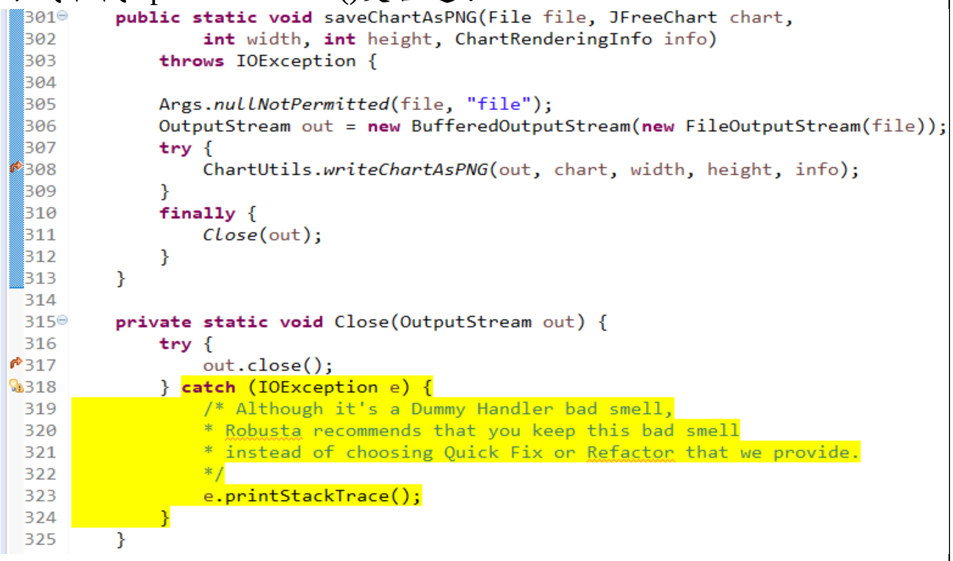
國立臺北科技大學　資訊工程系　碩士口試之問題回覆

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

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？

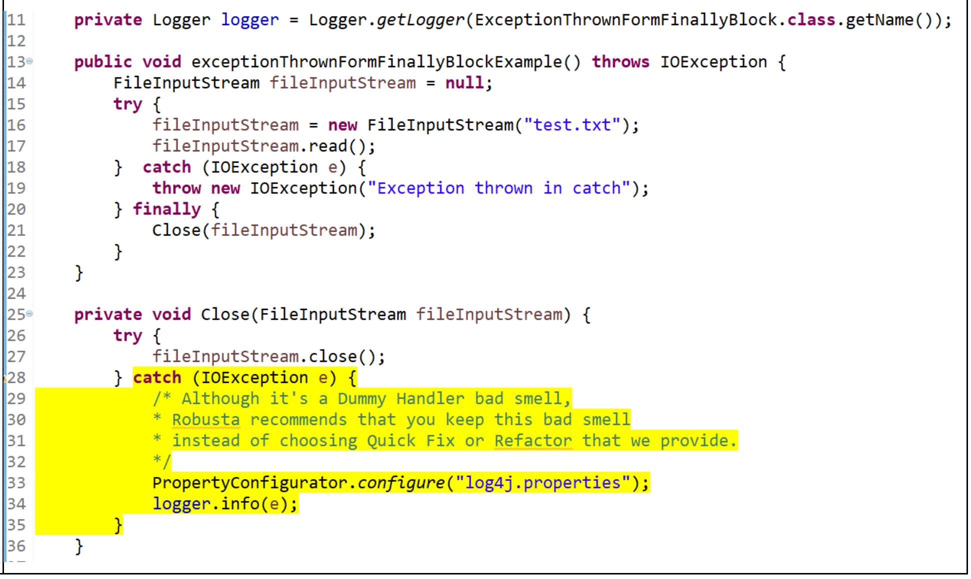


```
    public static void saveChartAsPNG(File file, JFreeChart chart,
            int width, int height, ChartRenderingInfo info)
        throws IOException {

        Args.nullNotPermitted(file, "file");
        OutputStream out = new BufferedOutputStream(new FileOutputStream(file));
        try {
            ChartUtils.writeChartAsPNG(out, chart, width, height, info);
        }
        finally {
            Close(out);
        }
    }

    private static void Close(OutputStream out) {
        try {
            out.close();
        } catch (IOException e) {
            /* Although it's a Dummy Handler bad smell,
             * Robusta recommends that you keep this bad smell
             * instead of choosing Quick Fix or Refactor that we provide.
             */
            e.printStackTrace();
        }
    }
```

