CSCI 5308 – Advanced Topics in Software Development

Assignment 3

Work done by,

Name: Guturu Rama Mohan Vishnu

Banner ID: B00871849

Email: rm286720@dal.ca

Set – I Refactoring Techniques:

Rename method/variable

[1]

The variable “STATE_CREATE_INDEX” in this file should be renamed to “STATE_RECREATE_INDEX” as the purpose of the variable is to be used when re-creating the index.

- Refactoring name: Rename variable
- Location: h2/src/main/org/h2/api/DatabaseEventListener.java
- Link of the file before refactoring: https://github.com/h2database/h2database/blob/master/h2/src/main/org/h2/api/DatabaseEventListener.java
- Link of the file after refactoring: https://github.com/grmvishnu/h2database/blob/master/h2/src/main/org/h2/api/DatabaseEventListener.java

[2]

The variable “ERROR_OPENING_DATABASE_1” in this file should be renamed to “ERROR_CREATING_LOCK” as the purpose of the variable is to return an error if creating a lock on the database is failed.

- Refactoring name: Rename variable
- Location: h2/src/main/org/h2/api/ErrorCode.java
- Link of the file before refactoring: https://github.com/h2database/h2database/blob/master/h2/src/main/org/h2/api/ErrorCode.java

- Link of the file after refactoring: https://github.com/grmvishnu/h2database/blob/master/h2/src/main/org/h2/api/ErrorCode.java

[3]

After doing one of the refactoring methods, the variable “TYPE_NAME” are present twice in this file, both assigned to different values. So, they both should be renamed as per their respective purpose and one of them should be renamed to “TYPE_NAME_TQ”, to should avoid the conflict of having same names.

- Refactoring name: Rename variable
- Location: h2/src/main/org/h2/util/CacheLRU.java
- Link of the file before refactoring: https://github.com/h2database/h2database/blob/master/h2/src/main/org/h2/util/CacheLRU.java
- Link of the file after refactoring: https://github.com/grmvishnu/h2database/blob/master/h2/src/main/org/h2/util/CacheLRU.java

[4]

After doing one of the refactoring methods, the variable “TYPE_NAME” are present twice in this file, both assigned to different values. So, they both should be renamed as per their respective purpose and one of them should be renamed to “TYPE_NAME_LRU”, to should avoid the conflict of having same names.

- Refactoring name: Rename variable
- Location: h2/src/main/org/h2/util/CacheLRU.java

- Link of the file before refactoring: https://github.com/h2database/h2database/blob/master/h2/src/main/org/h2/util/CacheLRU.java
- Link of the file after refactoring: https://github.com/grmvishnu/h2database/blob/master/h2/src/main/org/h2/util/CacheLRU.java

Extract method

[1]

There is some part of code in methods “executeUpdate()” and “executeLargeUpdate()” which is common and it can be extracted out and can be put into a separate method “checkForDDL()”.

- Refactoring name: Extract method
- Location: h2/src/main/org/h2/jdbc/JdbcCallableStatement.java
- Link of the file before refactoring: https://github.com/h2database/h2database/blob/master/h2/src/main/org/h2/jdbc/JdbcCallableStatement.java
- Link of the file after refactoring: https://github.com/grmvishnu/h2database/blob/master/h2/src/main/org/h2/jdbc/JdbcCallableStatement.java

[2]

The method “getStackTraceForThread()” has a Cylometric Complexity of 10, which means the method has 10 different flows of execution. This method can be simplified a bit by extracting some part of code from the method and making it as a separate method “buildString()”, and calling the new method in this getStackTraceForThread method.

- Refactoring name: Extract method
- Location: h2/src/main/org/h2/util/AbbaLockingDetector.java

- Link of the file before refactoring: https://github.com/h2database/h2database/blob/master/h2/src/main/org/h2/util/AbbaLockingDetector.java
- Link of the file after refactoring: https://github.com/grmvishnu/h2database/blob/master/h2/src/main/org/h2/util/AbbaLockingDetector.java

Set – II Refactoring Techniques:

Move method/field

[1]

The field “TYPE_NAME” in the CacheTQ.java class should be moved to CacheLRU.java class since both these classes are in a cyclic dependency and the field is defined in CacheTQ class but it is being used in CacheLRU class. So, to refactor this situation, moving the variable from CacheTQ to CacheLRU is the best solution.

