**Refactoring after Build 2:**

**Potential refactoring targets were identified on basis of some of the following factors**.

* To ease the further adding of new functionality.
* To clean up the code to reduce further possibility of introducing of bugs.
* Making the code readable and easy to understand.
* Simplifying the overall code structure and project architecture.

**List of potential refactoring targets.**

1. Order was being created in Order Creation phase, Changed package structure where Strategy pattern is used and based on the strategy of the player, issueOrder method of Player class invokes create order method of respective Strategy class.
2. From loading one Map, Adapter pattern is being implemented. Since two type of Map patterns are being used. To use either the original ‘domination’ file reader or ‘conquest’ file reader, map handler adapter has been introduced
3. The Order command such as deploy for human behaviour was being created in the OrderCreationPhase class but is now being refactored with strategy pattern handling deploy order command in Human Strategy class
4. Every time number of countries owned by a player becomes 0, the player was being removed from the list and the game would move on. This was initially implemented in executeAllOrders method. But now the code is refactored and this done after the all the orders of a turn have been executed before checking for the end of the game.
5. Initially, in the GameEngine only edit option was available since only with user interaction commands we could play the game but now single and tournament mode both exist
6. Taking of Orders while maintaining the loop was done in create order method of order creation phase but is now implemented in Player class’s issue order method
7. Renaming method names in *ValidateMap*, *GameEngine* and other classes to a better name that can easily identify its purpose.
8. Now a linked list of Orders is maintained to execute all the orders
9. Initially only Domination map was being loaded, now the user is given an option from which to chose- Domination or conquest
10. Moving takeOrder method from OrderCreation to Human class
11. Moving all the processordercommands method from OrderCreation to Human and other Strategy classes
12. The start function of the GameEngine class is also used to start the tournament and load the saved game
13. Moving takeOrder tests from OrderCreationTest to HumanTest class
14. Map load and save test was being done in MapEditor test and now its done in DominationMapHandlerTest
15. Order creation was tested in OrderCreationPhaseTest now moved to PlayerTest class

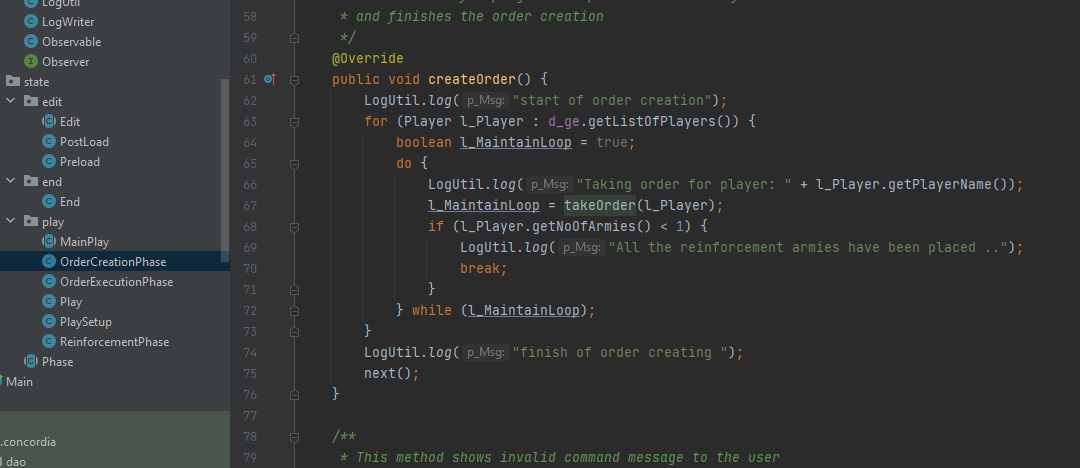
**List of refactoring implemented:**

1. Order was being created in Order Creation phase, Changed package structure where Strategy pattern is used and based on the strategy of the player, issueOrder method of Player class invokes create order method of respective Strategy class.
2. From loading one Map, Adapter pattern is being implemented. Since two type of Map patterns are being used. To use either the original ‘domination’ file reader or ‘conquest’ file reader, map handler adapter has been introduced
3. The Order command such as deploy for human behaviour was being created in the OrderCreationPhase class but is now being refactored with strategy pattern handling deploy order command in Human Strategy class
4. Every time number of countries owned by a player becomes 0, the player was being removed from the list and the game would move on. This was initially implemented in executeAllOrders method. But now the code is refactored and this done after the all the orders of a turn have been executed before checking for the end of the game.
5. Initially, in the GameEngine only edit option was available since only with user interaction commands we could play the game but now single and tournament mode both exist

