

# Test Plan

## Introduction

- This is the test plan for the java application chess game.
- The application has a terminal interface, and all chess pieces are printed via Unicode.
- The application has a main class Chess which controls players' movement, game, detects game ending conditions and sets the board at the beginning of the game; a Board class which mainly consists of a 2D array of Piece and a lot methods to set, move, get pieces at certain locations on the board; an abstract Piece class which defines common attributes and methods for all chess pieces; six subclass of Piece that are Bishop, King, Knight, Pawn, Queen and Rook; two special subclass of Piece that are Tank and Soldier. The two are custom chess piece with different moving methods from the original six pieces.

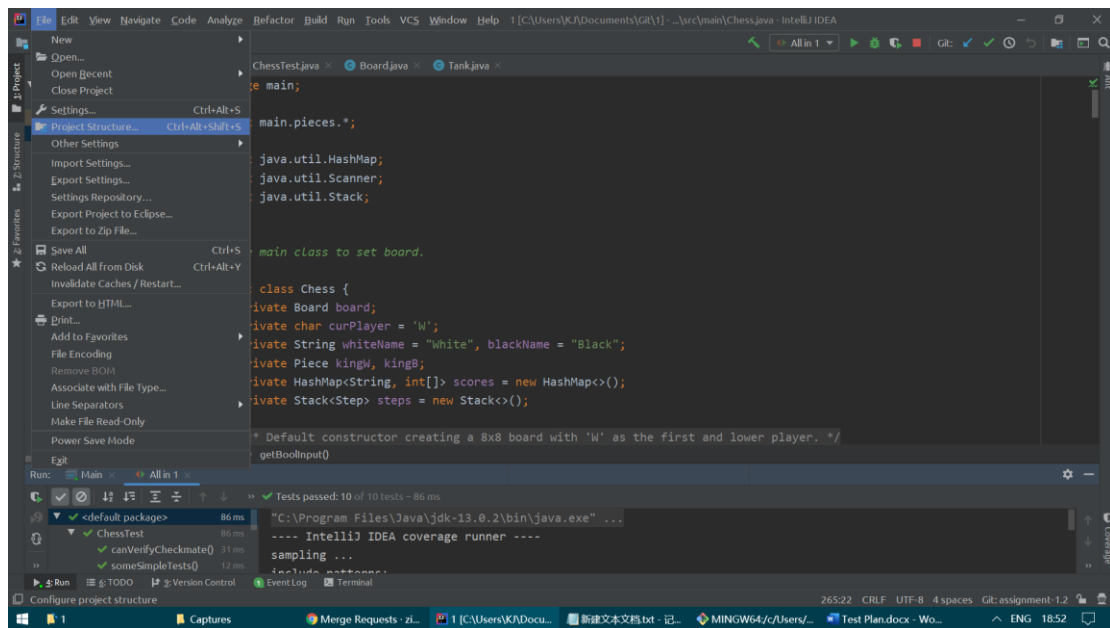
## Prerequisites

The application is built with java version 13.0.2.

