軟體測試 HW02 報告

D1149576 曹世杰

程式運作：

1、 將 JSON 檔導入成 JSONObject 再轉成 JSONArray 之後再存入自定義 Class 中。

2、 **最差收入：**

排名若為 1 或 2，則找出同分區另外兩隊座位*滿席率最低者並打最少場次戰敗，連敗三場。

排名若為 3 ~ 6，則最差收入為連敗兩場。

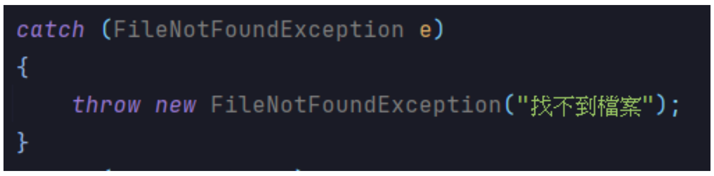
最佳收入：

排名若為 1 或 2，分區是找出同分區另外兩隊座位*滿席率最高者並打滿場次後勝利。聯盟冠軍為從另一分區找出座位*滿席率最高者並打滿場次後勝利。世界冠軍是從另一區域找出座位*滿席率最高者並打滿場次後勝利。

排名若為 3 ~ 6，外卡時三場打滿後勝利，分區則與種子隊打滿場次後勝利。聯盟冠軍為從另一分區找出座位*滿席率最高者並打滿場次後勝利。世界冠軍是另一區域找出座位*滿席率最高者並打滿場次後勝利。

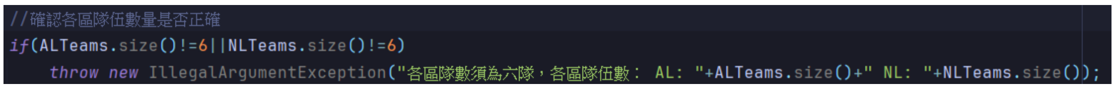
自己設計：

1、 讀取檔案的程式碼放在 try-catch 中，如果找不到檔案則會跳 Exception 並通知使用者。



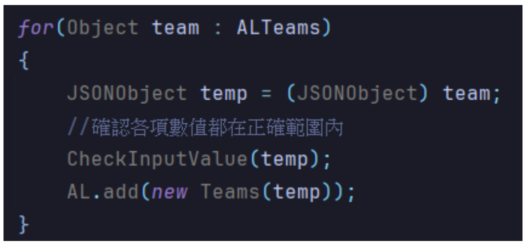
```java
catch (FileNotFoundException e)
{
    throw new FileNotFoundException("找不到檔案");
}
```

2、 在導入 JSON 完成後，會把資料轉換進程式中的 Teams 的自定義型態，途中會檢查導入的隊伍數量是否為各 6 隊，如果不是會跳出 Exception 並告知讀取到的各區的隊伍數量。



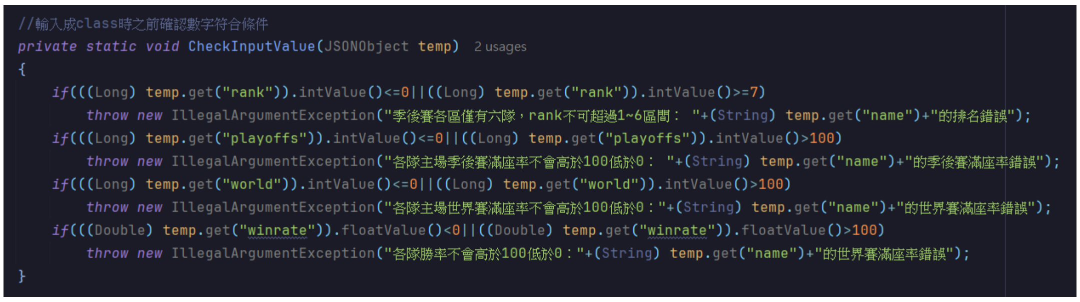
```java
//確認各區隊伍數量是否正確
if(ALTeams.size()!=6||NLTeams.size()!=6)
    throw new IllegalArgumentException("各區隊數須為六隊，各區隊伍數： AL: "+ALTeams.size()+" NL: "+NLTeams.size());
```

