

java初学者实践教程11 - 泛型声明 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/270/2021_2022_java_E5_88_9D_E5_AD_A6_c67_270392.htm 泛型声明就是泛泛的声明类型

。我们用其它的语言做一个比较：Javascript声明变量：`var i=1.var c = ' char '` VB 声明变量：`dim i=1.dim c= ' char '` Perl 声明变量：`$i = 1.$c = ' char '` 这些脚本语言，在声明变量的时候，根本就不想java那样还得声明类型。他们的类型已经自己声明完了，是泛泛的声明的，这些语言本身就是泛型。

因为数据类型可能会改变，所以用不着像java定的那么死。但是数据类型可不是说变就变的，java的强类型机制，保证了逻辑的严谨性，而且确保着程序员犯错误，这是java的优点。同时使得它开发起来没有上述语言那样简单，一点小事上做起来都很笨拙。这是其中一个方面，另一个方面如我们上节讲的java的集合类里面的类型是不确定的，放什么都行啊。这样的话你明明自己放进去的类型，也就是说你自己已经知道类型了，拿出来时候，还得不断的转换。我们在介绍拆箱与装箱机制的时候已经说过这个类型的问题了。拆箱与装箱确实能解决不少问题但那是不够的。所以接着上节课学集合类的劲头，趁热打铁。我们讲一下刚才说的第二个方面，关于集合类的问题。我们刚才说，java这种类型安全之下的破绽，我们要用泛型的方式来弥补。我们来实践一个例子。实践：

```
import java.io.IOException.import java.io.PrintStream.import
java.util.HashMap.import java.util.Iterator.import
java.util.LinkedList.import java.util.List.import java.util.Map.public
class GenericsTester { public void testTypeSafeMaps(PrintStream
```

```

out) throws IOException { Map squares = new HashMap(). for (int
i=0. i squares.put(i, i*i). } for (int i=0. i int n = i*3. out.println(n "的
平方是" squares.get(n)). } }//测试安全的链表 public void
testTypeSafeLists(PrintStream out) throws IOException { List
listOfStrings = getListOfStrings(). for (Iterator i =
listOfStrings.iterator(). i.hasNext(). ) { String item = (String)i.next().
} List onlyStrings = new LinkedList(). onlyStrings.add("Legal
addition"). /** * Uncomment these two lines for an error
onlyStrings.add(new StringBuilder("Illegal Addition")).
onlyStrings.add(25). */ } public void
testTypeSafeIterators(PrintStream out) throws IOException { //初始
化迭代 List listOfStrings = new LinkedList().
listOfStrings.add("Happy"). listOfStrings.add("Birthday").
listOfStrings.add("To"). listOfStrings.add("You"). for (Iterator i =
listOfStrings.iterator(). i.hasNext().) { String s = i.next().
out.println(s). } printListOfStrings(getListOfStrings(), out). }//得到
普通链表 private List getList() { List list = new LinkedList().
list.add(3). list.add("Blind"). list.add("Mice"). return list. } //得到安
全的链表 private List getListOfStrings() { List list = new
LinkedList(). list.add("Hello"). list.add("World"). list.add("How").
list.add("Are"). list.add("You?"). return list. } public void
testTypeSafeReturnValues(PrintStream out) throws IOException {
List strings = getListOfStrings(). for (String s : strings) {
out.println(s). } } //接受参数化类型的链表 private void
printListOfStrings(List list, PrintStream out) throws IOException {
for (Iterator i = list.iterator(). i.hasNext(). ) { out.println(i.next()). }

```

```
} public void printList(List list, PrintStream out) throws IOException
{ for (Iterator i = list.iterator(). i.hasNext(). ) {
out.println(i.next().toString()). } } public static void main(String[]
args) { GenericsTester tester = new GenericsTester(). try {
tester.testTypeSafeLists(System.out).
tester.testTypeSafeMaps(System.out).
tester.testTypeSafeIterators(System.out).
tester.testTypeSafeReturnValues(System.out). List ints = new
LinkedList(). ints.add(1). ints.add(2). ints.add(3).
tester.printList(ints, System.out). } catch (Exception e) {
e.printStackTrace(). } } }
```

100Test 下载频道开通，各类考试题目
直接下载。详细请访问 www.100test.com