圖1、原來Careless Cleanup消除後結果

問題回覆

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。



```
    private Logger logger = Logger.getLogger(ExceptionThrownFormFinallyBlock.class.getName());

    public void exceptionThrownFormFinallyBlockExample() throws IOException {
        FileInputStream fileInputStream = null;
        try {
            fileInputStream = new FileInputStream("test.txt");
            fileInputStream.read();
        }  catch (IOException e) {
            throw new IOException("Exception thrown in catch");
        } finally {
            Close(fileInputStream);
        }
    }

    private void Close(FileInputStream fileInputStream) {
        try {
            fileInputStream.close();
        } catch (IOException e) {
            /* Although it's a Dummy Handler bad smell,
             * Robusta recommends that you keep this bad smell
             * instead of choosing Quick Fix or Refactor that we provide.
             */
            PropertyConfigurator.configure("log4j.properties");
            logger.info(e);
        }
    }
```

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？

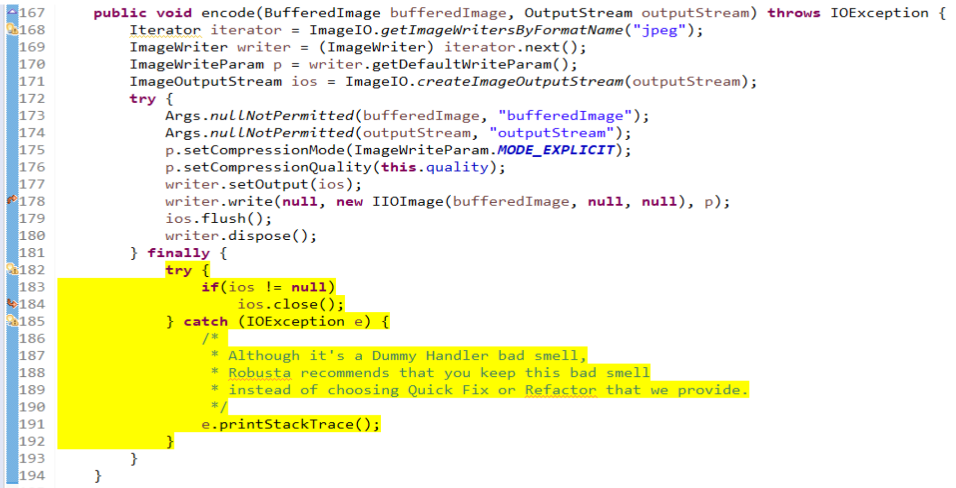
壞味道種類	原本的Robusta	本論文改善或提供的方法
Dummy Handler & Empty Catch Block	1. 快速修復功能：丟出例外 2. 快速修復功能：丟出RuntimeException。	快速修復功能：丟出例外
Unprotected Main Program	產生try/catch保護主程式，且catch捕捉Exception類別後不做任何事。	產生try/catch保護主程式，catch捕捉Throwable類別且印出例外訊息和將例外訊息寫入日誌中。
Careless Cleanup	無	增加快速修復功能。

