Java 開發規範

規範項目：

Coding Style

1. 務必排版格式化
2. 移除無用程式碼（變數 / 測試資料等）
3. 變數相關
    - 名稱須有意義並符合駝峰或大寫底線
    - 變數不用特別給初值，統一由判斷賦值
4. 優先使用字元
5. public 方法傳入參數為 4 個以上改傳 Map
6. 僅自己類別用到的方法應宣告為 private
7. private 方法應置於 public 方法之後

安全性

1. 對資料庫新刪修前，請先確認資料庫資料狀況，避免資料庫報錯
2. 不要直接異動傳入物件參數，須先 clone，例如：傳入方法的 List 或 Map 參數。
3. 異動 DB 一律要做 Transaction 交易控制
4. 輸出物件應關閉資源或使用 try-with-resource
5. 僅 throw 系統建議之 Exception
6. 避免 NullPointerException
7. 金額計算使用 BigDecimal

效能

1. 變數相關
    - 變數有效範圍最小化 (減少全域變數)
    - 變數要使用前再宣告
    - 重複用方法取值應宣告為變數
    - 只使用一次可不用宣告為變數
    - 固定常數用 private static final 宣告為成員變數
    - 與迴圈內資料無關的變數可於迴圈外宣告共用
2. 重複程式碼拆出為方法共用
3. 若有 3 個以上字串相加應使用 StringBuilder

優化項目：

Coding Style

1. 善用反向判定就 return 提早結束程式碼

功能有誤

1. String 相比用 equals

效能

1. 參數/資料不要過度包裝
2. 減少不必要資料參數傳遞
3. 未滿三個字串或無變數串接需求不用使用 StringBuilder
4. 善用泛型
5. 使用泛型需定義型別
6. 建立空 Map，善用 Collections.EMPTY_MAP
7. 善用類別提供的內建常數（BigDecimal.Zero）
8. 善用已宣告的 class 物件 (ex. sb)

程式設計優化

1. 善用三元運算子
2. 善用增強式 for 迴圈:foreach
3. 優化資料處理（減少 query 與 for 迴圈次數）