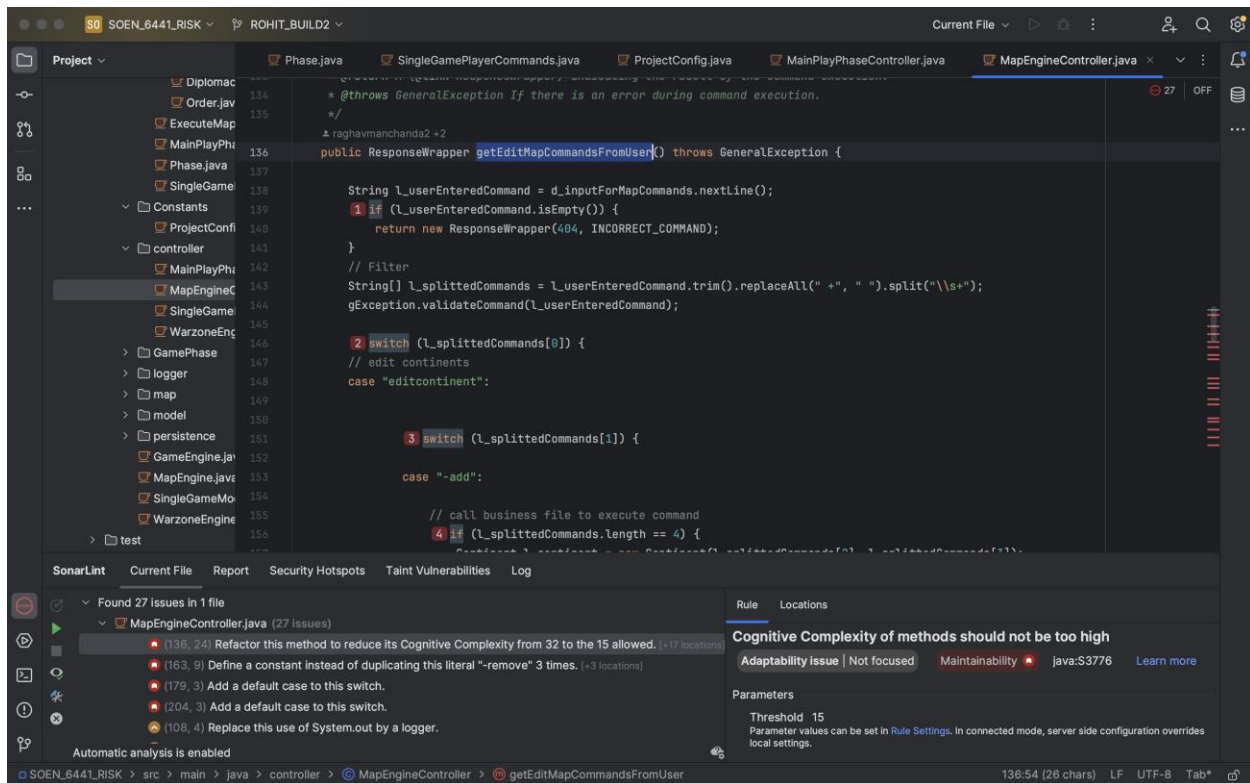


Refractor Documentation

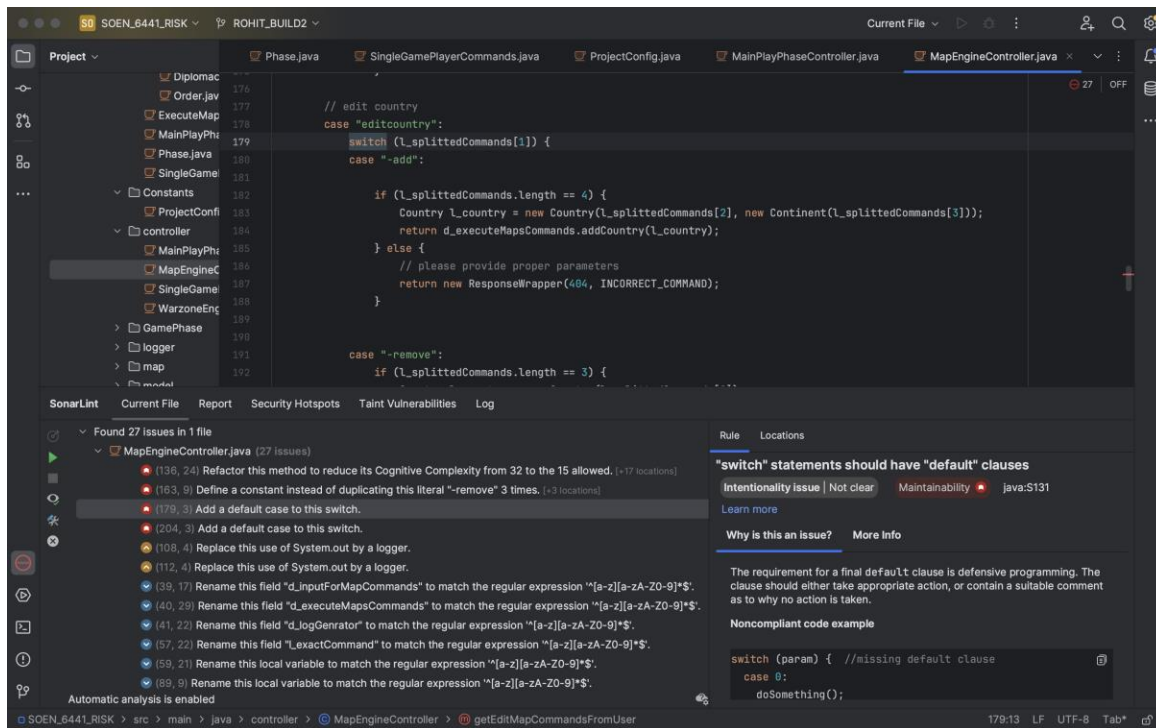
In MapEngineController.java

Problem: Remove cognitive complexity (Line 136)



Solution: To reduce the cognitive complexity of the `getEditMapCommandsFromUser` method in the `MapEngineController` class, we can break it down into smaller methods and use a switch-case structure.

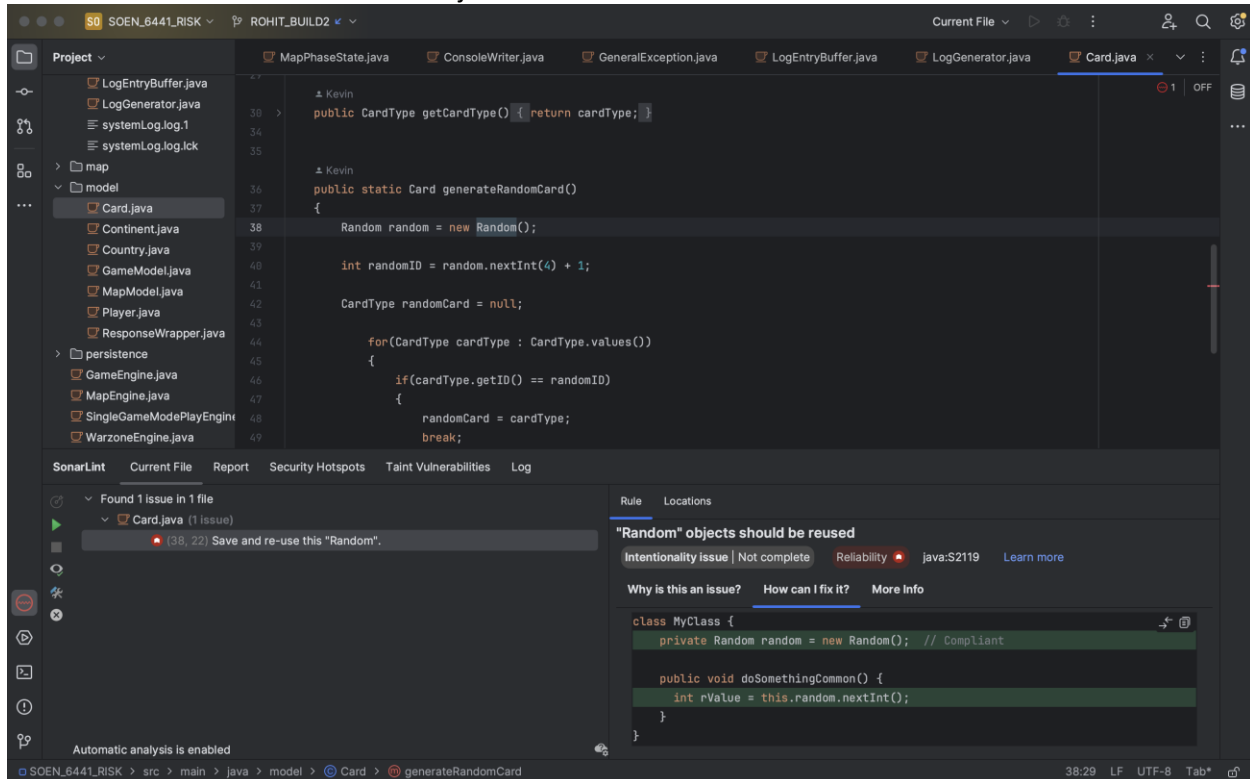
Problem: Add default case to the switch (Line 179,204)