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。

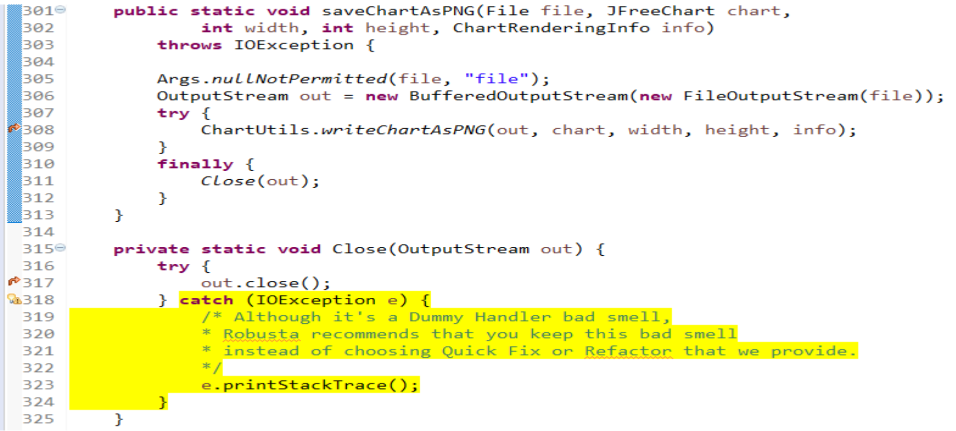
建議一

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()



```java
public void encode(BufferedImage bufferedImage, OutputStream outputStream) throws IOException {
    Iterator iterator = ImageIO.getImageWritersByFormatName("jpeg");
    ImageWriter writer = (ImageWriter) iterator.next();
    ImageWriteParam p = writer.getDefaultWriteParam();
    ImageOutputStream ios = ImageIO.createImageOutputStream(outputStream);
    try {
        Args.nullNotPermitted(bufferedImage, "bufferedImage");
        Args.nullNotPermitted(outputStream, "outputStream");
        p.setCompressionMode(ImageWriteParam.MODE_EXPLICIT);
        p.setCompressionQuality(this.quality);
        writer.setOutput(ios);
        writer.write(null, new IIOImage(bufferedImage, null, null), p);
        ios.flush();
        writer.dispose();
    } finally {
        try {
            if(ios != null)
                ios.close();
        } catch (IOException e) {
            /*
             * Although it's a Dummy Handler bad smell,
             * Robusta recommends that you keep this bad smell
             * instead of choosing Quick Fix or Refactor that we provide.
             */
            e.printStackTrace();
        }
    }
}
```

圖1、原來Careless Cleanup消除後結果

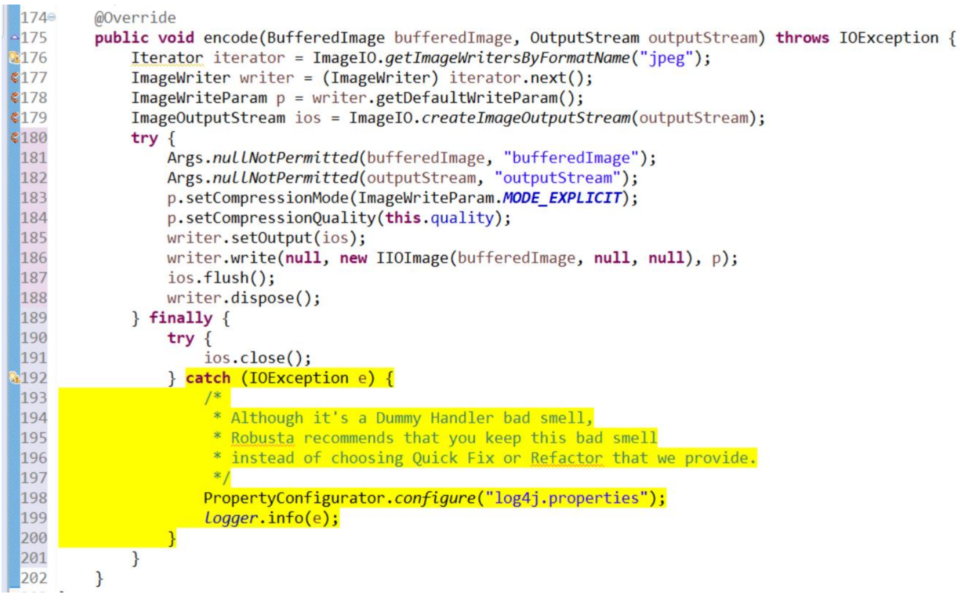


```java
public static void saveChartAsPNG(File file, JFreeChart chart,
        int width, int height, ChartRenderingInfo info)
    throws IOException {

    Args.nullNotPermitted(file, "file");
    OutputStream out = new BufferedOutputStream(new FileOutputStream(file));
    try {
        ChartUtils.writeChartAsPNG(out, chart, width, height, info);
    }
    finally {
        Close(out);
    }
}

private static void Close(OutputStream out) {
    try {
        out.close();
    } catch (IOException e) {
        /* Although it's a Dummy Handler bad smell,
        * Robusta recommends that you keep this bad smell
        * instead of choosing Quick Fix or Refactor that we provide.
        */
        e.printStackTrace();
    }
}
```

圖2、原來Exception Thrown From Finally Block消除後結果

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。



```
@Override
public void encode(BufferedImage bufferedImage, OutputStream outputStream) throws IOException {
    Iterator iterator = ImageIO.getImageWritersByFormatName("jpeg");
    ImageWriter writer = (ImageWriter) iterator.next();
    ImageWriteParam p = writer.getDefaultWriteParam();
    ImageOutputStream ios = ImageIO.createImageOutputStream(outputStream);
    try {
        Args.nullNotPermitted(bufferedImage, "bufferedImage");
        Args.nullNotPermitted(outputStream, "outputStream");
        p.setCompressionMode(ImageWriteParam.MODE_EXPLICIT);
        p.setCompressionQuality(this.quality);
        writer.setOutput(ios);
        writer.write(null, new IIOImage(bufferedImage, null, null), p);
        ios.flush();
        writer.dispose();
    } finally {
        try {
            ios.close();
        } catch (IOException e) {
            /*
             * Although it's a Dummy Handler bad smell,
             * Robusta recommends that you keep this bad smell
             * instead of choosing Quick Fix or Refactor that we provide.
             */
            PropertyConfigurator.configure("log4j.properties");
            logger.info(e);
        }
    }
}
```

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