3、 將資料加進型態陣列前會先呼叫函式檢查輸入的值是否在要求的區間以內。



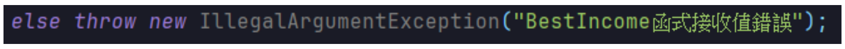
```java
for(Object team : ALTeams)
{
    JSONObject temp = (JSONObject) team;
    //確認各項數值都在正確範圍內
    CheckInputValue(temp);
    AL.add(new Teams(temp));
}
```

只要有任何值不屬於範圍內都會跳出 Exception 並告知是哪一隊的資料錯誤。



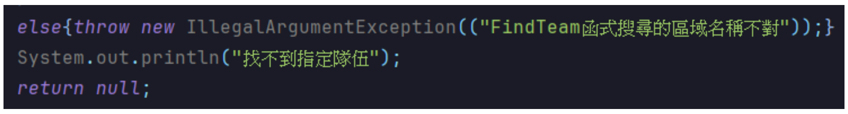
```java
//輸入成class時之前確認數字符合條件
private static void CheckInputValue(JSONObject temp)  2 usages
{
    if(((Long) temp.get("rank")).intValue()<=0||((Long) temp.get("rank")).intValue()>=7)
        throw new IllegalArgumentException("季後賽各區僅有六隊，rank不可超過1~6區間： "+(String) temp.get("name")+"的排名錯誤");
    if(((Long) temp.get("playoffs")).intValue()<=0||((Long) temp.get("playoffs")).intValue()>100)
        throw new IllegalArgumentException("各隊主場季後賽滿座率不會高於100低於0： "+(String) temp.get("name")+"的季後賽滿座率錯誤");
    if(((Long) temp.get("world")).intValue()<=0||((Long) temp.get("world")).intValue()>100)
        throw new IllegalArgumentException("各隊主場世界賽滿座率不會高於100低於0："+(String) temp.get("name")+"的世界賽滿座率錯誤");
    if(((Double) temp.get("winrate")).floatValue()<0||((Double) temp.get("winrate")).floatValue()>100)
        throw new IllegalArgumentException("各隊勝率不會高於100低於0："+(String) temp.get("name")+"的世界賽滿座率錯誤");
}
```