- Refactoring name: Move field
- Location: h2/src/main/org/h2/util/CacheLRU.java , h2/src/main/org/h2/util/CacheTQ.java
- Link of the files before refactoring: https://github.com/h2database/h2database/blob/master/h2/src/main/org/h2/util/CacheTQ.java , https://github.com/h2database/h2database/blob/master/h2/src/main/org/h2/util/CacheLRU.java
- Link of the files after refactoring: https://github.com/grmvishnu/h2database/blob/master/h2/src/main/org/h2/util/CacheLRU.java , https://github.com/grmvishnu/h2database/blob/master/h2/src/main/org/h2/util/CacheTQ.java

[2]

The method “currentNanoTime()” in the Utils.java class should be moved to DateTimeUtils.java class since the method purpose is to return the current nano value of the time and all the date-time conversion functions are being placed in DateTimeUtils.java class.

- Refactoring name: Move method
- Location: h2/src/main/org/h2/util/Utils.java , h2/src/main/org/h2/util/DateTimeUtils.java
- Link of the files before refactoring: https://github.com/h2database/h2database/blob/master/h2/src/main/org/h2/util/Utils.java , https://github.com/h2database/h2database/blob/master/h2/src/main/org/h2/util/DateTimeUtils.java
- Link of the files after refactoring: https://github.com/grmvishnu/h2database/blob/master/h2/src/main/org/h2/util/Utils.java , https://github.com/grmvishnu/h2database/blob/master/h2/src/main/org/h2/util/DateTimeUtils.java

[3]

The method “currentNanoTimePlusMillis()” in the Utils.java class should be moved to DateTimeUtils.java class since the method purpose is to return the current nano value of the time including the specified offset and all the date-time conversion functions are being placed in DateTimeUtils.java class.

- Refactoring name: Move method
- Location: h2/src/main/org/h2/util/Utils.java , h2/src/main/org/h2/util/DateTimeUtils.java
- Link of the files before refactoring: https://github.com/h2database/h2database/blob/master/h2/src/main/org/h2/util/Utils.java , https://github.com/h2database/h2database/blob/master/h2/src/main/org/h2/util/DateTimeUtils.java
- Link of the files after refactoring: https://github.com/grmvishnu/h2database/blob/master/h2/src/main/org/h2/util/Utils.java , https://github.com/grmvishnu/h2database/blob/master/h2/src/main/org/h2/util/DateTimeUtils.java

[4]

The method “nanoTimePlusMillis()” in the Utils.java class should be moved to DateTimeUtils.java class since the method purpose is to return the nano value of the time including the specified offset and all the date-time conversion functions are being placed in DateTimeUtils.java class.

- Refactoring name: Move method
- Location: h2/src/main/org/h2/util/Utils.java , h2/src/main/org/h2/util/DateTimeUtils.java
- Link of the files before refactoring: https://github.com/h2database/h2database/blob/master/h2/src/main/org/h2/util/Utils.java ,

https://github.com/h2database/h2database/blob/master/h2/src/main/org/h2/util/DateTimeUtils.java

- Link of the files after refactoring: https://github.com/grmvishnu/h2database/blob/master/h2/src/main/org/h2/util/Utils.java , https://github.com/grmvishnu/h2database/blob/master/h2/src/main/org/h2/util/DateTimeUtils.java

Extract class

[1]

When an abstraction has more than one responsibility assigned to it, in that case, to resolve the issue of multiple responsibilities, we apply the refactoring method Extract Class. The class file “Session.java” had possessed Multifaceted Abstraction, which means that the class had more than one responsibility. So, this refactoring has been applied on that class and the methods which do not interact with the other methods in the same class, and also which do not belong here are being taken out and put in a new class created “StaticSetting.java” as a process of this refactoring. Many changes had to be made in multiple files since this class was imported, called and used in many places.

- Refactoring name: Extract class
- Location: h2/src/main/org/h2/engine/Session.java , h2/src/main/org/h2/engine/StaticSetting.java
- Link of the files before refactoring: https://github.com/h2database/h2database/blob/master/h2/src/main/org/h2/engine/Session.java
- Link of the files after refactoring: https://github.com/grmvishnu/h2database/blob/master/h2/src/main/org/h2/engine/Session.java , https://github.com/grmvishnu/h2database/blob/master/h2/src/main/org/h2/engine/StaticSetting.java

[2]

When an abstraction has more than one responsibility assigned to it, in that case, to resolve the issue of multiple responsibilities, we apply the refactoring method Extract Class. The class file “Session.java” had possessed Multifaceted Abstraction, which means that the class had more than one responsibility. So, this refactoring has been applied on that class and the methods which do not interact with the other methods in the same class, and also which do not belong here are being taken out and put in a new class created “DynamicSetting.java” as a process of this refactoring. Many changes had to be made in multiple files since this class was imported, called and used in many places.