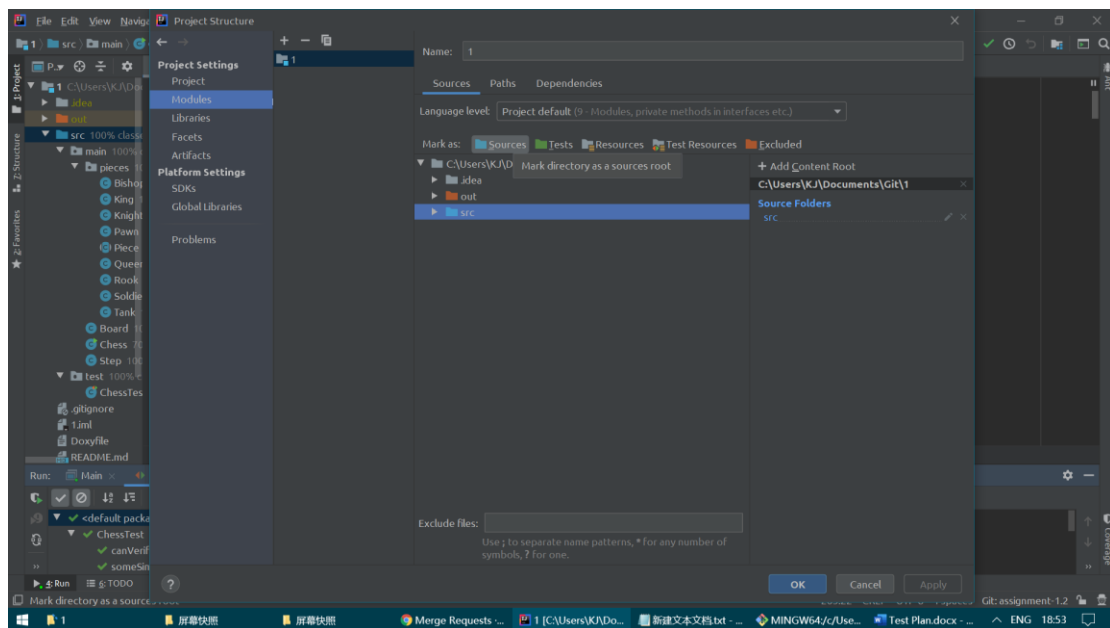
You need to import Junit5.4 for the test class to work.

To set the prerequisites, do as the following.

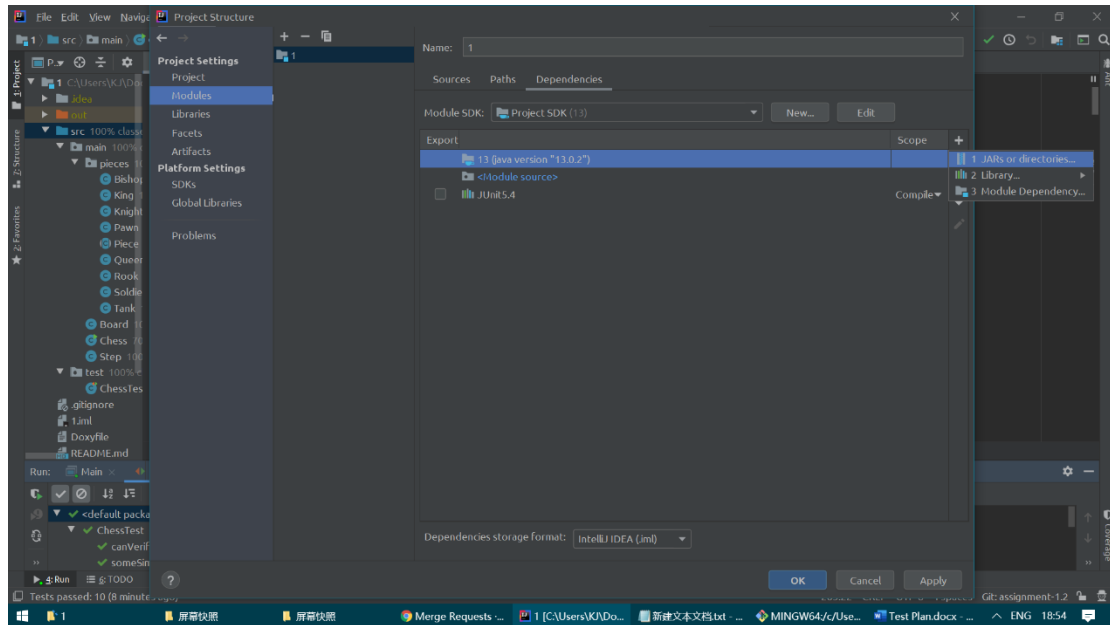
Open project structure.



