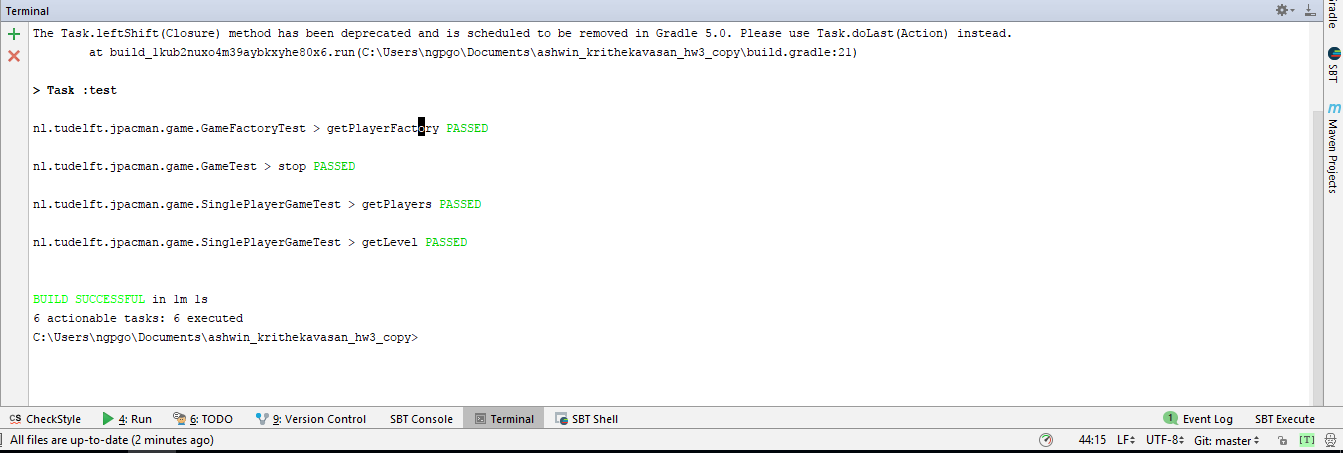
**README**

The main program to launch application is in **src/main/java/nl.tudelft.jpacman/Launcher.java**

**Gradle and SBT builds:** Both Gradle and SBT builds have been included in the project

**Application Run Using Gradle:**

**Gradle Build** - Type “ **Gradle build ”** to build Gradle.



**Gradle Tasks:**

**Run OriginalApp task –** Runs the original application.

**Run Mutate task-** Mutates the original class files (also the task depends on Run Original jar task).

**Run MutatedApp task -** Runs the mutated jar files

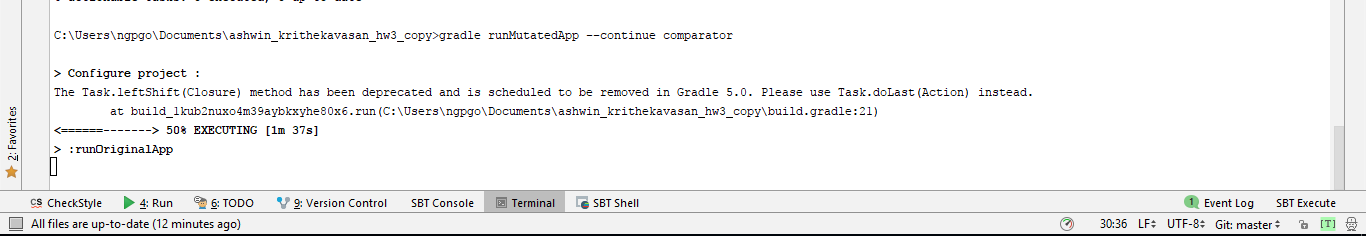
**Comparator task -** which invokes the comparator class and displays the error line which causes the application and therefore build to fail.



**Gradle Run:** To run the original jar file and mutated jar file type in

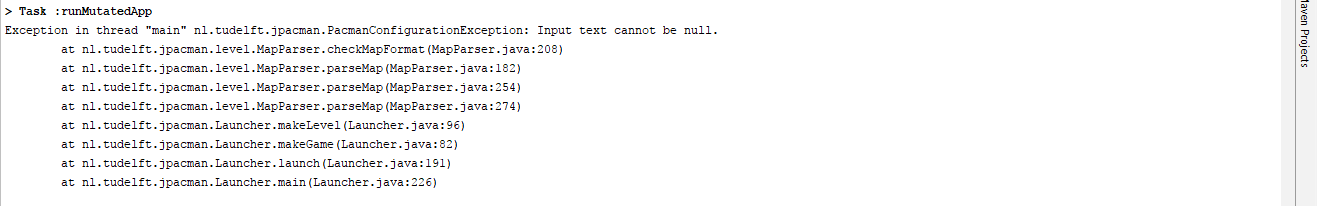
“**gradle runMutatedApp --continue comparator**”