- Refactoring name: Extract class
- Location: h2/src/main/org/h2/engine/Session.java , h2/src/main/org/h2/engine/DynamicSetting.java
- Link of the files before refactoring: https://github.com/h2database/h2database/blob/master/h2/src/main/org/h2/engine/Session.java
- Link of the files after refactoring: https://github.com/grmvishnu/h2database/blob/master/h2/src/main/org/h2/engine/Session.java , https://github.com/grmvishnu/h2database/blob/master/h2/src/main/org/h2/engine/DynamicSetting.java

Push-down variable/method

[1]

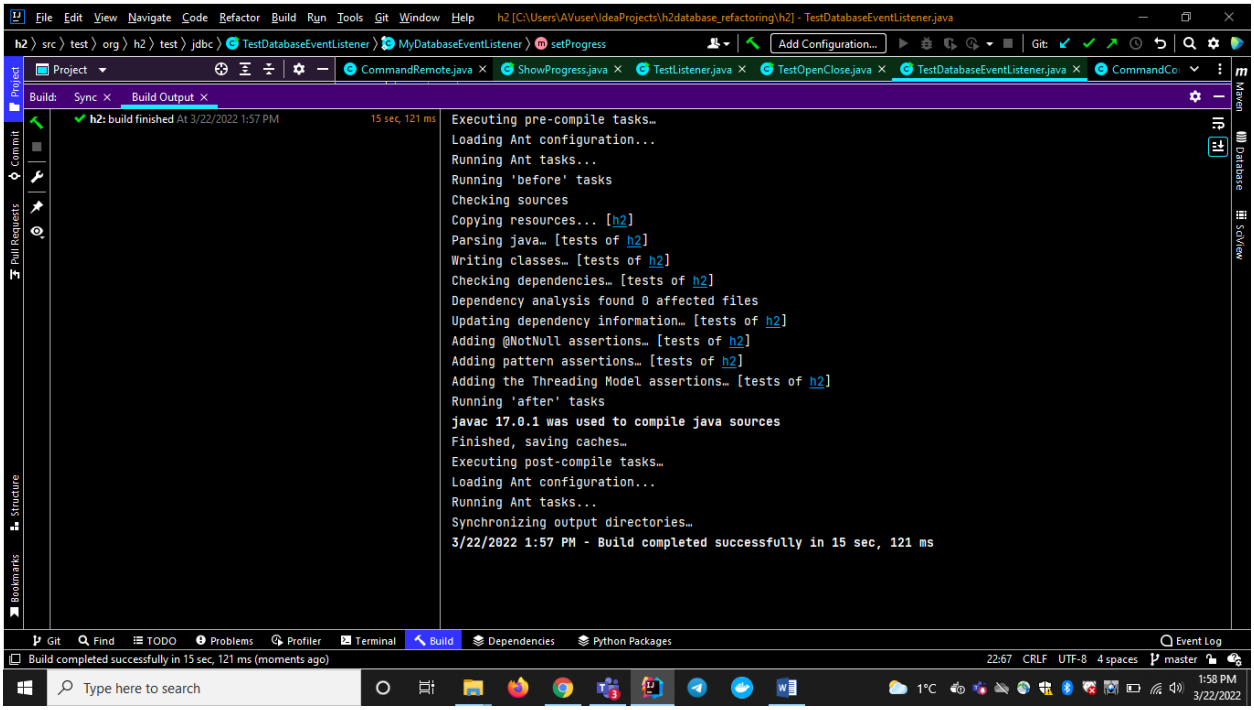
Rebellious Hierarchy happens when the methods of the subclass rejects the methods of the superclass or interface. An interface “CommandInterface.java” has two classes which implement it: “Command.java” and “CommandRemote.java”. In this case, the interface “CommandInterface.java” has some methods which are being rejected by the class “Command.java” but they are being used by the class “CommandRemote.java”. So, by following the push-down method refactoring, I am making the super type generic and moving those few methods from interface which are being rejected by subclass into the subclass. Those methods are “getCommandType()” and “isQuery()”. Many other files and many classes are using many methods from this interface in the program.

