|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Code Review紀錄內容 | | | | 更新日期 : 2024/08/13 |
| 日期 | CR人員 | 題號 | 備註 | |
| 2024/08/08 | 吳恬安 | 第1題 |  | |
| 1. 每列乘積須向右對齊 | 修改前：    修改後：  printf可使用%2d限定輸出為兩位數，若未滿兩位則向左補空白  也可自訂正負位數，如%10d、%-5d | | | |
| 2024/08/08 | 吳恬安 | 第1題 |  | |
| 1. 印出含多個型別的字串用printf | 在print內轉型太多，易讀性低，且3個以上字串串接用StringBuilder但因有多型態也可使用printf  修改前：    修改後： | | | |
| 2024/08/08 | 吳恬安 | 第1題 |  | |
| 1. 排版格式化 | 存檔前都宜再排版一次  修改前：    修改後： | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 2024/08/08 | 吳恬安 | 第2題 | 備註08/09)：修改前圖上紅線處可省略 |
| 1. 迴圈跑六次可能不會拿到六個值 | 由於重複值判定在for迴圈中，若重複就不加入lottoList，因此若有重複值發生，會跑完判斷但少一個值輸入。應將for迴圈跑六次改成while(lottoList.size() < 6){…}，以實際放滿六個值入lottoList為判斷標準  修改前：    修改後： | | |
| 2024/08/08 | 吳恬安 | 第2題 |  |
| 1. rand放在迴圈內重複宣告 | 重複宣告造成記憶體使用過多，且因Java內建GC(Garbage Collection)收集不常被使用的記憶體到定量才自動清除，使用者不可控，因此應避免重複宣告  修改前：    修改後： | | |
| 1. Set內不可重複，但不能存入後變更排序 | Set只能放不重複的內容，但因需排序前後打印，無法使用TreeSet(存入後按自然順序排序)，而Set也不能事後更改排序，因此這題維持使用List | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 2024/08/08 | 吳恬安 | 第3題 |  |
| 1. 父類別建構子給子類別用 | 子類別需要的都可以請父類別做，精簡程式  因Supervisor、Sales子類別，Employee父類別都有name、department、salary三個property，可以請父類別做自定義建構子取代default建構子，再從子類別建構子更改default 的 super()，引入參數  使用class內的propery記得用this.語法，且取class內的propery不須使用函數  修改前：    修改後： | | |
| 1. 不必要的程式碼移除 | Interface預設public abstract method，若有property則必為public static final，因此皆可省略  記得刪除TO DO等系統提示註解  修改前：    修改後： | | |
| 2024/08/08 | 吳恬安 | 第4題 |  |
|  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. Excel開啟會有亂碼 | | 修改前： | | |
|  | | 修改後：  新增BOM入file：端序記號，或稱位元組順序記號（byte-order mark，BOM）  端序在UTF-8中沒有任何意義，它在UTF-8中唯一用途是在開始時發出訊號，表明文字流是用UTF-8編碼  新增OutputStreamWriter(轉化流)、更改FileWriter成FileOutputStream(只需處理UTF-8)  數據從內存寫入硬盤的流程：緩衝流 > 轉化流 > 文件流 | | |
| 2024/08/08 | | 吳恬安 | 第4題 |  |
| 1. try-catch程式拋錯時程式進行方式 | | 由於write()和flush()都是拋IOException，若想分清出錯環節，依此方法馬上try-catch加上例外訊息才可分辨  但若拋不同種例外，則用try-catch-catch..抓取由小到大的Exception  測試1：迴圈內寫入緩衝區的try丟Exception  迴圈內寫入緩衝區的try跑i == 0、1皆成功寫入，直到i == 2 丟 Exception e，馬上被寫入緩衝區的catch抓取並執行例外處理(打印例外訊息)，已跳出i == 2寫入緩衝區的try，但仍在寫手宣告的try中for迴圈裡，因此會繼續跑迴圈 i == 3直到程式結束      測試2：寫手宣告的try丟Exception  以下圖為例，加入i ==2測試在寫手宣告try-catch丟Exception。  寫手宣告try內迴圈跑i == 0、1皆成功寫入，直到i == 2 丟 Exception e，馬上被寫手宣告的catch抓取並執行例外處理(打印例外訊息)，因已跳出寫手宣告的try，不會再次進入for迴圈。 | | |
| 1. 位址字串可用/代替\\ | | 不可寫單斜線\，由於是提示字元(\t、\n…)，因此若需要在字串內打出斜線\，應打\\  修改前：    修改後： | | |
| 2024/08/12 | | 吳恬安 | 第1題 |  |
| 1. Product變數只會用到一次，無須宣告 | | 另外，若counter無意義且被多次應用，可直接將counter命名成i, j  修改前：    修改後： | | |
| 2024/08/12 | | 吳恬安 | 第5題 |  |
| 1. 置中空格可納入%s格式 | | 修改前：    修改後： | | |
| 1. 需增加防呆訊息 | | 修改前：產生錯誤訊息，無法繼續執行    修改後：增加try-catch-finally | | |
| 2024/08/12 | | 吳恬安 | 第6題 |  |
| 1. 輸出/入檔路徑拿出main寫成public static final | | 修改前：    修改後： | | |
| 1. 刪除成功log訊息 | | 太多try-catch，程式完成後應刪除確定不出錯的try-catch，並保留使用者可能出錯的try-catch，且僅保留其中的失敗訊息  修改前：    修改後： | | |
| 1. BOM拉入write區塊 | | BOM區塊也涵蓋在write過程中，應統一區塊一同偵錯  修改前：    修改後： | | |
| 1. BigDecimal.ZERO 用法 | | 修改前：    修改後： | | |
| 1. 考量針對未知表格的執行方式 | | 此處限定nextManufacturer成Acura，設定成已知的第一筆資料的Manufacturer以比對和下一筆的Manufacturer是否相同。但若應用於未知的表格，則應更改寫法成拿取tableList內第一筆Map的Manufacturer  修改前：    修改後： | | |
| 1. 可將表頭先取出，減少後續排序問題 | | 由於整題須依Price、Manufacturer分別排序，表頭若也參與排序容易造成轉檔失敗/排序錯誤，應先取出並另外處理  修改前：需添加tableList.removeLast()移除剛好排在最後的表頭    修改後：將表頭令存入headMap，並與tableList分開操作 | | |
| 1. 避免NullPointer   Exception | | 應將確定非NULL的變數放前面，才有方法可以取  修改前：    修改後： | | |
| 1. 避免重複取值 | | 修改前：上方小框和下方大框為兩種不同取值方法，可統一為一次，並存成變數(因以下有多次使用)    修改後： | | |
| 2024/08/12 | | 吳恬安 | 第7題 |  |
| 1. Username、Password可拿出成public static final | 修改前：    修改後： | | |
| 1. 取ResultSet值時使用欄位名 | 避免使用數字index：若表格變更順序，取值時無須更改  修改前：    修改後： | | |
| 1. MIN\_PRICE和PRICE都設成Double | 原始表格訂為number(19, 2)：(有效位,小數位)，因此應該設定為double而非integer  修改前：    修改後： | | |
| 1. 縮減switch內不同case的重複行為 | CR當下建議將紅框處合併manufacturer和type的宣告與賦值於Line30，以減少switch內不同case的重複程式碼，但由於switch添加防呆default case，若在上方宣告賦值會在上方顯示出不需要的輸入訊息(應先顯示指令錯誤訊息)，因此決定添加方法以省略manufacturer和type的宣告賦值  修改前：    修改後： | | |
| 1. 增加防呆 | select：    delete：      update：      insert： | | |