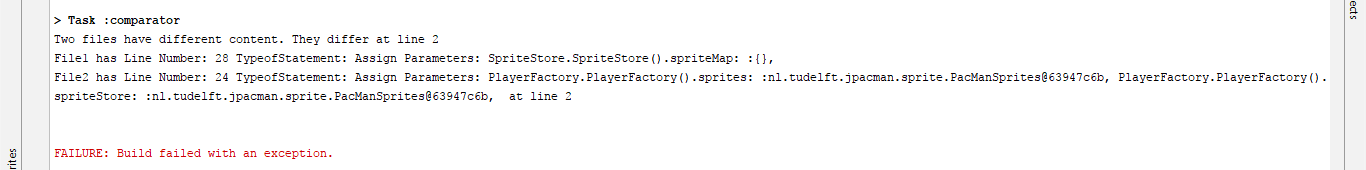
“**gradle runMutatedApp --continue comparator**”

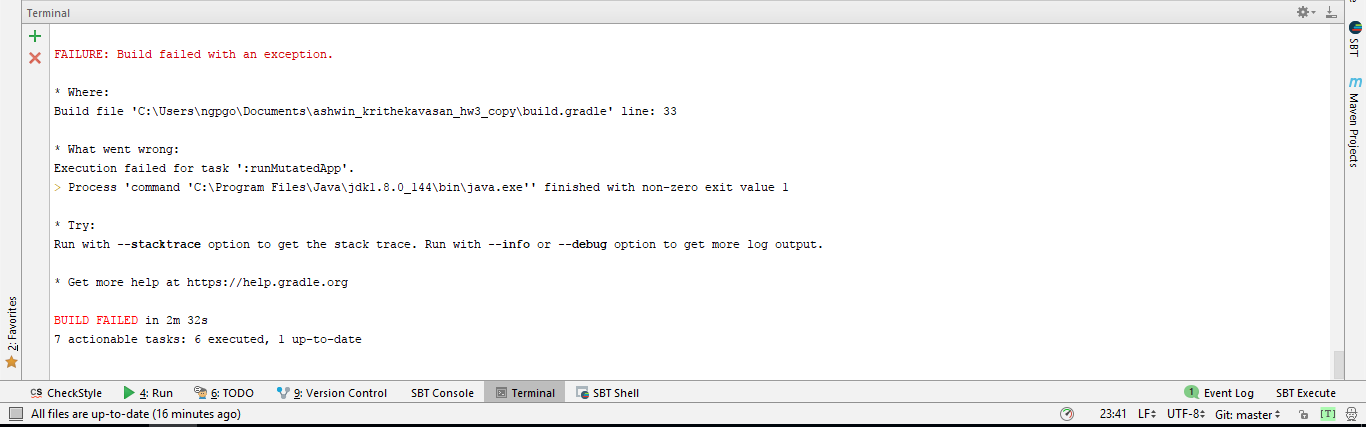


This will execute the run original app task, run mutated app task and generate logging text for both the runs.

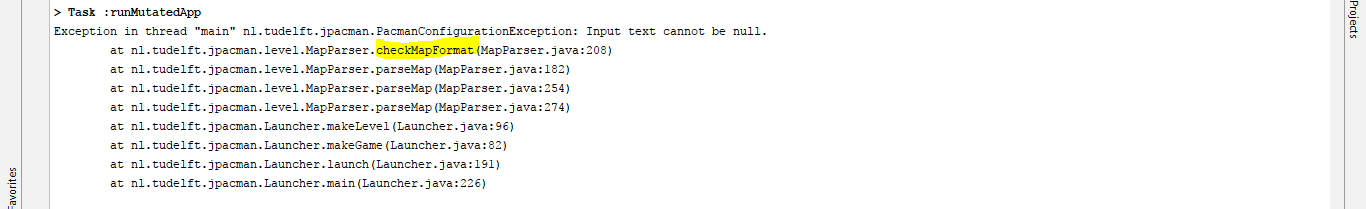
Also, it will run a comparator task to compare both the text files generated by running original jar and mutated jar.



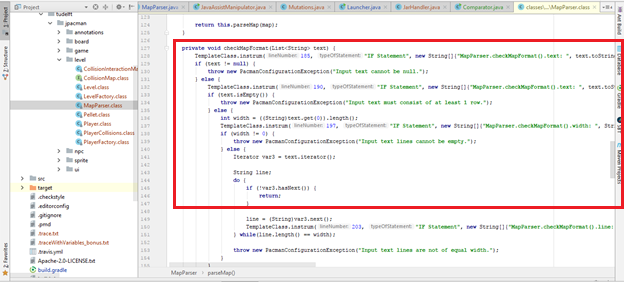




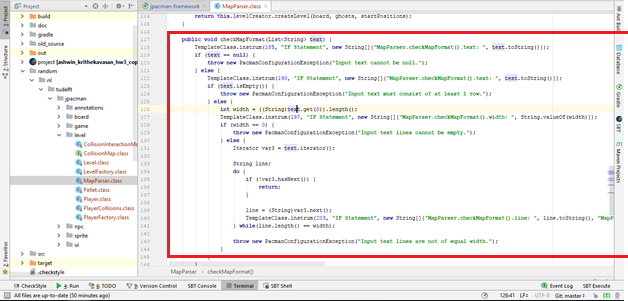
**The build fails** because the of difference in mutated class file and original class file (Screenshot attached below).We can see that the operator is changed from == to != which causes application to throw exception, however the trace of the application is captured till it throws the application



