

作業一：擲骰子遊戲。

遊戲規則：

2 個玩家 Tom 跟 Mary，兩個玩家手上都有 3 個骰子，其中兩個骰子有 6 面，另一個骰子有 8 面。

每一個回合，兩個玩家都會擲出手中的 2 個 6 面骰。用骰子的總和比大小。

如果雙方平手，則擲出另一個 8 面骰再比一次結果。如果還是平手就紀錄為「draw」。

讓兩個玩家玩十次這種比賽後，統計勝負結果。

印出類似如下的結果：(註：執行結果可能不會每次都相同)

Tom wins:4

Mary wins:5

Draw:1

請回答下面的問題：

Q1. 這個程式裡你會設計幾個「東西」？他們「知道」些什麼？他們又可以「做」些什麼？請參考簡報第四章第三頁的圖，畫出每個東西的結構。

Q2. 針對每個「東西」可以「做」的事情，那些是你認為需要撰寫對應的 Test code？請列出。

Q3. 實做出你的設計，包含正式的程式碼與 Test code (注意：Test code 原則上不應該全部擠在同一個檔案內)

Hint 1: 在開發的過程中，你應該要按照 Q1~Q3 的順序進行回答(也就是說，不要在完成 Q3 之後才開始回頭做 Q1, Q2)

Hint 2: 如果你按照教材中的方式(簡報第 5 章第 32 頁)去生成骰子的點數，請觀察一下你的骰子擲出的點數機率是否接近？(你也許需要查一下 Java 中有關 Math 類別提供的方法來改善它)

Hint 3: 請在設計你的程式時，請考慮以下的問題 (不是要完成這些問題)：

- 如果要改成跑 20 次計算結果，你需要改動那些檔案？
- 如果要改成玩家丟出 3 個六面骰，你需要改動那些檔案？
- 如果要改成骰子不是一個公平的骰子 (也就是說，每一面出現的機率不是相同的)，你需要改動那些檔案？

上述的改動是否符合你當初設計「東西」的初衷？也就是說，這項改動是否是這個「東西」可能會有行為？

繳交方式：

- 請把 Q1, Q2 的回答填在一個檔案中，該檔案請註明組員的學號姓名。未註明學號姓名者不予計分。
- 請上傳遊戲能正常運作的程式與 test code(至少須有 .java 檔)及項目 1。
- 程式碼中的每一個方法(method)都需要有說明該功能的註解(寫法請參考 javadoc 的規範)
- 一組只需上傳一份。

評分標準：

程式正確執行(40%) 符合物件導向設計精神(40%) 符合 Java 命名慣例(10%) 完整的說明文件(10%)