集

由两个classAB,和generic class G

没有任何关系，哪怕B是A的subclass

x = cl1.<Integer>searchArr(Arr1, i1); // Notice that the following is also okay: x = c1.searchArr(Arr1, i1);

x是integer版的cl1，找i1