Original class File:



Mutated class file:



**Mutations Implemented: ( the mutations are implemented on all the class files).**

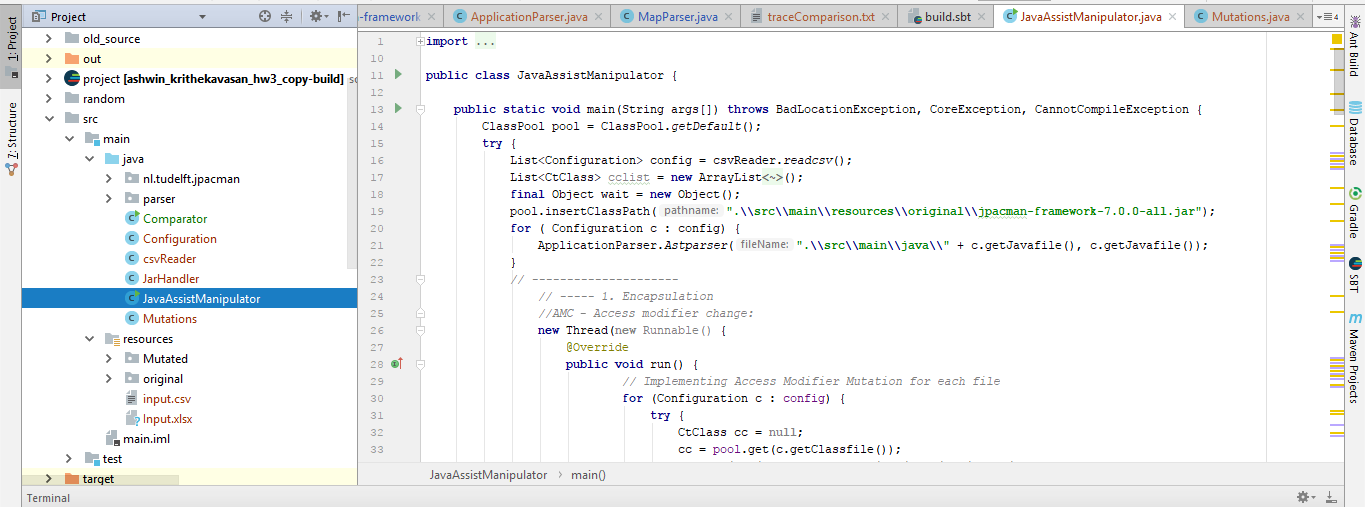
1. Encapsulation – AMC

2. Polymorphism - IPC – Explicit call of a parent’s constructor deletion:

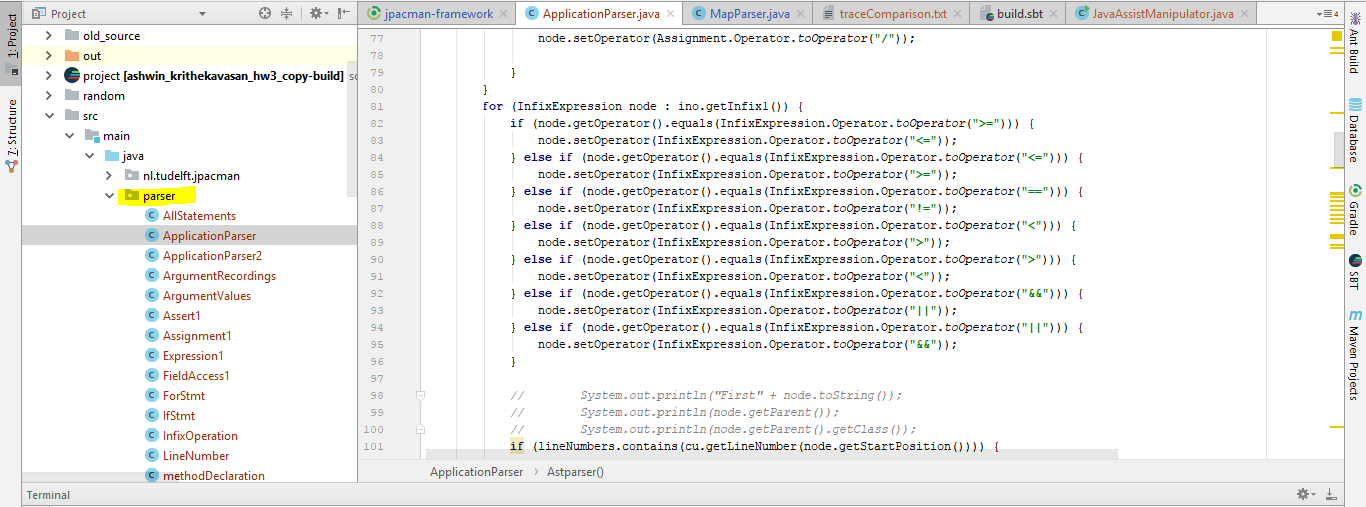
3. Arithmetic Operator Mutation  
4. Relational Operator Mutation  
5. Logical Operator Mutation  
6. Conditional Operator Mutation  
7. Shift Operator Mutation  
8. Assignment Operator Mutation

**Java Assist:** \*\src\main\java\JavaAssistManipulator.java

This program creates the mutated class files using parsed programs and implements mutations.