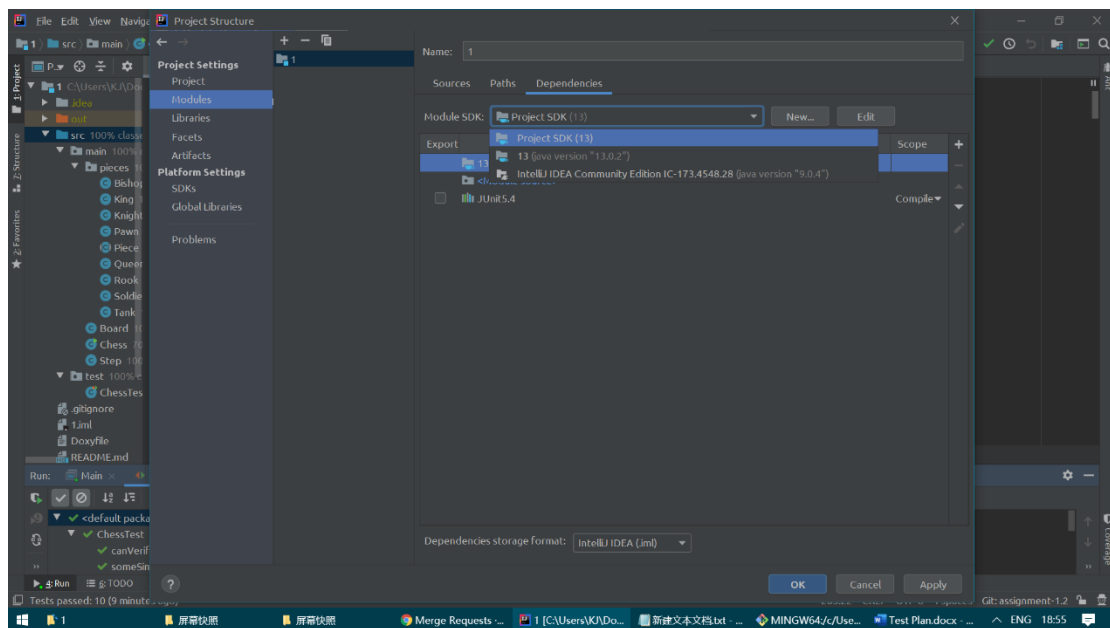
Set src as source file.



Add junit as dependency.



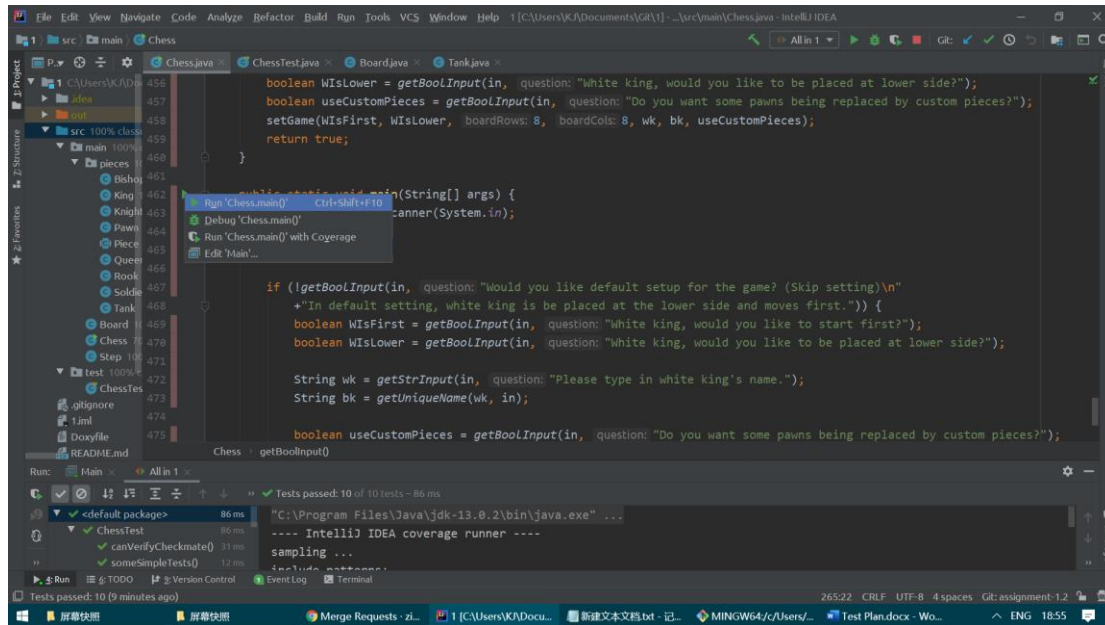
And don't forget to set project SDK with JAVA 13.



