



链滴

Java Lambda表达式初探

作者: [zjhch123](#)

原文链接: <https://ld246.com/article/1462763984097>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

<p>上课时没事干，随便入门了一下JDK8的新特性，Lambda表达式。 </p>

<p>在Python，Swift里都有很多的用到Lambda。 </p>

<p>Java里引入的Stream()对象还是不怎么理解。 </p>

<p>上代码记录一下</p>

```
<pre class="prettyprint">import java.io.File;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Collections;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.stream.Collectors;
<p>public class Lambda1 {<br>
public static void main(String[] args) {<br>
File file = new File("/usr");<br>
File[] directions = file.listFiles((File f) -&gt; f.isDirectory());</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">
for (File f : directions) {
</span></span><span class="highlight-line"><span class="highlight-cl">    System.out.pr
ntln(f.getName());
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> List&lt;Int
eger&gt; list = Arrays.asList(0, 1, 2, 3, 4, 5, 6, 7, 8, 9);
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> System.out.print
n("*****不用Stream筛选*****");
</span></span><span class="highlight-line"><span class="highlight-cl"> List&lt;Int
eger&gt; newList = new ArrayList&lt;Integer&gt;();
</span></span><span class="highlight-line"><span class="highlight-cl"> list.forEach(i -&
mp;gt; {
</span></span><span class="highlight-line"><span class="highlight-cl">     if (i % 2 == 0)
{
</span></span><span class="highlight-line"><span class="highlight-cl">         newList.ad
d(i);
</span></span><span class="highlight-line"><span class="highlight-cl">     }
</span></span><span class="highlight-line"><span class="highlight-cl"> });
</span></span><span class="highlight-line"><span class="highlight-cl"> System.out.print
n(newList);
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> System.out.print
n("*****用Stream筛选*****");
</span></span><span class="highlight-line"><span class="highlight-cl"> List&lt;Int
eger&gt; newList2 = list.stream().filter(i -&mp;gt; i % 2 == 0).collect(Collectors.toList());
</span></span><span class="highlight-line"><span class="highlight-cl"> System.out.print
n(newList2);
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> System.out.print
n("*****不用Stream排序*****");
</span></span><span class="highlight-line"><span class="highlight-cl"> Collections.sort(
list, (o1, o2) -&mp;gt; o2.compareTo(o1));
</span></span><span class="highlight-line"><span class="highlight-cl"> System.out.print
n(list);
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span></pre>
```

```

</span></span><span class="highlight-line"><span class="highlight-cl"> System.out.print
n("*****用Stream排序*****");
</span></span><span class="highlight-line"><span class="highlight-cl"> System.out.print
n(list.stream().sorted((o1,o2) -&gt; o1.compareTo(o2)).collect(Collectors.toList()));
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> System.out.print
n("*****用Stream映射*****");
</span></span><span class="highlight-line"><span class="highlight-cl"> List<Integer>
sortList = list.stream().map(i -> {
</span></span><span class="highlight-line"><span class="highlight-cl">     return i + 3;
</span></span><span class="highlight-line"><span class="highlight-cl"> }).collect(Collectors.toList());
</span></span><span class="highlight-line"><span class="highlight-cl"> System.out.print
n(sortList);
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> Map<Integer, String>
map = new HashMap<Integer, String>();
</span></span><span class="highlight-line"><span class="highlight-cl"> map.put(1, "on
");
</span></span><span class="highlight-line"><span class="highlight-cl"> map.put(2, "tw
");
</span></span><span class="highlight-line"><span class="highlight-cl"> map.put(3, "thr
e");
</span></span><span class="highlight-line"><span class="highlight-cl"> map.put(4, "fou
");
</span></span><span class="highlight-line"><span class="highlight-cl"> map.put(5, "five
");
</span></span><span class="highlight-line"><span class="highlight-cl"> map.put(6, "six
");
</span></span><span class="highlight-line"><span class="highlight-cl"> map.put(7, "sev
n");
</span></span><span class="highlight-line"><span class="highlight-cl"> System.out.print
n("*****不用Stream遍历map*****");
</span></span><span class="highlight-line"><span class="highlight-cl"> map.forEach( (k
v) -&gt; System.out.println(k + "|" + map.get(k)));
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span></code></pre>
<p><br>
</p></pre>
<p></p>
<p>&nbsp;</p>

```