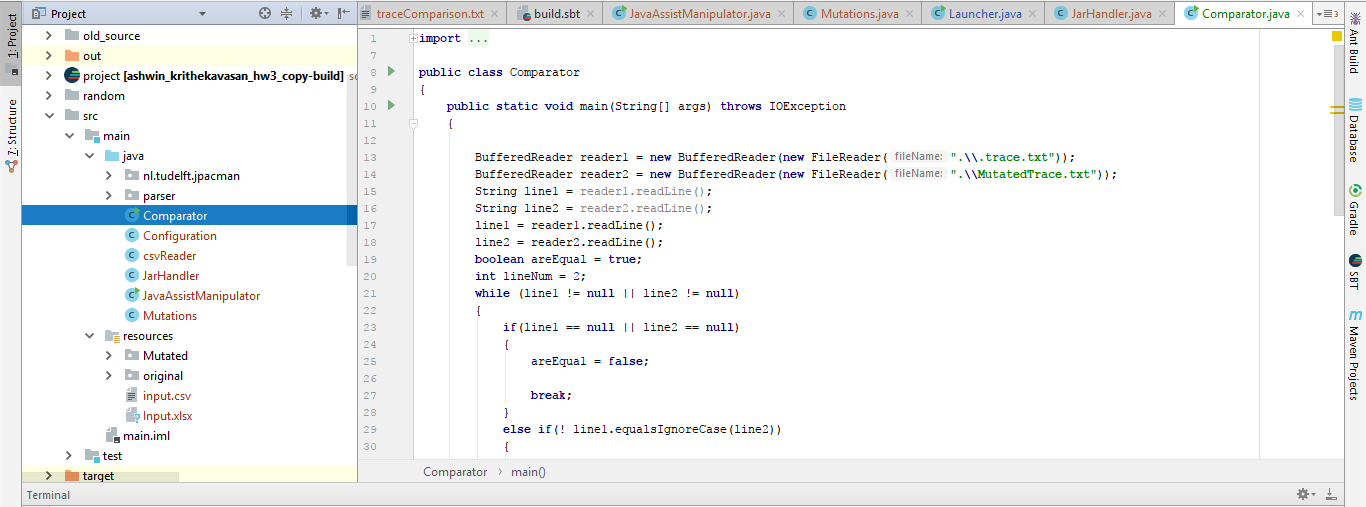


**Parser logic** isin the path : \*\src\main\java\parser\ApplicationParser.java



**Comparator**: \*\src\main\java\Comparator.java

This program compares the logging text generated by running mutated application and original application