**Refactoring Implemented along with all the Test classes.**

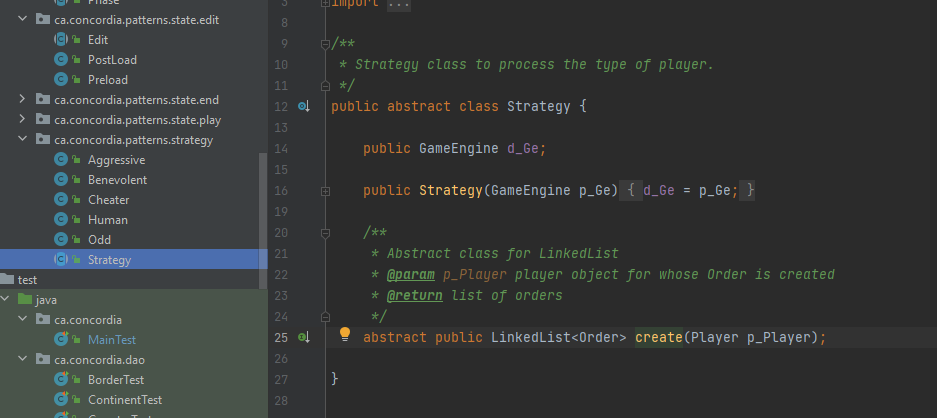
1. **In previous build**

Order was being created in Order Creation phase

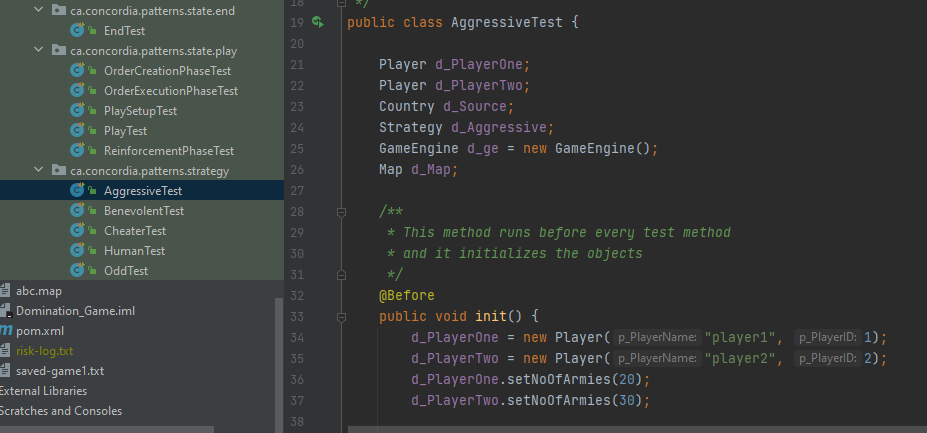


**Code after Refactoring:**

Changed package structure where Strategy pattern is used and based on the strategy of the player, issueOrder method of Plyer class invokes create order method of respective Strategy class