In Card.java

Problem: Save and reuse the random (Line 38)

Solution: we can create a **Random** object as a static field in the **Card** class.



In ExecuteMapsCommands.java

Problem: Define constant instead of using the literal 3 times (Line 202,208)

The screenshot shows an IDE with the file `ExecuteMapsCommands.java` open. The code is as follows:

```
@Override
public ResponseWrapper editContinent(Continent p_continent, String command) throws GeneralException {
    if(command.equals("-add"))
    {
        return this.addContinent(p_continent);
    }
    else if (command.equals("1 -remove"))
    {
        return this.removeContinent(p_continent);
    }
    else
    {
        return new ResponseWrapper(404, "Incorrect Command");
    }
}
```

SonarLint has identified an issue: "String literals should not be duplicated". The issue is located at line 202, column 44. The rule is `java:S1192`. The issue description states: "Duplicated string literals make the process of refactoring error-prone, since you must be sure to update all occurrences. On the other hand, constants can be referenced from". The parameters for the rule are: threshold 3. The issue is marked as a "Maintainability" issue.

The screenshot shows the same IDE with the file `ExecuteMapsCommands.java` open. The code is now updated to use a constant for the string literal:

```
@Override
public ResponseWrapper editContinent(Continent p_continent, String command) throws GeneralException {
    if(command.equals("-add"))
    {
        return this.addContinent(p_continent);
    }
    else if (command.equals("1 -remove"))
    {
        return this.removeContinent(p_continent);
    }
    else
    {
        return new ResponseWrapper(404, 1 "Incorrect Command");
    }
}
```

The SonarLint issue is still present, but the code snippet now shows the constant `1 "Incorrect Command"` being used in the return statement. The issue description and parameters remain the same.

In MainPlayPhaseBusinessCommands.java

Problem: Replace string concatenation with Text Block (Line 106,168)

The screenshot shows the IntelliJ IDEA IDE with the file `MainPlayPhaseBusinessCommands.java` open. The editor displays a method `showMap()` with a multiline string concatenation on line 106. The SonarLint panel at the bottom shows 23 issues in 1 file. The first two issues are highlighted: (106, 34) and (168, 35), both with the message "Replace this String concatenation with Text block." The rule details on the right explain that multiline concatenation should be replaced with Text Blocks for better maintainability.

```
break;
}

Order bomb = new BombOrder(p_currentPlayer, targetCountry);
if(bomb.valid()) {
    p_currentPlayer.addOrder(bomb);
    return new ResponseWrapper(200, " Bomb order added in queue");
}else {
    return new ResponseWrapper(204, "One of the following occurred: \n"
        + "1. You do not possess a bomb card\n"
        + "2. You are targetting a country that belongs to you\n"
        + "3. You are targetting a country that is not adjacent to one of your countries\n"
        + "4. That country does not exist in the map\n");
}
}
```

