

java匿名内部类的使用场景(2)计算机等级考试 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/645/2021_2022_java_E5_8C_BF_E5_90_8D_c97_645477.htm 这样，一个模板类就完成了。

我们也通过这个模板类将catch...finally...两段代码提出来了。我们来看看使用了这个模板类的数据层方法是怎么实现的：

```
new DataTemplate().execute(new DataManager() { public void
manageData() { String[] divisionData = null. conn =
manager.getInstance().getConnection(). stmt =
(OracleCallableStatement)conn.prepareCall("{ Call
PM_GET_PRODUCT.HEADER_DIVISION(?, ?) }").
stmt.setLong(1 ,productId.longValue() ).
stmt.registerOutParameter(2, oracle.jdbc.OracleTypes.CURSOR).
stmt.execute(). ResultSet rs = stmt.getCursor(2). int i = 0 . String
strDivision = "". while( rs.next() ) { strDivision =
rs.getString("DIVISION_ID") ", " . } int length = strDivision.length()
. if(length != 0 ) { strDivision = strDivision.substring(0,length - 1). }
divisionData = StringUtil.split(strDivision, ",") . map.put("Division",
strDivision ) .
```

```
LoggerAgent.debug("GetHeaderProcess","getDivisionData","getVal
ue " strDivision " " productId) . } } ). 注意：本段代码仅供思路上的
参考，没有经过上机测试。我们可以看到，正是这个实现了
DataManager接口得匿名内部类的使用，才使得我们解决了对try...catch...finally...语句的改造。这样，第一为我们解决了
令人痛苦的重复代码；第二也让我们在数据层方法的编码中
，直接关注对数据的操作，不用关心那些必需的但是与数据
```

操作无关的东西。我们现在来回想一下Spring框架的数据层，是不是正是使用了这种方法呢？场景之三：一些多算法场合假如我们有这样一个需求：我们的一个方法用来对数组排序并且依次打印各元素，对数组排序方法有很多种，用哪种方法排序交给用户自己确定。对于这样一个需求，我们很容易解决。我们决定给哪些排序算法定义一个接口，具体的算法实现由用户自己完成，只要求他实现我们的接口就行。

```
public interface SortAlgor { public void sort(int[] is). }
```

这样，我们再在方法里实现先排序后打印，代码如下：

```
public void printSortedArray(int[] is,SortAlgor sa) { ..... sa.sort(is). for(int i=0;i<is.length;i++) System.out.println(is[i]); }
```

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