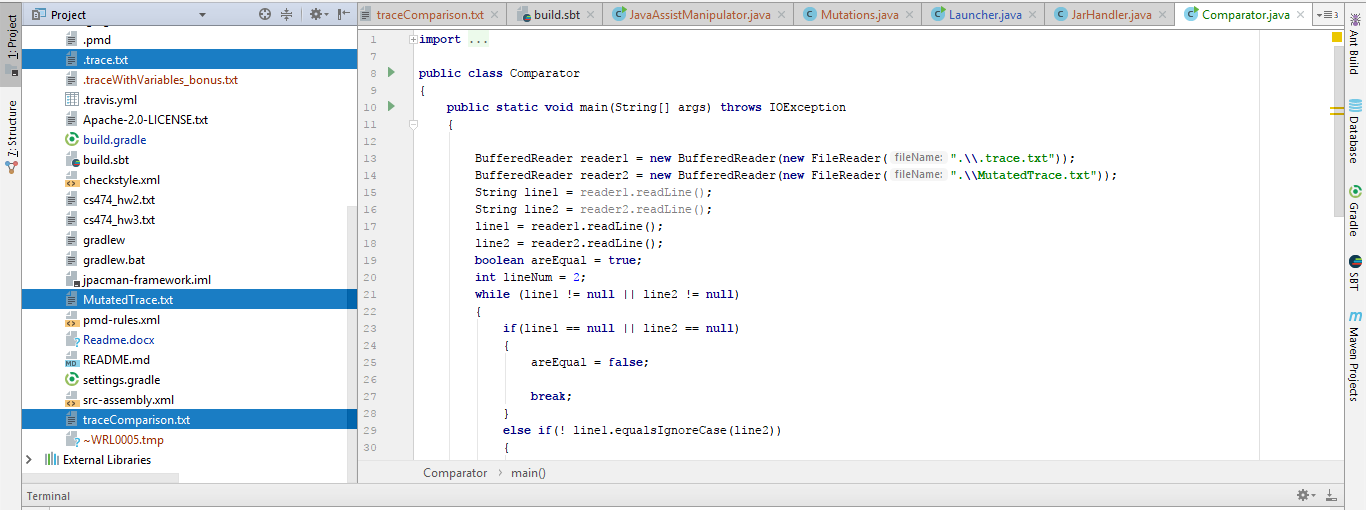


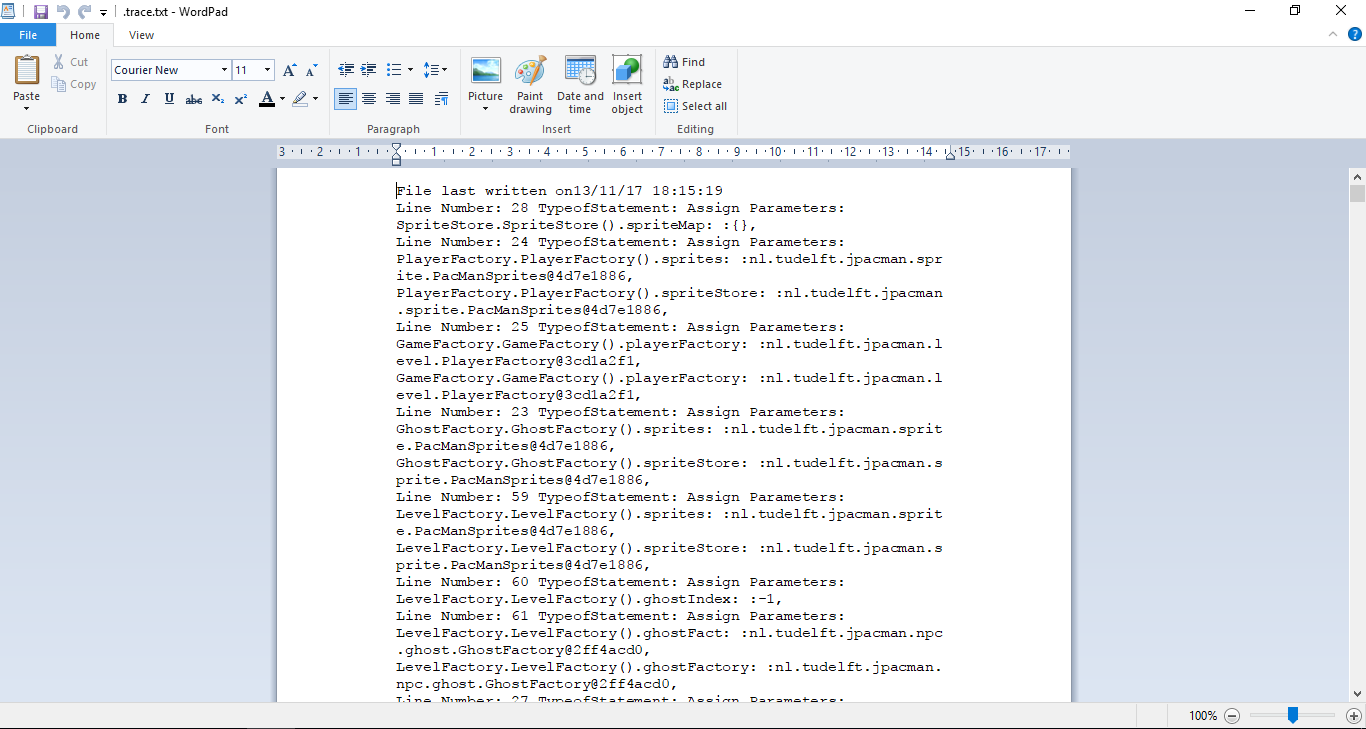
**Text Files:**

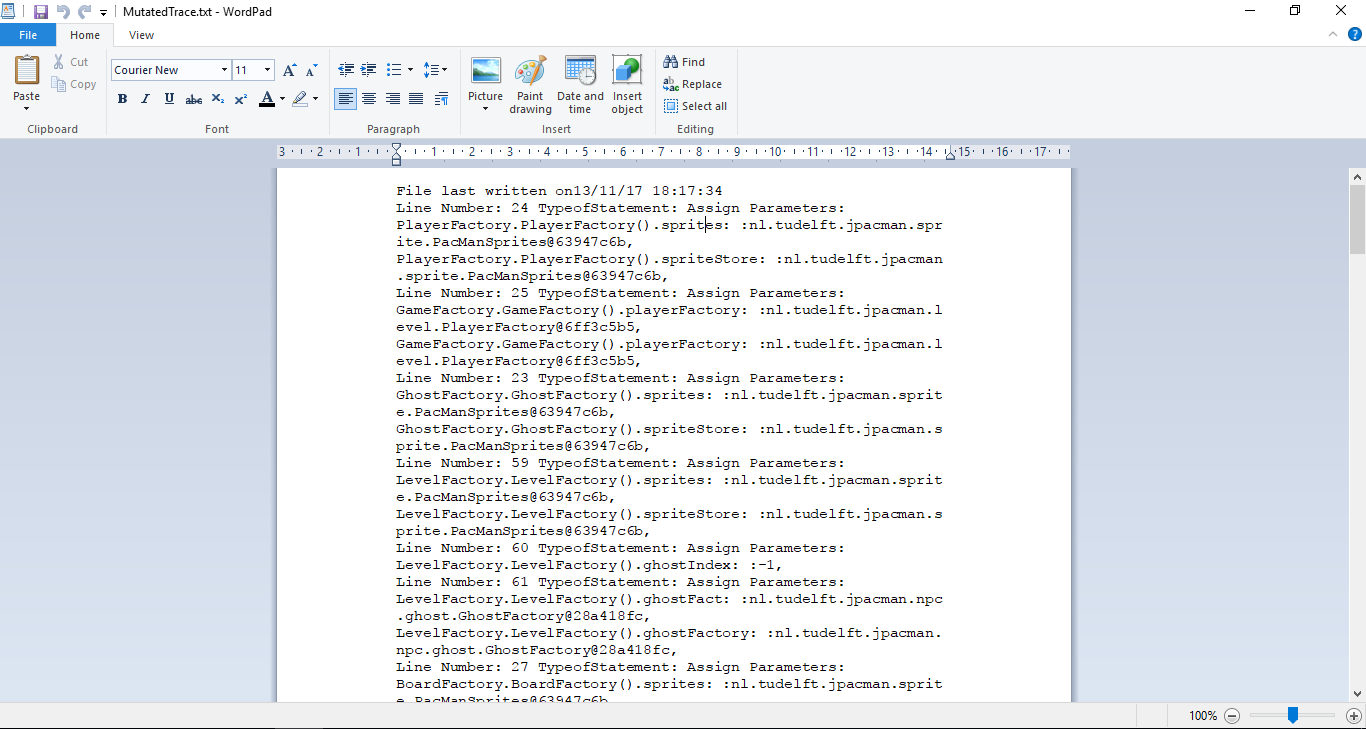
\*\.trace.txt – Generated by running the original application