SonarLint Found 23 issues in 1 file

- MainPlayPhaseBusinessCommands.java (23 issues)
- (106, 34) Replace this String concatenation with Text block.
- (168, 35) Replace this String concatenation with Text block.
- (32, 38) Rename this local variable to match the regular expression "[a-z][a-zA-Z0-9]*\$".
- (32, 62) Rename this local variable to match the regular expression "[a-z][a-zA-Z0-9]*\$".
- (32, 78) Rename this local variable to match the regular expression "[a-z][a-zA-Z0-9]*\$".
- (60, 39) Rename this local variable to match the regular expression "[a-z][a-zA-Z0-9]*\$".
- (60, 63) Rename this local variable to match the regular expression "[a-z][a-zA-Z0-9]*\$".
- (60, 89) Rename this local variable to match the regular expression "[a-z][a-zA-Z0-9]*\$".
- (60, 110) Rename this local variable to match the regular expression "[a-z][a-zA-Z0-9]*\$".
- (91, 36) Rename this local variable to match the regular expression "[a-z][a-zA-Z0-9]*\$".
- (91, 60) Rename this local variable to match the regular expression "[a-z][a-zA-Z0-9]*\$".

Automatic analysis is enabled

Rule Locations

String multiline concatenation should be replaced with Text Blocks

Consistency issue | Not conventional | Maintainability | java:S6126

Learn more

Why is this an issue? **More Info**

String textBlock = ""

```
<html>
<body>
<tag>
</tag>
</body>
</html>"";
```

The screenshot shows the IntelliJ IDEA IDE with the file `MainPlayPhaseBusinessCommands.java` open. The editor displays a method `showMap()` with a multiline string concatenation on line 168. The SonarLint panel at the bottom shows 23 issues in 1 file. The first two issues are highlighted: (106, 34) and (168, 35), both with the message "Replace this String concatenation with Text block." The rule details on the right explain that multiline concatenation should be replaced with Text Blocks for better maintainability.

```
Order airLift = new AirLiftOrder(p_currentPlayer, countryFrom, countryTo, p_numArmies);
if(airLift.valid()) {
    p_currentPlayer.addOrder(airLift);
    return new ResponseWrapper(200, " AirLift order added in queue");
}else {
    return new ResponseWrapper(204, "One of the following occurred: \n"
        + "1. You do not possess an airLift card\n"
        + "2. Country you are airLifting from does not belongs to you\n"
        + "3. Country you are airLifting to does not belongs to you\n"
        + "4. That country does not exist in the map\n");
}
}
```

SonarLint Found 23 issues in 1 file

- MainPlayPhaseBusinessCommands.java (23 issues)
- (106, 34) Replace this String concatenation with Text block.
- (168, 35) Replace this String concatenation with Text block.
- (32, 38) Rename this local variable to match the regular expression "[a-z][a-zA-Z0-9]*\$".
- (32, 62) Rename this local variable to match the regular expression "[a-z][a-zA-Z0-9]*\$".
- (32, 78) Rename this local variable to match the regular expression "[a-z][a-zA-Z0-9]*\$".
- (60, 39) Rename this local variable to match the regular expression "[a-z][a-zA-Z0-9]*\$".
- (60, 63) Rename this local variable to match the regular expression "[a-z][a-zA-Z0-9]*\$".
- (60, 89) Rename this local variable to match the regular expression "[a-z][a-zA-Z0-9]*\$".
- (60, 110) Rename this local variable to match the regular expression "[a-z][a-zA-Z0-9]*\$".
- (91, 36) Rename this local variable to match the regular expression "[a-z][a-zA-Z0-9]*\$".
- (91, 60) Rename this local variable to match the regular expression "[a-z][a-zA-Z0-9]*\$".