- Refactoring name: Push-down method
- Location: h2/src/main/org/h2/command/CommandInterface.java , h2/src/main/org/h2/command/Command.java , h2/src/main/org/h2/command/CommandRemote.java
- Link of the files before refactoring: https://github.com/h2database/h2database/blob/master/h2/src/main/org/h2/command/CommandInterface.java , https://github.com/h2database/h2database/blob/master/h2/src/main/org/h2/command/Command.java , https://github.com/h2database/h2database/blob/master/h2/src/main/org/h2/command/CommandRemote.java
- Link of the files after refactoring: https://github.com/grmvishnu/h2database/blob/master/h2/src/main/org/h2/command/CommandInterface.java ,

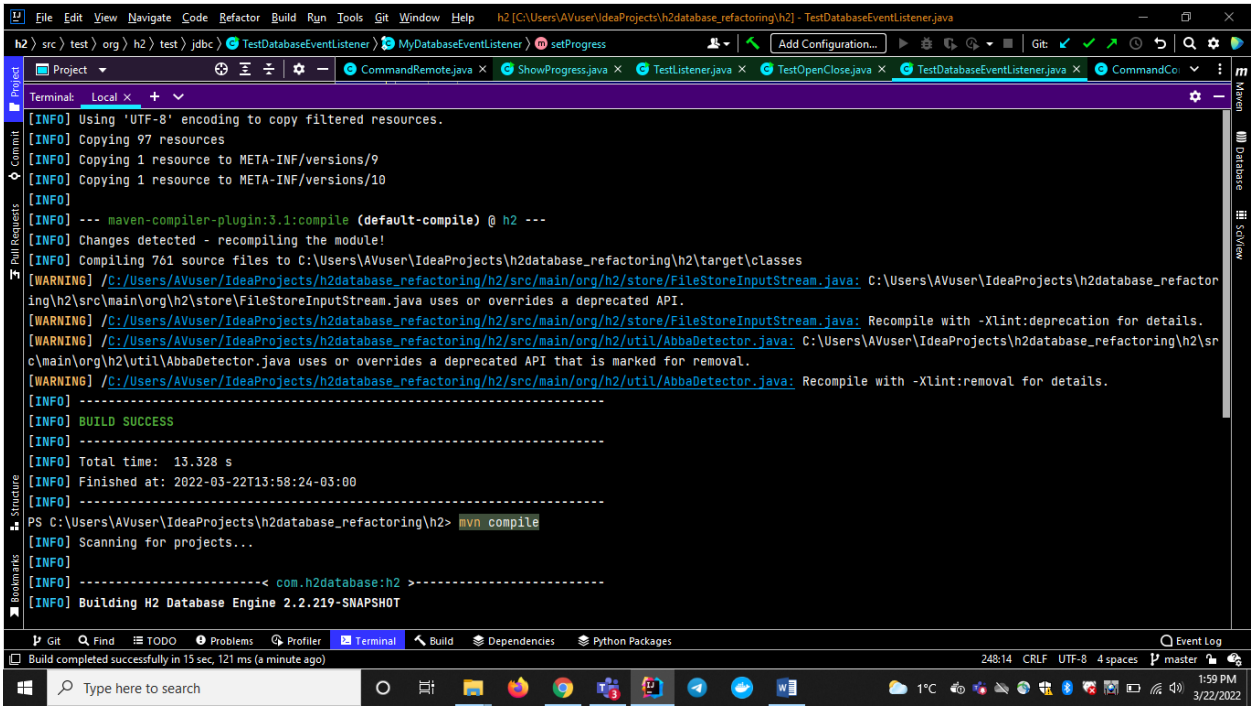
https://github.com/grmvishnu/h2database/blob/master/h2/src/main/org/h2/command/Command.java , https://github.com/grmvishnu/h2database/blob/master/h2/src/main/org/h2/command/CommandRemote.java

Screenshot of the code compiling after refactoring as a proof:

“Build project”