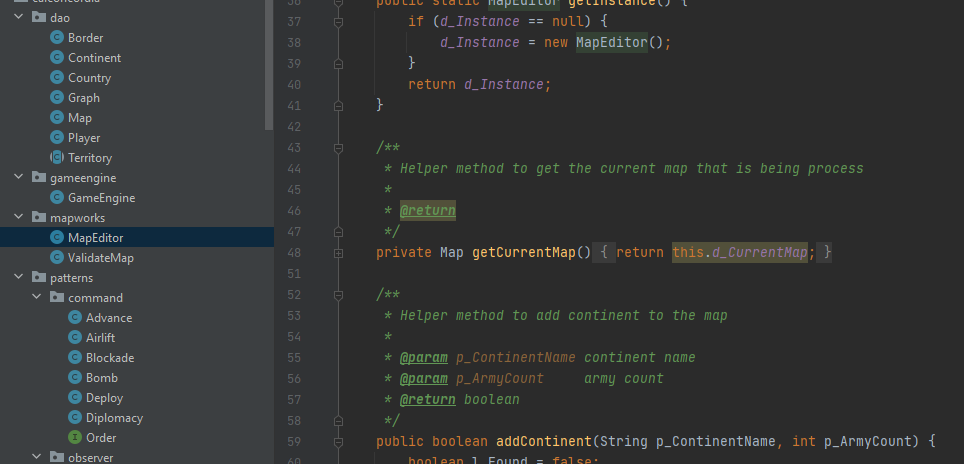


**Test class for Strategy and respective classes:-**



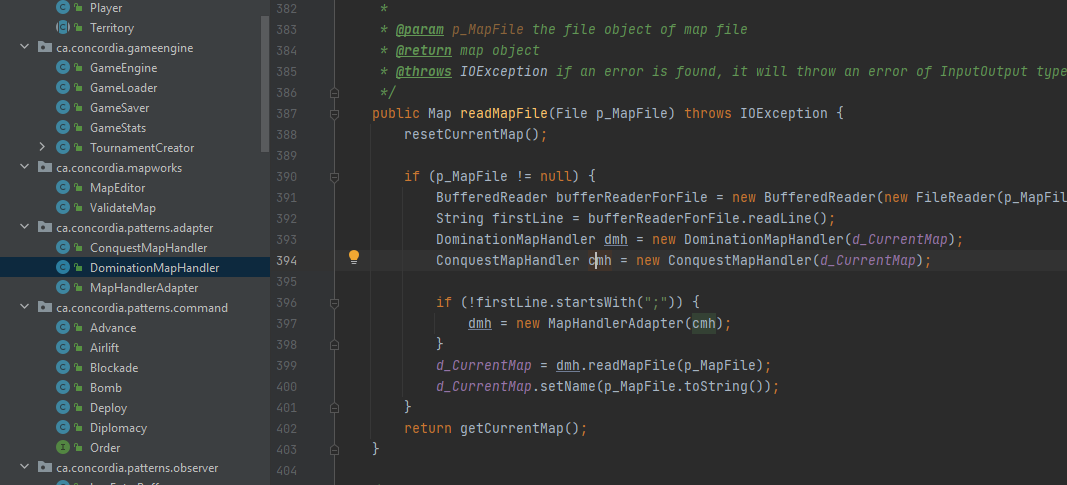
1. **In previous Build:**

Since we used only on one Map, we loaded it and used it directly

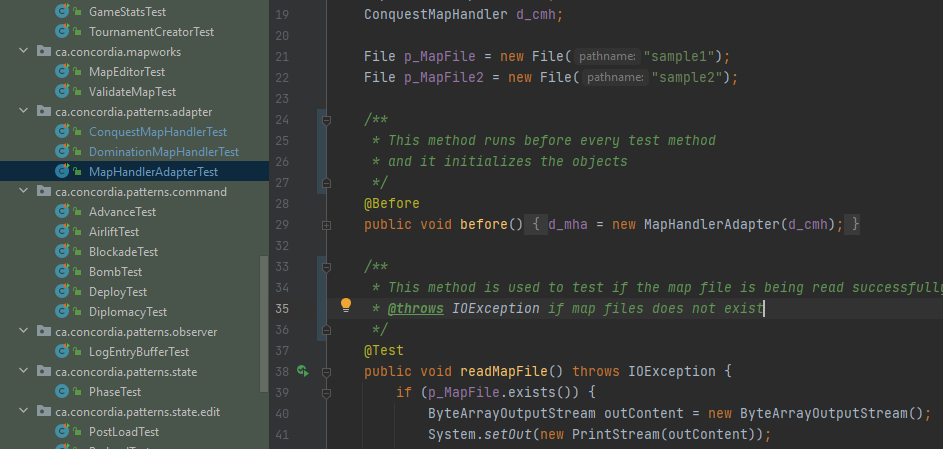


**Code after Refactoring:**

Adapter pattern: Since two type of Map patterns are being used. . To use either the original ‘domination’ file reader or ‘conquest’ file reader, map handler adapter has been introduced.