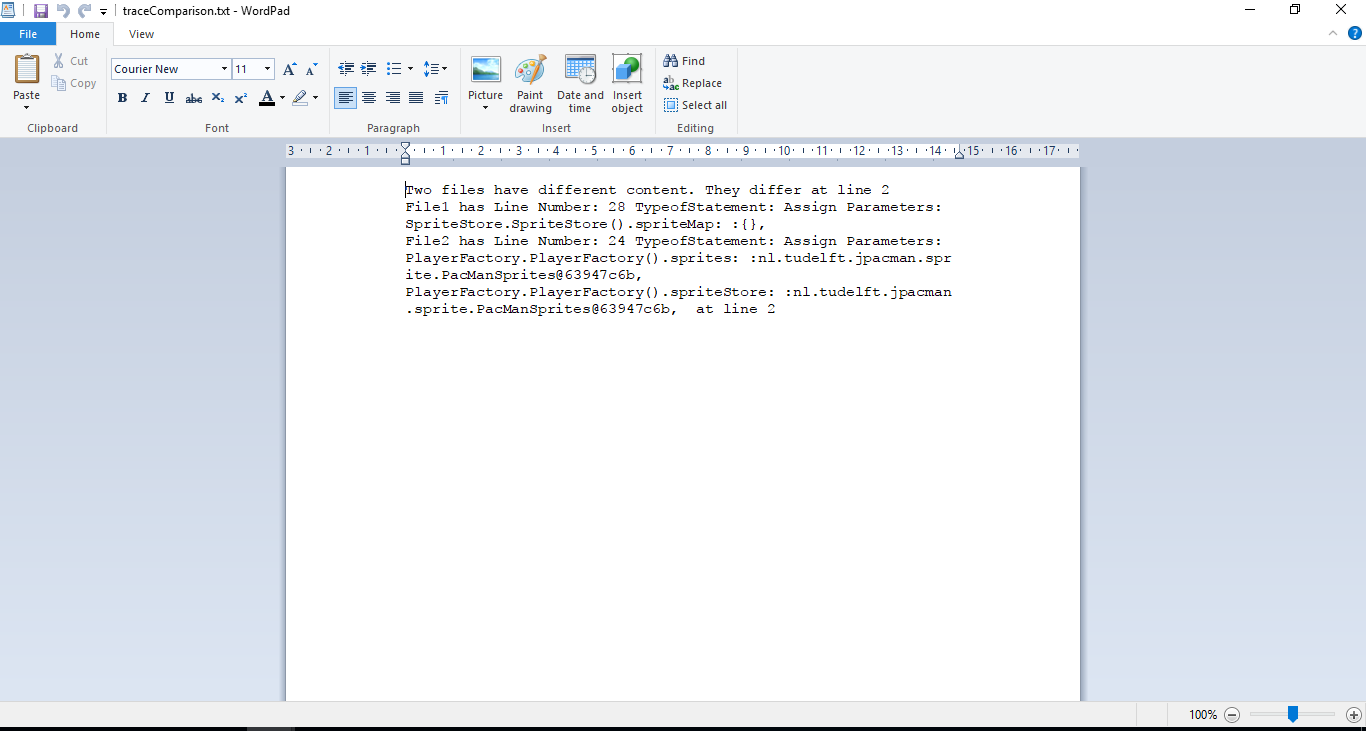
\*\MutatedTrace.txt – Generated by running the Mutated application

\*\traceComparison.txt – Generated by running the comparison module.