4、 在計算最佳收益和最差收益的區域判別中，如果接收值沒辦法被函式拿來使用就會跳出 Exception 並告知使用者此函式的接收值可能會錯誤的。



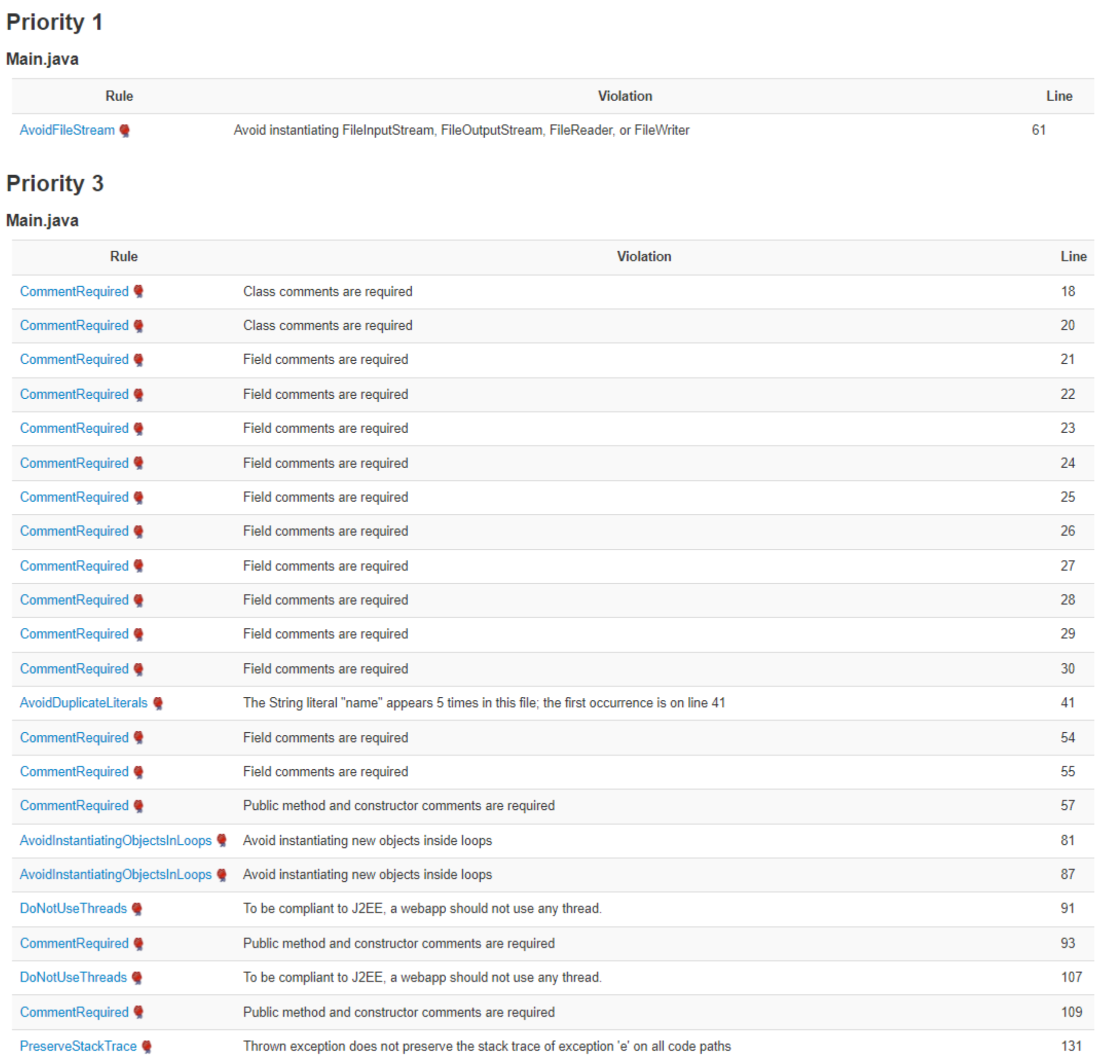
```java
else throw new IllegalArgumentException("BestIncome函式接收值錯誤");
```

5、 在使用 FindTeam 函式時若傳入值的區域名稱不對或是找不到對應條件的隊伍都會跳出提示。



```java
else{throw new IllegalArgumentException(("FindTeam函式搜尋的區域名稱不對"));}
System.out.println("找不到指定隊伍");
return null;
```

POM：



Priority 1

Main.java

Rule	Violation	Line
AvoidFileStream	Avoid instantiating FileInputStream, FileOutputStream, FileReader, or FileWriter	61

Priority 3

Main.java

Rule	Violation	Line
CommentRequired	Class comments are required	18
CommentRequired	Class comments are required	20
CommentRequired	Field comments are required	21
CommentRequired	Field comments are required	22
CommentRequired	Field comments are required	23
CommentRequired	Field comments are required	24
CommentRequired	Field comments are required	25
CommentRequired	Field comments are required	26
CommentRequired	Field comments are required	27
CommentRequired	Field comments are required	28
CommentRequired	Field comments are required	29
CommentRequired	Field comments are required	30
AvoidDuplicateLiterals	The String literal "name" appears 5 times in this file; the first occurrence is on line 41	41
CommentRequired	Field comments are required	54
CommentRequired	Field comments are required	55
CommentRequired	Public method and constructor comments are required	57
AvoidInstantiatingObjectsInLoops	Avoid instantiating new objects inside loops	81
AvoidInstantiatingObjectsInLoops	Avoid instantiating new objects inside loops	87
DoNotUseThreads	To be compliant to J2EE, a webapp should not use any thread.	91
CommentRequired	Public method and constructor comments are required	93
DoNotUseThreads	To be compliant to J2EE, a webapp should not use any thread.	107
CommentRequired	Public method and constructor comments are required	109
PreserveStackTrace	Thrown exception does not preserve the stack trace of exception 'e' on all code paths	131