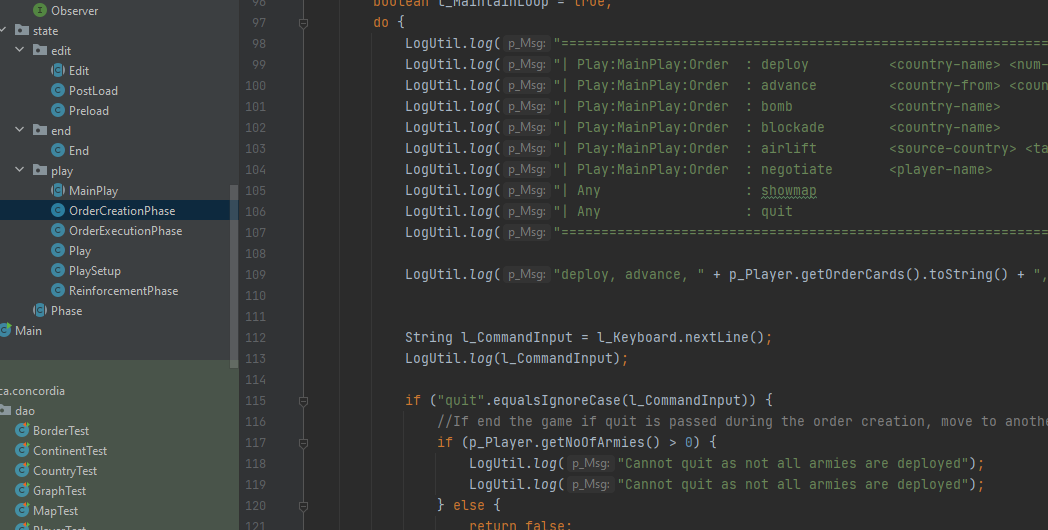


**Test classes for Adapter Pattern**



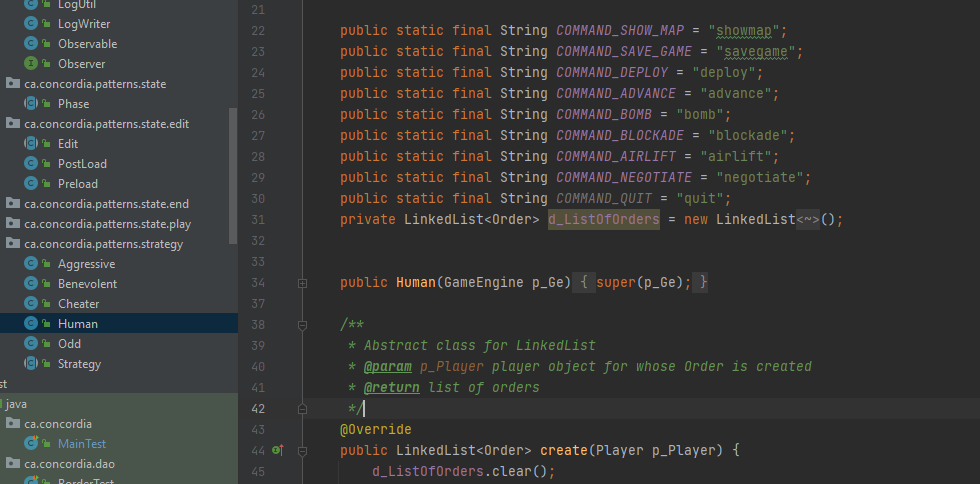
1. **In previous Build:**

The Order command such as deploy for human behaviour was being created in the OrderCreationPhase class but is now being refactored with strategy pattern handling deploy order command in Human Strategy class