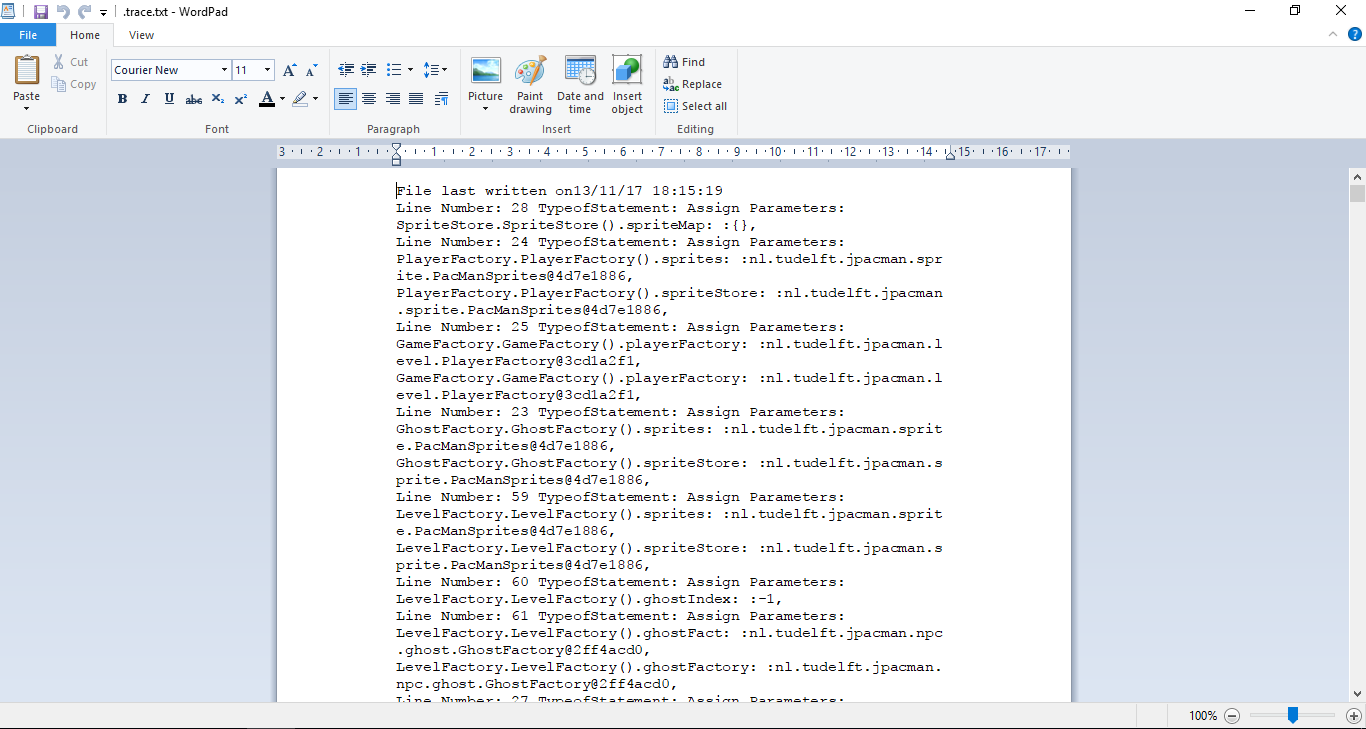
**SBT Build and Run:**

**Type the following command in terminal**

java -jar .\src\main\resources\original\jpacman-framework-7.0.0-all.jar

to run original pacman application. The trace for the original application run is created with a latest time stamp.

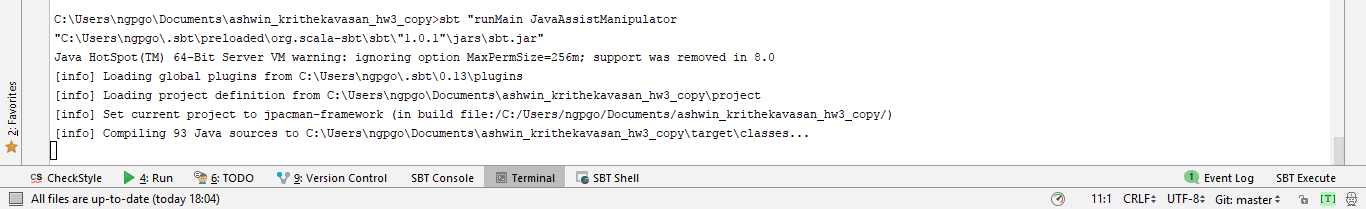




**Type the following command**

sbt "runMain JavaAssistManipulator"