The IDE currently using is IntelliJ IDEA community 2019.3.

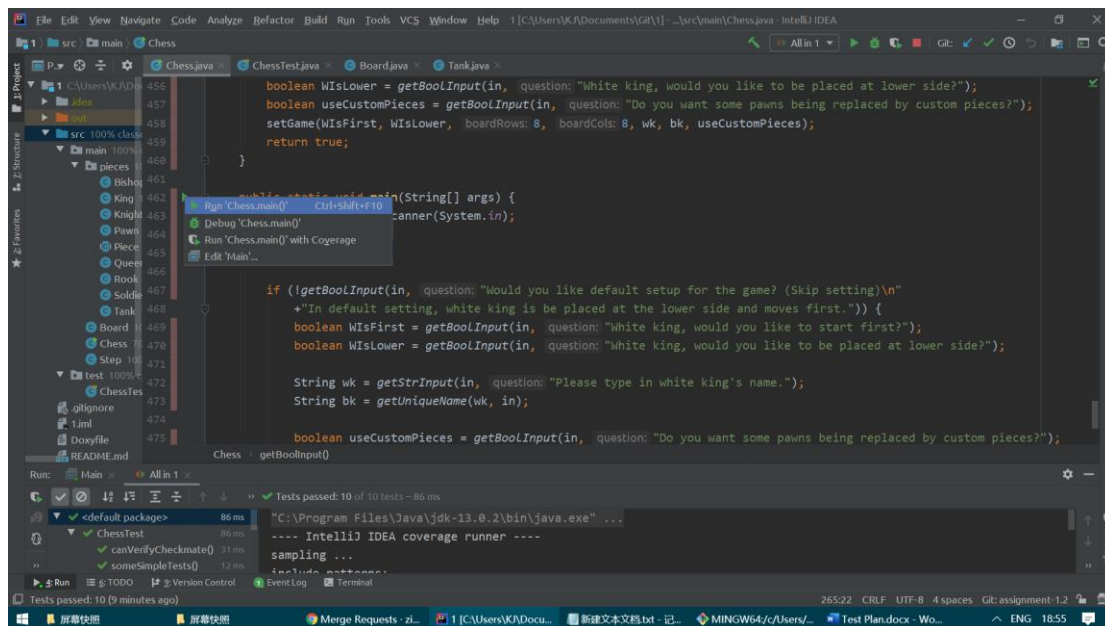
## Environment Setup and configurations

Import the project in an IDE, then run main() method in Chess class.