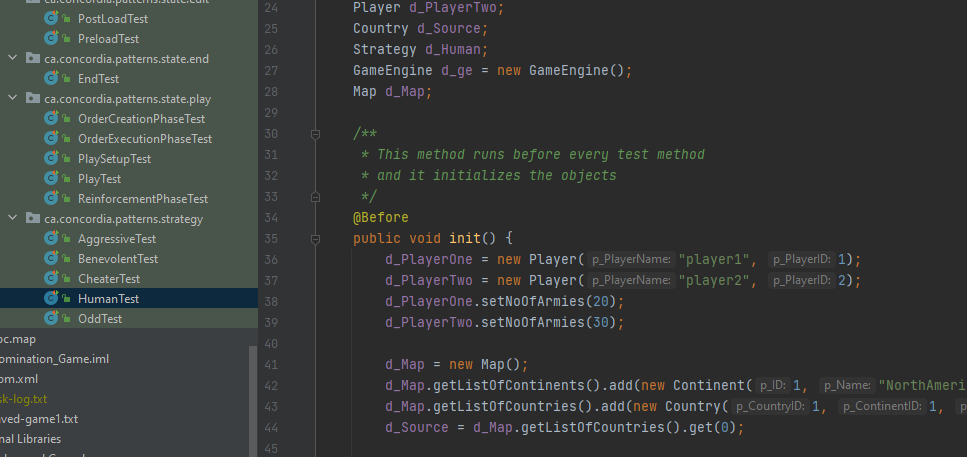


**Code after Refactoring:**

Human class is handling the command orders(input via user interaction) for creating the orders

