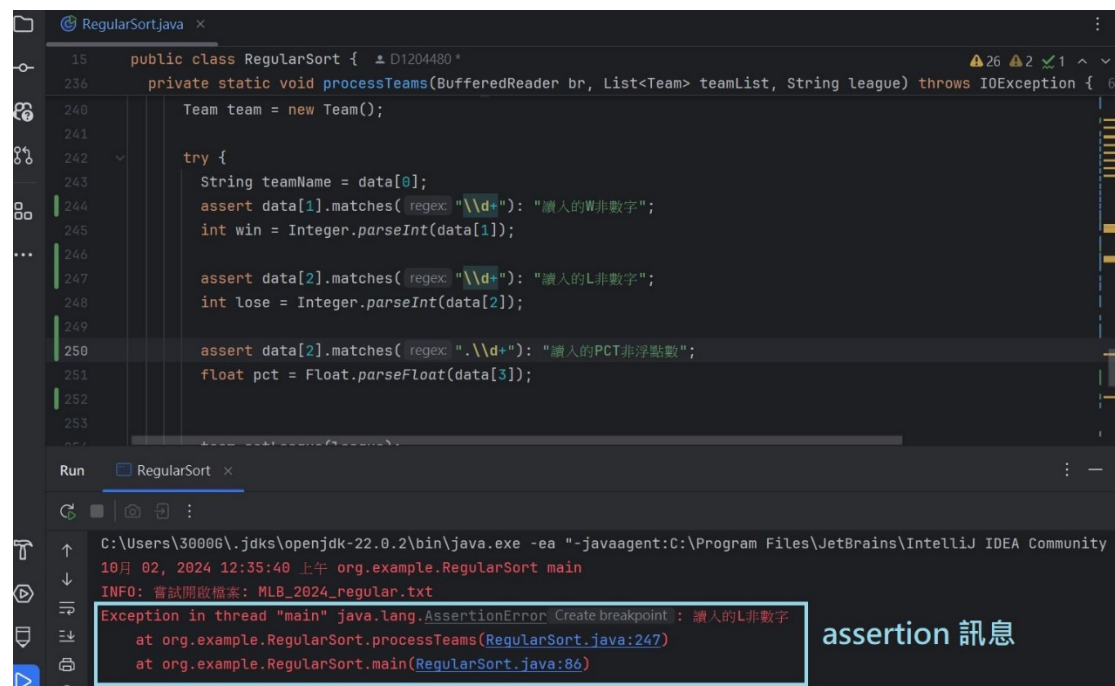


# 防禦性程式設計的應用與心得

## 1. assertion (斷言)

確保程式在執行某些條件是正確的。如果條件不滿足，程式會拋出一個 `AssertionError`。

- 使用時機: 程式開發階段。
- 如何使用: `assertion` 發生時，系統會立刻中斷，並執行使用者自訂的斷言內容。
- 優缺點:
  - 優: 開發階段可以大量放置斷言，開發完成後 IDE 可以關閉斷言，不影響程式執行。
  - 缺: 僅作為找出錯誤的位置，不能作為商業邏輯區塊程式的錯誤判斷，例如: 判斷使用者輸入正確或異常。



```
15 public class RegularSort {
236 private static void processTeams(BufferedReader br, List<Team> teamList, String league) throws IOException {
    Team team = new Team();
    try {
        String teamName = data[0];
        assert data[1].matches(regex: "\\d+"): "讀入的W非數字";
        int win = Integer.parseInt(data[1]);
        assert data[2].matches(regex: "\\d+"): "讀入的L非數字";
        int lose = Integer.parseInt(data[2]);
        assert data[3].matches(regex: "\\d+\\.\\d+"): "讀入的PCT非浮點數";
        float pct = Float.parseFloat(data[3]);
    }
}
```

Run RegularSort

C:\Users\30006\.jdk\openjdk-22.0.2\bin\java.exe -ea "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community\lib\idea\_rt.jar=5000:C:\Program Files\JetBrains\IntelliJ IDEA Community\bin" -Dfile.encoding=UTF-8