UnusedPrivateMethod	Avoid unused private methods such as 'TestArray()'.	144
UnusedAssignment	The initializer for variable 'world_Team' is never used (overwritten on lines 319 and 455)	196
LiteralsFirstInComparisons	Position literals first in String comparisons	199
AvoidLiteralsInIfCondition	Avoid using literals in if statements	205
AvoidLiteralsInIfCondition	Avoid using literals in if statements	210
AvoidLiteralsInIfCondition	Avoid using literals in if statements	217
AvoidLiteralsInIfCondition	Avoid using literals in if statements	261
AvoidLiteralsInIfCondition	Avoid using literals in if statements	266
AvoidLiteralsInIfCondition	Avoid using literals in if statements	273
AvoidDuplicateLiterals	The String literal "\u6700\u4f73\u6536\u5165\u904e\u7a0b(\u5404\u968e\u6bb5\u6bd4\u8cfd\u7686\u6253\u6eff)\uff1a\u7e3d\u6536\u5165\u70ba" appears 4 times in this file; the first occurrence is on line 325	325
AvoidDuplicateLiterals	The String literal ", \u806f\u76df\u51a0\u8ecd\u8cfd\u5c0d\u4e0a" appears 4 times in this file; the first occurrence is on line 326	326
AvoidDuplicateLiterals	The String literal ", \u4e16\u754c\u8cfd\u5c0d\u4e0a" appears 4 times in this file; the first occurrence is on line 326	326
LiteralsFirstInComparisons	Position literals first in String comparisons	336
AvoidLiteralsInIfCondition	Avoid using literals in if statements	342
AvoidLiteralsInIfCondition	Avoid using literals in if statements	347
AvoidLiteralsInIfCondition	Avoid using literals in if statements	353
AvoidLiteralsInIfCondition	Avoid using literals in if statements	397
AvoidLiteralsInIfCondition	Avoid using literals in if statements	402
AvoidLiteralsInIfCondition	Avoid using literals in if statements	409
UnusedLocalVariable	Avoid unused local variables such as 'num'.	479
LiteralsFirstInComparisons	Position literals first in String comparisons	484
LiteralsFirstInComparisons	Position literals first in String comparisons	498
AvoidLiteralsInIfCondition	Avoid using literals in if statements	509
LiteralsFirstInComparisons	Position literals first in String comparisons	515
LiteralsFirstInComparisons	Position literals first in String comparisons	526
AvoidLiteralsInIfCondition	Avoid using literals in if statements	536
LiteralsFirstInComparisons	Position literals first in String comparisons	542
LiteralsFirstInComparisons	Position literals first in String comparisons	553
LiteralsFirstInComparisons	Position literals first in String comparisons	569
LiteralsFirstInComparisons	Position literals first in String comparisons	579

Priority 4

Main.java

Rule	Violation	Line
OneDeclarationPerLine	Use one line for each declaration, it enhances code readability.	196
OneDeclarationPerLine	Use one line for each declaration, it enhances code readability.	197
OneDeclarationPerLine	Use one line for each declaration, it enhances code readability.	478
OneDeclarationPerLine	Use one line for each declaration, it enhances code readability.	480

Files

Main.java

Rule	Violation	Priority	Line
CommentRequired	Class comments are required	3	18
CommentRequired	Class comments are required	3	20
CommentRequired	Field comments are required	3	21
CommentRequired	Field comments are required	3	22
CommentRequired	Field comments are required	3	23
CommentRequired	Field comments are required	3	24
CommentRequired	Field comments are required	3	25
CommentRequired	Field comments are required	3	26
CommentRequired	Field comments are required	3	27

