CS-368

Software Re-Engineering

[B]

***Group Assignment # 2*:**

Write a brief report about the Refactoring support available on your preferred IDE. The IDE chosen by you must have this support. Put this report on your Github repository.

Group No. 1

Group Members:

Areeb Arshad	181400149
Sharif Sadique	18140099
Shoaib Ali	181400182

Submitted to: Sir Fakhar Lodhi

Refactoring in NetBeans:

Refactoring is a technique to restructure the existing code. It is a controlled technique for improving the design of existing code. Refactoring changes the internal structure and design of code, rather than changing its external behavior.

NetBeans provides many refactoring feature to make your code more understandable and readable. IDE refactoring can make the life of developer much easier as it provides much reliable refactoring and easy way to do so. The possibilities of making errors and code crash in manual refactoring is very high while IDE does not make much error as it provide safe refactoring.

Some refactoring features provided by NetBeans are as follow:

1. **Deleting Code Safely:**

   Over time, your code might gather elements that have limited or no usefulness. To make the code easier to maintain, it is desirable to remove as much of this code as possible. However, it might be hard to immediately determine whether you can delete such code without causing errors elsewhere.

2. **Changing a Method's Signature:**

   If you want to change a method's signature, you can use the IDE's Refactor/Change Method Parameters command to update all matching method signatures wherever it is used. Besides this, you can:

   1. Add parameters.
   2. Change the order of parameters.
   3. Change the access modifier for the method.
   4. Remove unused parameters.

   You cannot use the Change Method Parameters command to remove a parameter from a method if the parameter is used in your code.

3. **Extract Interface refactoring:**

   This refactoring helps create a new interface based on a selected type. ReSharper suggests choosing members to be transferred to the new interface. After extraction, the original type is updated to implement the new interface. If the current type already implements any interfaces, those interfaces can also be extracted into the new interface.

4. **Pull Up refactoring:**

   In software engineering, Pull Up refactoring involves moving a member/attribute/ method of a lower hierarchy class to an upper hierarchy class, such as moving the method from a subclass into a super-class.

5. **Push down refactoring:**

   To move a member, a method's implementation, or an implemented clause from the current class to that class' subclasses. It is the opposite to pull-up refactoring.

6. **Renaming:**

   It helps in renaming the code to reflect the purpose of the method or class for which refactoring is being done. NetBeans gives the developer the possibility of renaming the folder where the project is contained to the same name as the project which is a good practice.

7. **Move Inner to Outer Level:**

   Move Inner to Outer Level refactoring converts an inner class to a separate external class declared in its own file. It also gives you the option to declare a field for the current outer class.

8. **Moving Elements**:

   NetBeans enables the developer to easily move classes around different projects. Their compatibility remains the same as all are handled by IDE itself and it does not increase the coupling. The package call is then updated automatically.

9. **Extracting a superclass:**

   NetBeans gives developers an option to create another level of hierarchy to make things clearer. Design patterns of structural type can also be introduced in the code with the help of this feature.

10. **Encapsulate Fields:**

    When developing object-oriented code, it is almost a standard to give the fields (variables) private access, and use public methods to change their values or retrieve them- mutator (setter) and accessor (getter), respectively. Encapsulate Fields refactoring allows the automation of this process; it generates getter and setter methods for the desired fields, enabling the changing of the access modifier for those fields and the accessor methods.

11. **Change Method Parameters:**

    Change Method Parameters allows you to safely change everything in a method header- access modifier and arguments- and notifies you if this change is going to affect your source file.

12. **Introduce:**

    NetBeans allows us to introduce multiple things like variables, constants, etc. without breaking or crashing the code. Some of these refactoring are:

    1. **Introduce Variable:**

       It supports introducing the variables to a statement that keeps on repeating or is redundant in nature.

    2. **Introduce Constant:**

       The Extract Constant refactoring makes your source code easier to read and maintain. It also helps you avoid using hardcoded constants without any explanations about their values or purpose.

3. **Introduce field:**

    This refactoring allows you to create a new field based on a selected expression, initialize it with the expression or from the constructor, and replace occurrences of the expression in the current type with references to the newly introduced field.

4. **Introduce Method:**

    Sometimes in your code, you notice that there are certain sections that contain similar blocks of code. Introduce Method refactoring can extract these code fragments into a separate method that can be called anywhere. This makes the code more readable and easier to maintain. Assume that the fruits in the inventory are submitted into a database, but before doing so, their names must be written in upper case letters. The for loop inside the insertIntoDB method does that. You find out later that you need to do this operation again somewhere else in the code. Instead of writing it again, you can simply extract it into a separate method.

5. **Introduce Parameter:**

    This refactoring allows you to move an expression from a method implementation to its callers by adding a new parameter. All occurrences of the expression are replaced with the new parameter

THE END