10月 02, 2024 12:35:40 上午 org.example.RegularSort main

INFO: 嘗試開啟檔案: MLB\_2024\_regular.txt

Exception in thread "main" java.lang.AssertionError: 讀入的L非數字  
at org.example.RegularSort.processTeams(RegularSort.java:247)  
at org.example.RegularSort.main(RegularSort.java:86)

assertion 訊息

## 2. exception (例外處理)

處理程式執行時的錯誤或異常的主要方法，透過捕捉錯誤或異常，避免程式崩潰，並回傳錯誤訊息。

- 使用時機: 用於處理可預期或非預期的錯誤，例如讀寫檔案、使用者輸入。
- 如何使用: **exception** 發生時，可以使用拋出異常 (**throw exception**)，或是使用 **try / catch** 處理異常。
- 優缺點:
  - 優: 避免程式崩潰。
  - 缺: 濫用異常可能導致程式碼變得複雜，且異常處理的負載較大，應謹慎使用。

```
try {
    // 設定 FileHandler，將日誌寫入檔案 "app.log"
    FileHandler fileHandler = new FileHandler(pattern: "app.log", append: true); // true 表示追加到文件中
    fileHandler.setFormatter(new SimpleFormatter()); // 設定格式為簡單格式
    logger.addHandler(fileHandler);
    logger.info("這是一條 INFO 等級的日誌訊息");
} catch (FileNotFoundException e) {
    logger.log(Level.SEVERE, msg: "發生找不到檔案: " + e.getMessage());
    throw new RuntimeException(e);
} catch (IOException e) {
    logger.log(Level.SEVERE, msg: "發生開檔IO錯誤: " + e.getMessage());
    throw new RuntimeException(e.getMessage());
} // end of BufferedReader

} catch (IOException e) {
    logger.log(Level.SEVERE, msg: "發生讀寫Log錯誤: " + e.getMessage());
    e.printStackTrace();
}
```

### 3. log 日誌

紀錄程式運行的狀況，包括一般訊息(Info)、警告(Warn)、錯誤(Error)或重要(SEVERE)事件等。

- 使用時機: 使用者可以根據需要記錄不同詳細程度的資料，以便用於調整、監控、後期分析。
- 如何使用: 日誌會把訊息寫入文件或其他儲存設備，每個類別有自己的日誌檔案，可以依照不同類別分別查詢日誌內容。
- 優缺點:

- 優：便於追蹤問題來源和系統狀態。
- 缺：記錄過多，可能產生無用的資訊。

```
C:\Users\30006\.jdk\openjdk-22.0.2\bin\java.exe -ea "-javaagent:C:\Program Files\JetBrains\IntelliJ
10月 01, 2024 11:50:46 下午 org.example.RegularSort main
INFO: 嘗試開啟檔案: MLB_2024_regular.txt
10月 01, 2024 11:50:46 下午 org.example.RegularSort processTeams
SEVERE: 發生處理數字格式錯誤: For input string: ".574"
java.lang.NumberFormatException: Create breakpoint : For input string: ".574"
    at java.base/java.lang.NumberFormatException.forInputString(NumberFormatException.java:67)
    at java.base/java.lang.Integer.parseInt(Integer.java:588)
    at java.base/java.lang.Integer.parseInt(Integer.java:685)
    at org.example.RegularSort.processTeams(RegularSort.java:249)
    at org.example.RegularSort.main(RegularSort.java:86)
Exception in thread "main" java.lang.AssertionError: Create breakpoint : 每個區應有5個隊伍
    at org.example.RegularSort.processTeams(RegularSort.java:265)
    at org.example.RegularSort.main(RegularSort.java:86)
```

Log 訊息

總結：

1. Assertion: 用於開發階段，檢查假設條件是否成立，不適用於處理商業邏輯的判斷。
2. Exception: 用來處理程式執行時的錯誤，避免程式崩潰。
3. Log: 用於記錄程式的執行狀態、異常和錯誤，便於事後分析。

```
private static void processTeams(BufferedReader br, List<Team> teamList, String league) throws IOException {
    for (int i = 0; i < 5; i++) {
        String line = br.readLine(); // 跳到標頭(例如:NL WEST)的下一行
        String[] data = line.split(" ");
        Team team = new Team();

        try { try/catch (exception 例外處理)
            String teamName = data[0];
            int win = Integer.parseInt(data[1]);
            int lose = Integer.parseInt(data[2]);
            float pct = Float.parseFloat(data[3]);

            team.setLeague(league);
            team.setTeamName(teamName);
            team.setWin(win);
            team.setLose(lose);
            team.setPct(pct);

            teamList.add(team); // 放入陣列

        } catch (NumberFormatException e) {
            logger.log(Level.SEVERE, "msg: 發生處理數字格式錯誤: " + e.getMessage());
            e.printStackTrace();
        }

    } // end of for()
    assert teamList.size() == 5: "每個區應有5個隊伍";
} // end of processTeams
```

log 日誌

assertion 斷言