

國立臺北科技大學 資訊工程系 碩士口試之問題回覆

論文題目	利用Robusta消除例外處理壞味道
口試日期	107/06/14
研究生	楊雅雯
指導教授	謝金雲教授、鄭有進教授

問題與回覆：

問題一	<p>Exception Thrown From Finally Block壞味道重構後，是否以Logger的方式取代e.printStackTrace()更合適？</p> <pre> 301 public static void saveChartAsPNG(File file, JFreeChart chart, 302 int width, int height, ChartRenderingInfo info) 303 throws IOException { 304 305 Args.nullNotPermitted(file, "file"); 306 OutputStream out = new BufferedOutputStream(new FileOutputStream(file)); 307 try { 308 ChartUtils.writeChartAsPNG(out, chart, width, height, info); 309 } 310 finally { 311 Close(out); 312 } 313 } 314 315 private static void Close(OutputStream out) { 316 try { 317 out.close(); 318 } catch (IOException e) { 319 /* Although it's a Dummy Handler bad smell, 320 * Robusta recommends that you keep this bad smell 321 * instead of choosing Quick Fix or Refactor that we provide. 322 */ 323 e.printStackTrace(); 324 } 325 } </pre> <p>圖1、原來Careless Cleanup消除後結果</p>
問題回覆	<p>是，如下圖所示，已經將e.printStackTrace()更新為Logger的方式，此壞味道重構後的相關敘述在論文中3.1.5小節有詳細的說明。</p> <pre> 11 private Logger logger = Logger.getLogger(ExceptionThrownFromFinallyBlock.class.getName()); 12 13 public void exceptionThrownFromFinallyBlockExample() throws IOException { 14 FileInputStream fileInputStream = null; 15 try { 16 fileInputStream = new FileInputStream("test.txt"); 17 fileInputStream.read(); 18 } catch (IOException e) { 19 throw new IOException("Exception thrown in catch"); 20 } finally { 21 Close(fileInputStream); 22 } 23 } 24 25 private void Close(FileInputStream fileInputStream) { 26 try { 27 fileInputStream.close(); 28 } catch (IOException e) { 29 /* Although it's a Dummy Handler bad smell, 30 * Robusta recommends that you keep this bad smell 31 * instead of choosing Quick Fix or Refactor that we provide. 32 */ 33 PropertyConfigurator.configure("log4j.properties"); 34 logger.info(e); 35 } 36 } </pre>

問題二	在這張表格中，你的主要貢獻在哪裡？ 為什麼沒有在案例分析中舉例？		
	壞味道種類	原本的Robusta	本論文改善或提供的方法
	Dummy Handler & Empty Catch Block	1. 快速修復功能：丟出例外 2. 快速修復功能：丟出 <u>RuntimeException</u> 。	快速修復功能：丟出例外
	Unprotected Main Program	產生try/catch保護主程式，且catch捕捉Exception類別後不做任何事。	產生try/catch保護主程式，catch捕捉 <u>Throwable</u> 類別且印出例外訊息和將例外訊息寫入日誌中。
	Careless Cleanup	無	增加快速修復功能。
問題回覆	我的主要貢獻為增加Careless Cleanup的快速修復功能，沒有在demo中舉例是因為當初在設計ppt時沒有思考清楚而忽略了，詳細的舉例在論文4.2節有詳細的說明。		

建議一	<p>Careless Cleanup 和 Exception Thrown From Finally Block壞味道快速修復或重構後，建議以Logger的方式取代e.printStackTrace()</p> <pre> 167 public void encode(BufferedImage bufferedImage, OutputStream outputStream) throws IOException { 168 Iterator iterator = ImageIO.getImageWritersByFormatName("jpeg"); 169 ImageWriter writer = (ImageWriter) iterator.next(); 170 ImageWriteParam p = writer.getDefaultWriteParam(); 171 ImageOutputStream ios = ImageIO.createImageOutputStream(outputStream); 172 try { 173 Args.notNullNotPermitted(bufferedImage, "bufferedImage"); 174 Args.notNullNotPermitted(outputStream, "outputStream"); 175 p.setCompressionMode(ImageWriteParam.MODE_EXPLICIT); 176 p.setCompressionQuality(this.quality); 177 writer.setOutput(ios); 178 writer.write(null, new IIOMemorySource(bufferedImage, null, null), p); 179 ios.flush(); 180 writer.dispose(); 181 } finally { 182 try { 183 if(ios != null) 184 ios.close(); 185 } catch (IOException e) { 186 /* 187 * Although it's a Dummy Handler bad smell, 188 * Robusta recommends that you keep this bad smell 189 * instead of choosing Quick Fix or Refactor that we provide. 190 */ 191 e.printStackTrace(); 192 } 193 } 194 } </pre> <p>圖1、原來Careless Cleanup消除後結果</p> <pre> 301 public static void saveChartAsPNG(File file, JFreeChart chart, 302 int width, int height, ChartRenderingInfo info) 303 throws IOException { 304 Args.notNullNotPermitted(file, "file"); 305 OutputStream out = new BufferedOutputStream(new FileOutputStream(file)); 306 try { 307 ChartUtils.writeChartAsPNG(out, chart, width, height, info); 308 } finally { 309 Close(out); 310 } 311 } 312 313 private static void Close(OutputStream out) { 314 try { 315 out.close(); 316 } catch (IOException e) { 317 /* 318 * Although it's a Dummy Handler bad smell, 319 * Robusta recommends that you keep this bad smell 320 * instead of choosing Quick Fix or Refactor that we provide. 321 */ 322 e.printStackTrace(); 323 } 324 } 325 } </pre> <p>圖2、原來Exception Thrown From Finally Block消除後結果</p>
-----	---

問題回覆

1. 如下圖所示，Careless Cleanup快速修復後已將e.printStackTrace()更新為Logger的方式，相關敘述在論文第3.1.4、4.2.3小節有詳細的說明及應用。

```
174 @Override
175 public void encode(BufferedImage bufferedImage, OutputStream outputStream) throws IOException {
176     Iterator iterator = ImageIO.getImageWritersByFormatName("jpeg");
177     ImageWriter writer = (ImageWriter) iterator.next();
178     ImageWriteParam p = writer.getDefaultWriteParam();
179     ImageOutputStream ios = ImageIO.createImageOutputStream(outputStream);
180     try {
181         Args.notNullPermitted(bufferedImage, "bufferedImage");
182         Args.notNullPermitted(outputStream, "outputStream");
183         p.setCompressionMode(ImageWriteParam.MODE_EXPLICIT);
184         p.setCompressionQuality(this.quality);
185         writer.setOutput(ios);
186         writer.write(null, new IIOMemorySource(bufferedImage, null, null), p);
187         ios.flush();
188         writer.dispose();
189     } finally {
190         try {
191             ios.close();
192         } catch (IOException e) {
193             /*
194              * Although it's a Dummy Handler bad smell,
195              * Robusta recommends that you keep this bad smell
196              * instead of choosing Quick Fix or Refactor that we provide.
197              */
198             PropertyConfigurator.configure("log4j.properties");
199             logger.info(e);
200         }
201     }
202 }
```

2. Exception Thrown From Finally Block重構後修改成Logger的方式已經在問題一解決了。

感謝各位口試委員於百忙之中撥空蒞臨口試，並給予詳細的指導及寶貴的建議。

學生 楊雅雯