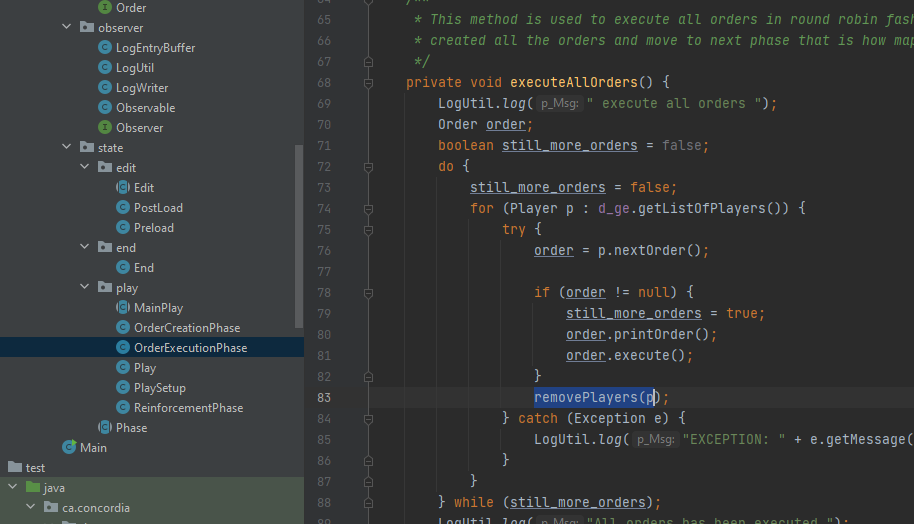


**Test classes for Human strategy pattern**



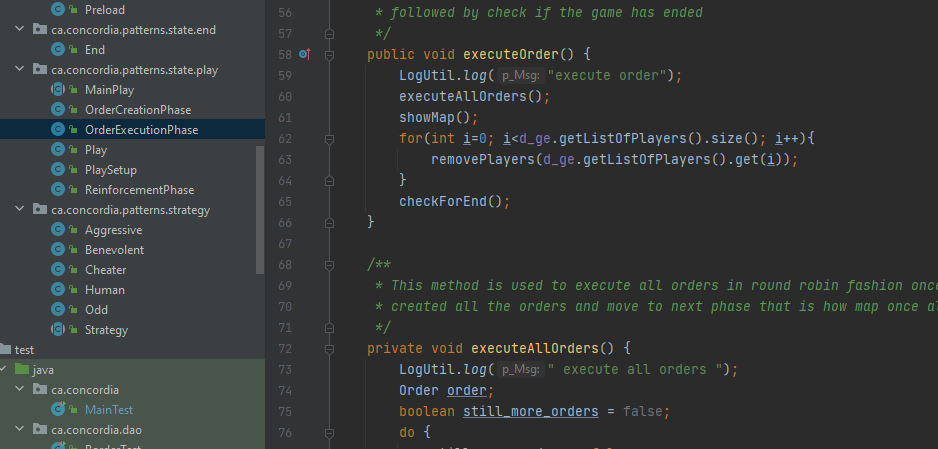
1. **Previous code:**

Every time number of countries owned by a player becomes 0, the player was being removed from the list and the game would move on. This was initially implemented in executeAllOrders method.

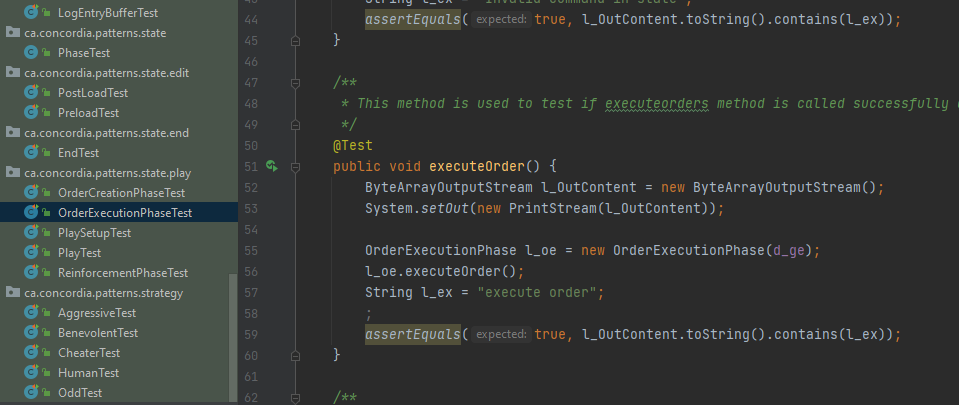


**Code after Refactoring:**

But now the code is refactored and this done after the all the orders of a turn have been executed before checking for the end of the game.

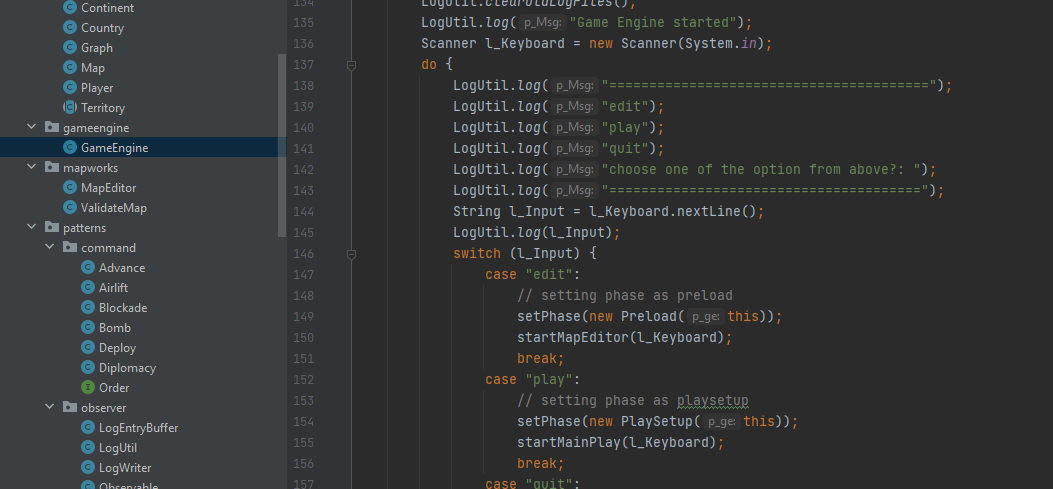


**Test class for OrderExecutionPhase:**



5. **Previous code:**

Initially, in the GameEngine only edit option was available since only with user interaction commands we could play the game but now single and tournament mode both exist



**Code after Refactoring:**

There’s play button for single mode and tournament option for tournament phase