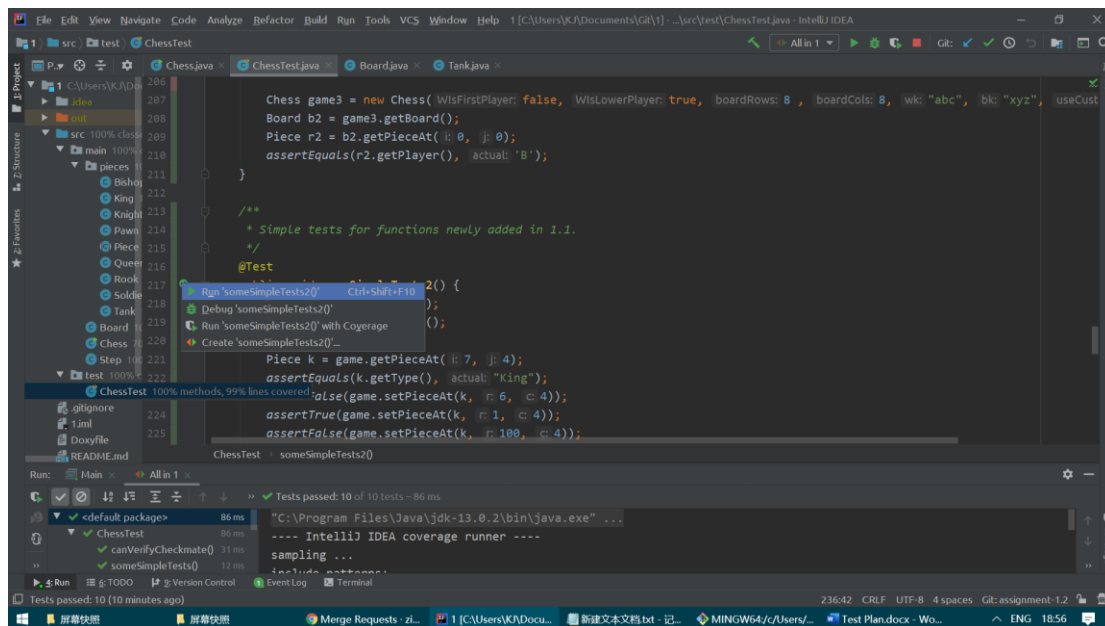


## Operations and the results

You can run test cases by right click on "src" and "Run 'All tests'".



If you would like to run test case one each time, do as the following.



After running main() method, open the console window. You should see a question asking whether to use default setting.

Now you should be able to type in the console window.

If you type y, then a default 8x8 board should be printed out immediately like this:



With the left most column as row number, and the second lowest row as column number.

The lowest row "W>" is asking for white king to type next movement command.

If you type n, then you will be asked to specify the location (upper or lower) of the

white king and who's the first to move. The game board will be set accordingly. So if black king moves first and black king is at lower part, the board should be like this:



By the way, at any time, any player should be able to exit the application by typing "exit" in the command prompt.

To move a chess piece, simply type in four integers in the same line, separated by single space, then hit enter. The first two integers should be the piece being picked up; the last two integers should be where the piece is going to be placed at. There are three rules to be tested:

- 1) A player cannot move another player's piece.
- 2) Both the picking location and the placing location should be a valid location on the board.
- 3) Every piece should follow its moving rule, such as only knights can leap over other pieces, and pawn can and only can move two steps in its first movement.
- 4) A player cannot move its piece to a location occupied by his / her own piece.

After each command, the updated board will be printed, followed with command prompt for another player.

If there is a checkmate in the game, the game should finish with a new line "<Winner> is Winner! Game ended." where <Winner> will be substituted by the actual winner (Either Black King or White King) in the real game. After that, you should see "Process finished with exit code 0".

The user shouldn't be able to interact with the interface at this time, but he or she may run the main() method again to start a new game.

## Error messages if any

- If the command doesn't follow the requirements, an error message of "Please give a valid input! (four integers in the same line, separated by single space)!" should be printed, followed by a new command prompt for the same player.
- If a player tries to move another player's piece, an error message of "<Current Player>, you should not move <Another Player>'s piece!" will be displayed, where the <Current Player> and <Another Player> will be substituted by "Black King" or "White King" in the real game. Similarly, a new command prompt for the same player will be printed.
- If any of the picking up location or the placing location is invalid (not on the board), an error message of "Please give locations in valid range (it should be somewhere on the board)!" will be displayed, followed by a new command prompt for the same player.

- If the given route doesn't fit in the piece's movement method, an error message of "Your piece cannot move in such way!" will be displayed, followed by a new command prompt for the same player.
- If a player tries to move a piece to another place occupied by his / her own piece, an error message of "You cannot capture another piece owned by you!" will be displayed, followed by a new command prompt for the same player.