Automatic analysis is enabled

Rule Locations

String multiline concatenation should be replaced with Text Blocks

Consistency issue | Not conventional | Maintainability | java:S6126

Learn more

Why is this an issue? **More Info**

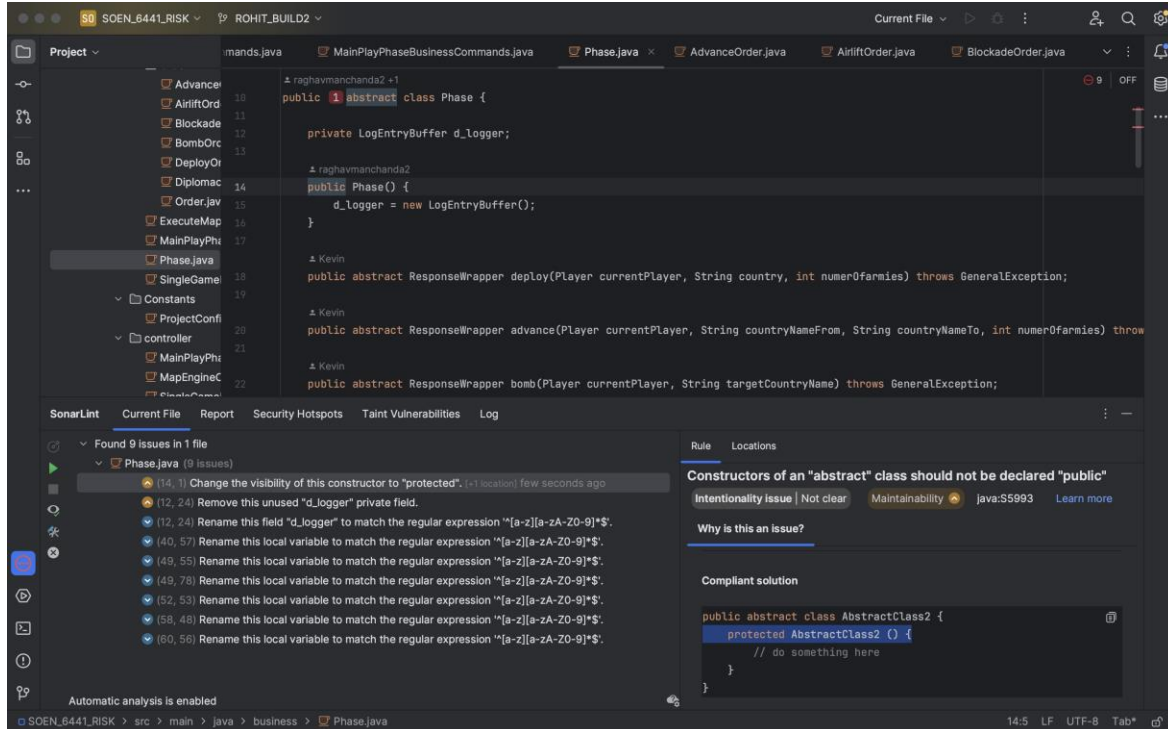
String textBlock =

```
"<html>\n" +
" <body>\n" +
" <tag>\n" +
" </tag>\n" +
" </body>\n" +
"</html>";
```


In Phase.java

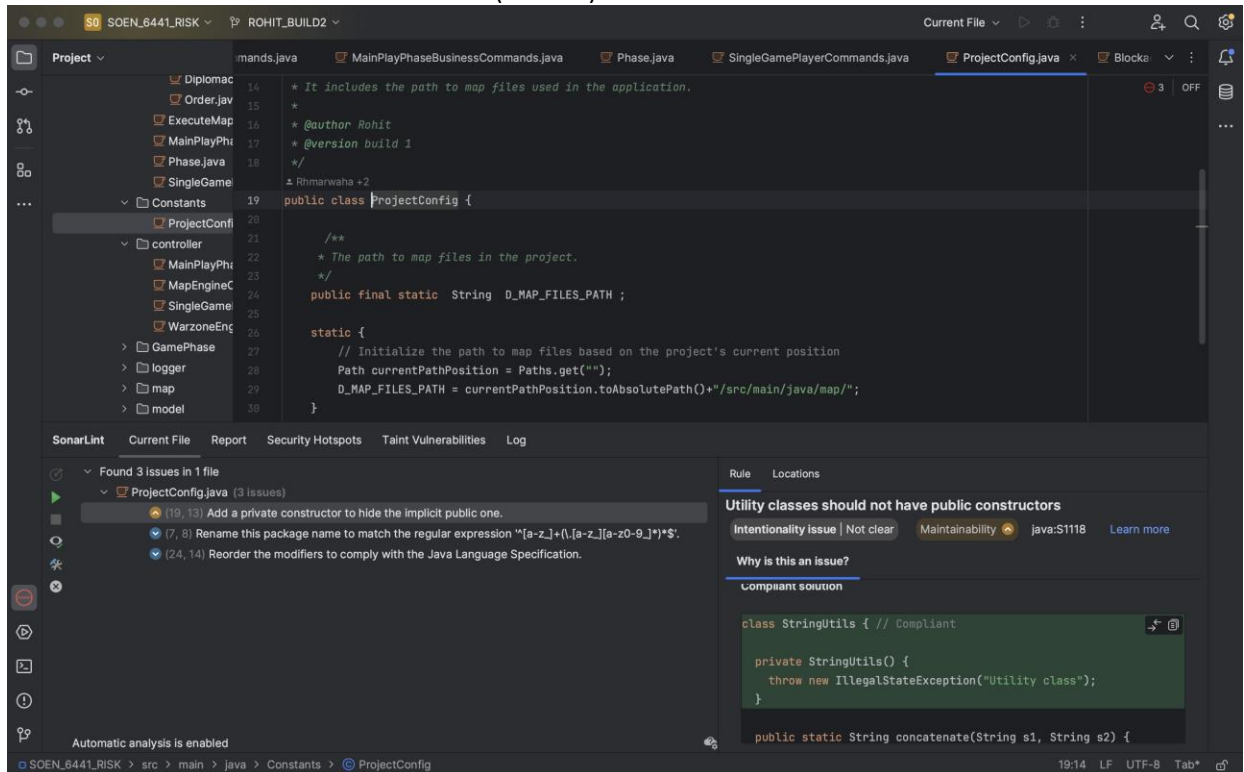
Problem: Line 14 change the visibility for the constructor