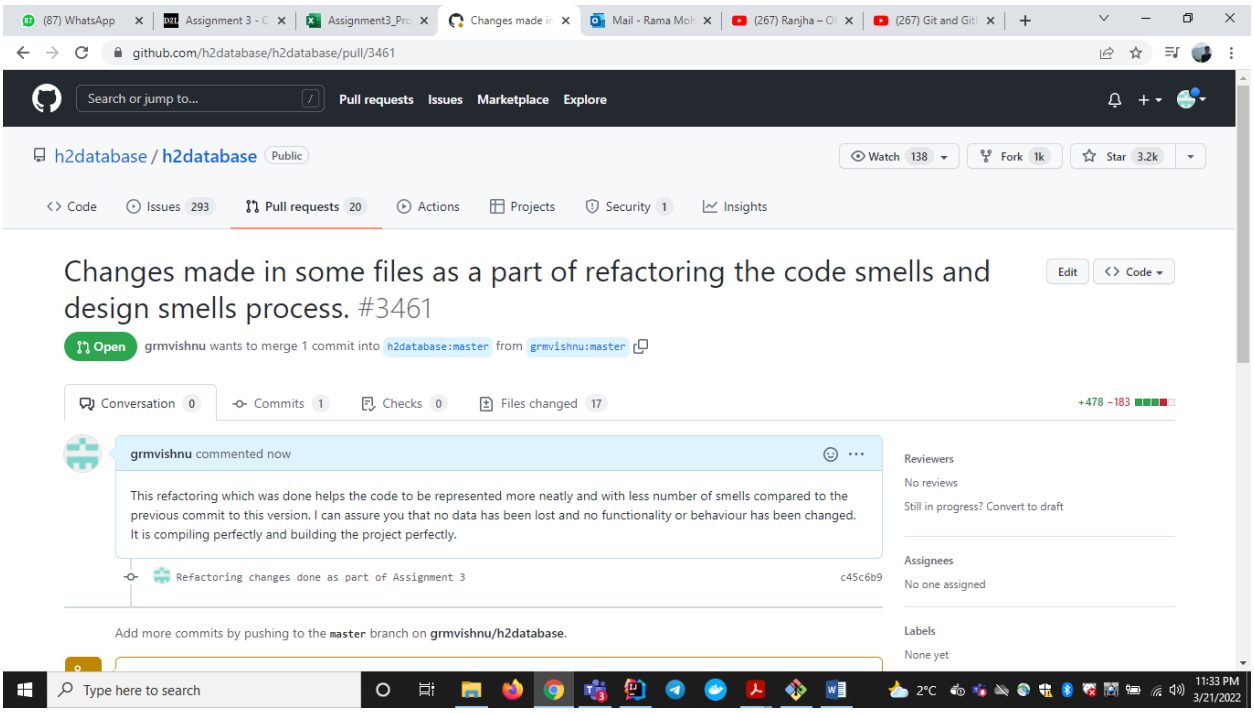


“mvn compile”


File Edit View Navigate Code Refactor Build Run Tools Git Window Help
h2 [C:\Users\AVuser\IdeaProjects\h2database_refactoring\h2] - TestDatabaseEventListener.java
h2 〉 src 〉 test 〉 org 〉 h2 〉 test 〉 jdbc 〉 TestDatabaseEventListener 〉 MyDatabaseEventListener 〉 setProgress
Add Configuration...
Git:
Project
CommandRemote.java
ShowProgress.java
TestListener.java
TestOpenClose.java
TestDatabaseEventListener.java
Terminal: Local
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 97 resources
[INFO] Copying 1 resource to META-INF/versions/9
[INFO] Copying 1 resource to META-INF/versions/10
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ h2 ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 761 source files to C:\Users\AVuser\IdeaProjects\h2database_refactoring\h2\target\classes
[WARNING] /C:/Users/AVuser/IdeaProjects/h2database_refactoring/h2/src/main/org/h2/store/FileStoreInputStream.java: C:\Users\AVuser\IdeaProjects\h2database_refactoring\h2\src\main\org\h2\store\FileStoreInputStream.java uses or overrides a deprecated API.
[WARNING] /C:/Users/AVuser/IdeaProjects/h2database_refactoring/h2/src/main/org/h2/store/FileStoreInputStream.java: Recompile with -Xlint:deprecation for details.
[WARNING] /C:/Users/AVuser/IdeaProjects/h2database_refactoring/h2/src/main/org/h2/util/AbbaDetector.java: C:\Users\AVuser\IdeaProjects\h2database_refactoring\h2\src\main\org\h2\util\AbbaDetector.java uses or overrides a deprecated API that is marked for removal.
[WARNING] /C:/Users/AVuser/IdeaProjects/h2database_refactoring/h2/src/main/org/h2/util/AbbaDetector.java: Recompile with -Xlint:removal for details.
[INFO] --
[INFO] BUILD SUCCESS
[INFO] --
[INFO] Total time: 13.328 s
[INFO] Finished at: 2022-03-22T13:58:24-03:00
[INFO] --
PS C:\Users\AVuser\IdeaProjects\h2database_refactoring\h2> mvn compile
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------< com.h2database:h2 >-------------------------
[INFO] Building H2 Database Engine 2.2.219-SNAPSHOT
Git Find TODO Problems Profiler Terminal Build Dependencies Python Packages
Event Log
Build completed successfully in 15 sec, 121 ms (a minute ago)
248:14 CRLF UTF-8 4 spaces master
Type here to search
1°C
1:59 PM
3/22/2022

Pull request to the original repository has been made and the link and the screenshot are provided below:

https://github.com/h2database/h2database/pull/3461



The pull request made by me was not accepted by the original repository owner, at the time of submission.