

Victoria University of Wellington
School of Engineering and Computer Science

SWEN224: Formal Foundations of Programming

Assignment 1 (worth 4% of overall mark)

Due: Thursday 6th August @ Mid-night

In this assignment you will take a small program for checking *chess games* and: 1) use FindBugs to identify and eliminate a number of problems; 2) use Eclipse's built-in `@NonNull` analysis to demonstrate that the program is free from a particular kind of defect — namely, it cannot throw any `NullPointerException`s. Finally, you will write a short report discussing some of the issues you encountered.

The program you are analysing is available for download from the SWEN224 homepage, as `chess.jar`. You should begin by downloading this and setting it up in Eclipse. The program comes with a suite of tests which, currently, are failing.

1 FindBugs (20%)

The first part of the assignment is to use FindBugs to identify and eliminate any problematic pieces of code in the chess program. In some cases, you will need to think about what the right change to the program is. *Having eliminated all issues identified by FindBugs, you should find that all tests now pass.* If they don't, this indicates you have introduced an error when fixing one of the problems FindBugs found. If this happens, you should retrace your steps and fix the problem.

2 NonNull Analysis (40%)

The second part of the assignment is to use Eclipse's built-in `@NonNull` analysis. You will need to enable this analysis within Eclipse before you can proceed with this part, and you should find instructions for doing this in the corresponding lab. The goal for this analysis is as follows:

1. **To annotated every non-primitive field, parameter and return with `@NonNull` or `Nullable`.** You will need to examine the program to decide which is the right option.
2. **To make updates to the program as necessary to help guide the analysis.** Any updates you make should not affect the program's functionality, and all tests should continue to pass. The updates needed are very small in nature.

At the end of the analysis, you should find that Eclipse does not report any “Null Type Safety” errors or similar. Furthermore, the program should work exactly as before, and all tests should pass.

3 Report (40%)

To complete this assignment, you should right a short report which answers the following questions:

1. **(5 marks)** Identify those issues reported by FindBugs which you believe corresponded to real bugs in the program. For each, briefly discuss why you thought it was a problem.
2. **(5 marks)** The `@Nullable` annotation is required in several places. Briefly, discuss one situation where you needed to use this annotation and why.
3. **(10 marks)** Several small changes are required before Eclipse will report this program as being free from “Null Type Safety” errors. For each change you made, briefly discuss why you made this change.

4 Submission

Your program code should be submitted electronically via the *online submission system*, linked from the course homepage. The report should be submitted as either PDF or a PNG files; other formats will not be accepted. The required source files are:

```
assignment1/chessview/moves/SinglePieceTake.java
assignment1/chessview/moves/MultiPieceMove.java
assignment1/chessview/moves/EnPassant.java
assignment1/chessview/moves/Move.java
assignment1/chessview/moves/Check.java
assignment1/chessview/moves/PawnPromotion.java
assignment1/chessview/moves/NonCheck.java
assignment1/chessview/moves/SinglePieceMove.java
assignment1/chessview/moves/Castling.java
assignment1/chessview/Round.java
assignment1/chessview/Position.java
assignment1/chessview/Board.java
assignment1/chessview/ChessGame.java
assignment1/chessview/pieces/King.java
assignment1/chessview/pieces/Piece.java
assignment1/chessview/pieces/Pawn.java
assignment1/chessview/pieces/Knight.java
assignment1/chessview/pieces/Queen.java
assignment1/chessview/pieces/Rook.java
assignment1/chessview/pieces/PieceImpl.java
assignment1/chessview/pieces/Bishop.java
```

You must ensure your submitted code meets the following requirements:

1. **Your code is packaged into a jar file, including the source code.** See the export-to-jar tutorial linked from the course homepage for more on how to do this. *Note, the jar file does not need to be executable.*
2. **You have removed any debugging code that produces output, or otherwise affects the computation.** *This ensures the output from the program does not include spurious information.*

Note: Failure to meet these requirements could result in you getting zero marks for the assignment.