Solution: Use Protected to the constructor



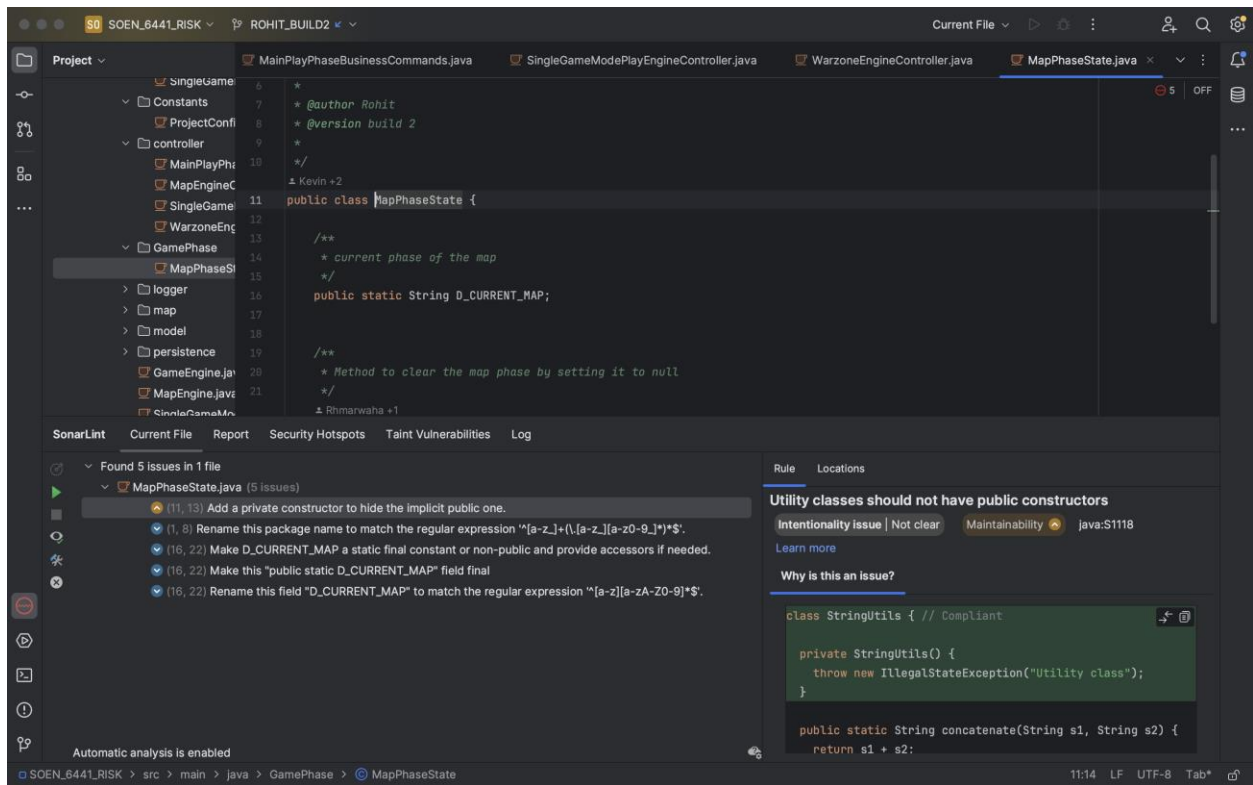
In ProjectConfig.java

Solution: Add Private for the constructor (Line 19)