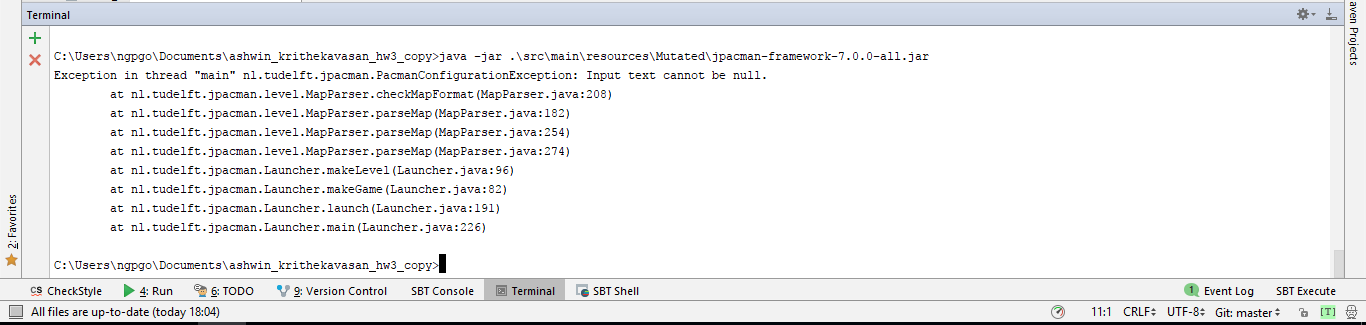
It will take original class files from the jar and implement mutations.

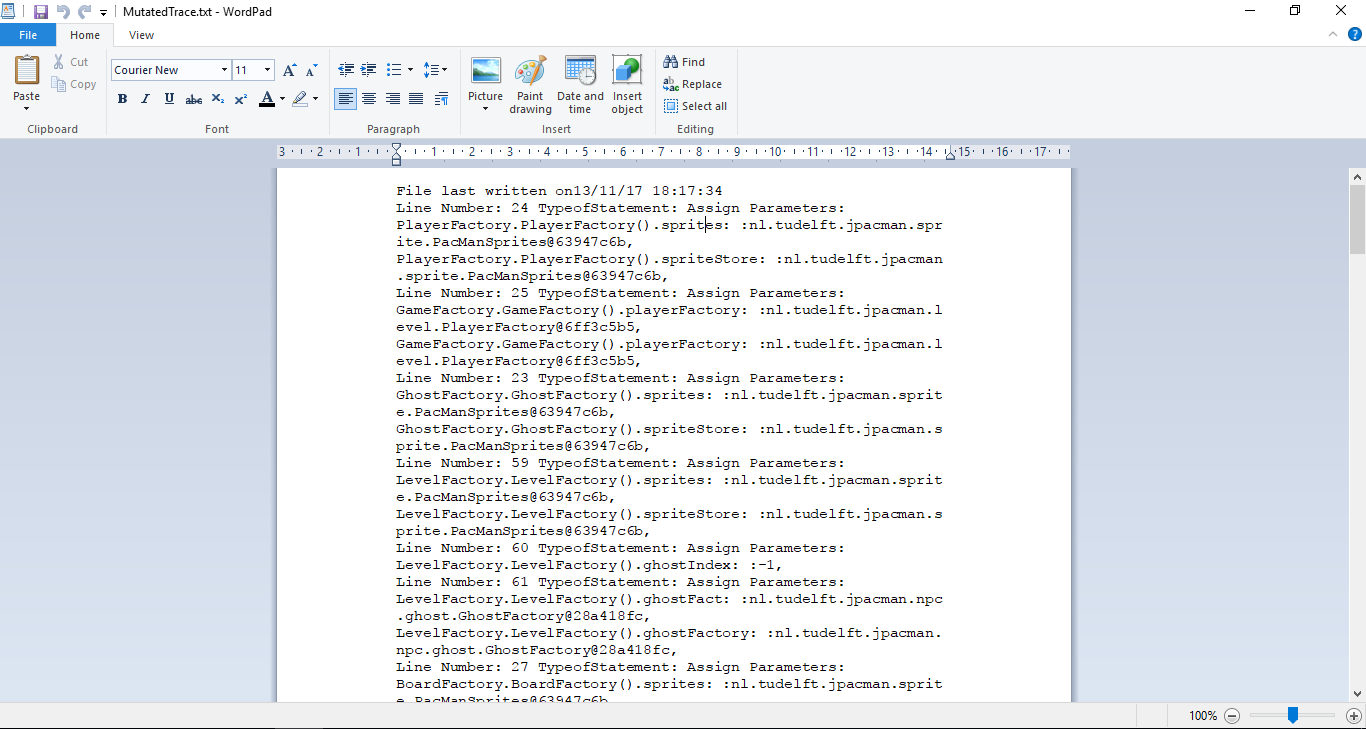


**Type the following command**

java -jar .\src\main\resources\Mutated\jpacman-framework-7.0.0-all.jar

The mutated class are written in this jar file and the mutated application is run. It will call an exception because of the mutation. It will also create the mutated text with the latest time stamp.





**The mutated text file is generated up to the point mutated application throws exception.**

**Type the following command** “

"runMain Comparator"

sbt "runMain Comparator"

It will compare the mutated text file and text file generated from the original application.