CommentRequired	Field comments are required	3	27
CommentRequired	Field comments are required	3	28
CommentRequired	Field comments are required	3	29
CommentRequired	Field comments are required	3	30
AvoidDuplicateLiterals	The String literal "name" appears 5 times in this file; the first occurrence is on line 41	3	41
CommentRequired	Field comments are required	3	54
CommentRequired	Field comments are required	3	55
CommentRequired	Public method and constructor comments are required	3	57
AvoidFileStream	Avoid instantiating FileInputStream, FileOutputStream, FileReader, or FileWriter	1	61
AvoidInstantiatingObjectsInLoops	Avoid instantiating new objects inside loops	3	81
AvoidInstantiatingObjectsInLoops	Avoid instantiating new objects inside loops	3	87
DoNotUseThreads	To be compliant to J2EE, a webapp should not use any thread.	3	91
CommentRequired	Public method and constructor comments are required	3	93
DoNotUseThreads	To be compliant to J2EE, a webapp should not use any thread.	3	107
CommentRequired	Public method and constructor comments are required	3	109
PreserveStackTrace	Thrown exception does not preserve the stack trace of exception 'e' on all code paths	3	131
UnusedPrivateMethod	Avoid unused private methods such as 'TestArray()'.	3	144
OneDeclarationPerLine	Use one line for each declaration, it enhances code readability.	4	196
UnusedAssignment	The initializer for variable 'world_Team' is never used (overwritten on lines 319 and 455)	3	196
OneDeclarationPerLine	Use one line for each declaration, it enhances code readability.	4	197
LiteralsFirstInComparisons	Position literals first in String comparisons	3	199
AvoidLiteralsInIfCondition	Avoid using literals in if statements	3	205
AvoidLiteralsInIfCondition	Avoid using literals in if statements	3	210
AvoidLiteralsInIfCondition	Avoid using literals in if statements	3	217
AvoidLiteralsInIfCondition	Avoid using literals in if statements	3	261
AvoidLiteralsInIfCondition	Avoid using literals in if statements	3	266
AvoidLiteralsInIfCondition	Avoid using literals in if statements	3	273
AvoidDuplicateLiterals	The String literal "\u6700\u4f73\u6536\u5165\u904e\u7a0b(\u5404\u968e\u6bb5\u6bd4\u8cfd\u7686\u6253\u6eff)\uff1a\u7e3d\u6536\u5165\u70ba" appears 4 times in this file; the first occurrence is on line 325	3	325
AvoidDuplicateLiterals	The String literal ", \u806f\u76df\u51a0\u8ecd\u8cfd\u5c0d\u4e0a" appears 4 times in this file; the first occurrence is on line 326	3	326
AvoidDuplicateLiterals	The String literal ", \u4e16\u754c\u8cfd\u5c0d\u4e0a" appears 4 times in this file; the first occurrence is on line 326	3	326
LiteralsFirstInComparisons	Position literals first in String comparisons	3	336
AvoidLiteralsInIfCondition	Avoid using literals in if statements	3	342
AvoidLiteralsInIfCondition	Avoid using literals in if statements	3	347
AvoidLiteralsInIfCondition	Avoid using literals in if statements	3	353
AvoidLiteralsInIfCondition	Avoid using literals in if statements	3	397
AvoidLiteralsInIfCondition	Avoid using literals in if statements	3	402
AvoidLiteralsInIfCondition	Avoid using literals in if statements	3	409
OneDeclarationPerLine	Use one line for each declaration, it enhances code readability.	4	478
UnusedLocalVariable	Avoid unused local variables such as 'num'.	3	479
OneDeclarationPerLine	Use one line for each declaration, it enhances code readability.	4	480
LiteralsFirstInComparisons	Position literals first in String comparisons	3	484
LiteralsFirstInComparisons	Position literals first in String comparisons	3	498
AvoidLiteralsInIfCondition	Avoid using literals in if statements	3	509
LiteralsFirstInComparisons	Position literals first in String comparisons	3	515
LiteralsFirstInComparisons	Position literals first in String comparisons	3	526
AvoidLiteralsInIfCondition	Avoid using literals in if statements	3	536
LiteralsFirstInComparisons	Position literals first in String comparisons	3	542
LiteralsFirstInComparisons	Position literals first in String comparisons	3	553
LiteralsFirstInComparisons	Position literals first in String comparisons	3	569
LiteralsFirstInComparisons	Position literals first in String comparisons	3	579