In MapPhaseState.java

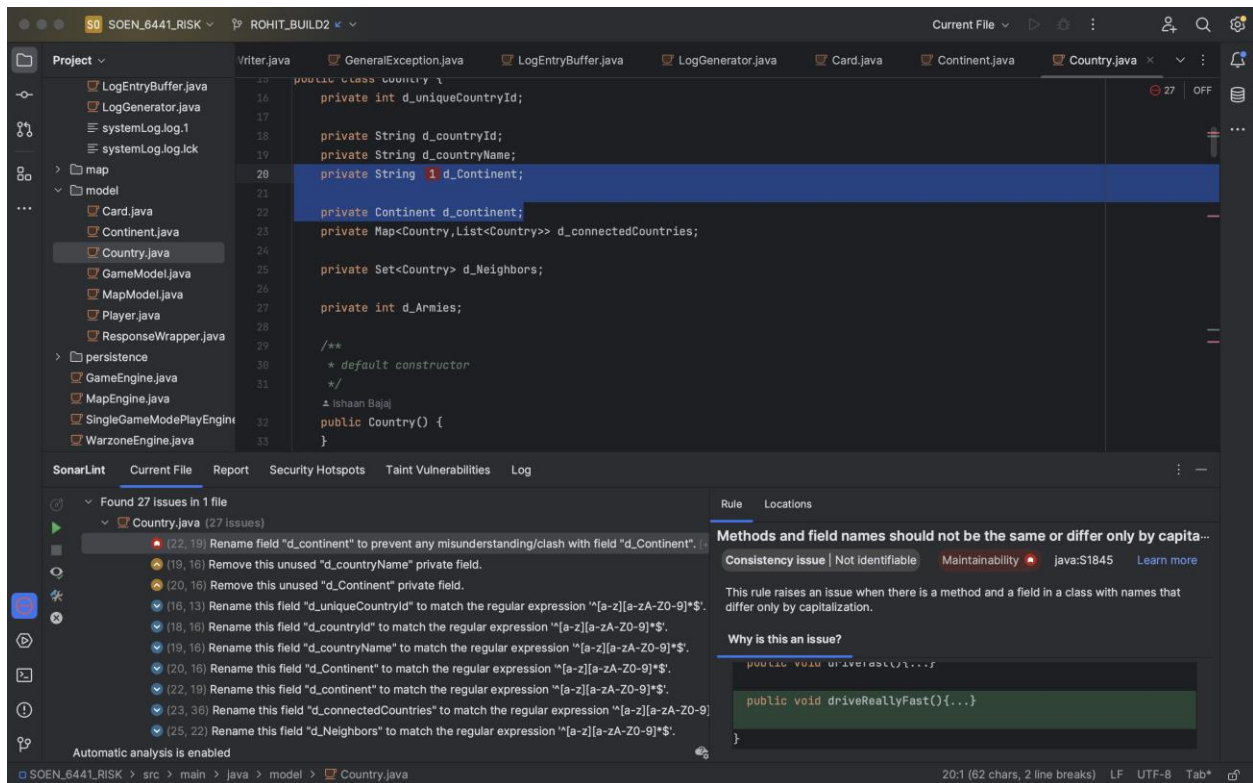
Solution1: To add private constructor (Line 11)



Solution2: Line 16- Make the field D_CURRENT_MAP final.

In Country.java

Problem: Rename the d_Continent of different classes (Line 20,22) or can remove the private variable in line 20.



In MapFileAlteration.java

Problem: Refactor the readMapFile and writeMapFile Methods to reduce the cognitive complexity

Solution: To reduce the cognitive complexity we can break it down to smaller methods to make them more focused

