**Scenario**

Rick Cj is the NTBank manager who has come forward to the software engineering requesting assistance with their bank system. The housing markets have dropped dramatically and as a result, several users are purchasing property. These users are requesting for bank loan however, before the ban can lend to the clients, we need to know several information regarding the users to identify if they are eligible to get a bank loan eg(income).

**Motivation**

Conditional logic is often used to decide which variation of an algorithm to use, in this case it’s the conditions that the client must meet before they can receive a loan. The Simplifying Conditional Expressions if not the system will have major problems with functionality, maintenance, performance and more.

**Solution**

In the system where several variants of calculation executed in the system and controlled by conditional logic. The fix is to create strategy for each of the variant and make the method delegate each of their calculation from a strategy instance.

**Benefits**

By removing conditional logic, we can clarify the algorithm

Easy to maintain and manage

This also simplifies the involve class by moving in to an hierarchy

Algorithm manipulation is made much easier (swapping algorithm to other run time)

**Cons**

Complication with how algorithms receive data from context class

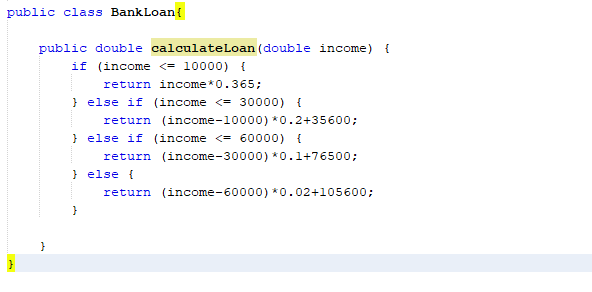
Code and interface is sightly complicated

**Mechanics**

The approach here is to create a strategy naming the strategy after the behaviour performed by the calculation method. Apply the move method to move the calculation to the strategy and since the strategy requires the data in order for the calculation, we may need to pass it a parameter.

**UML**

**Code before**

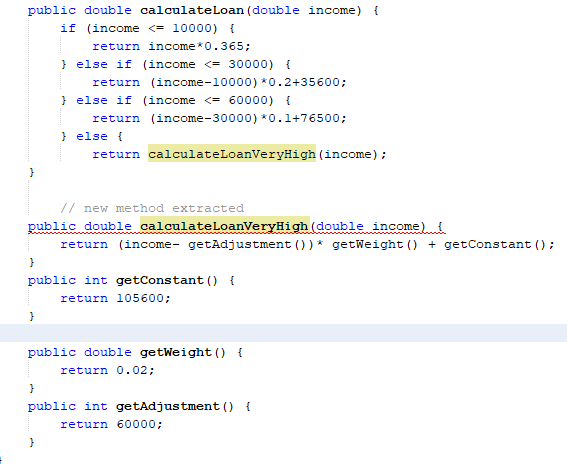


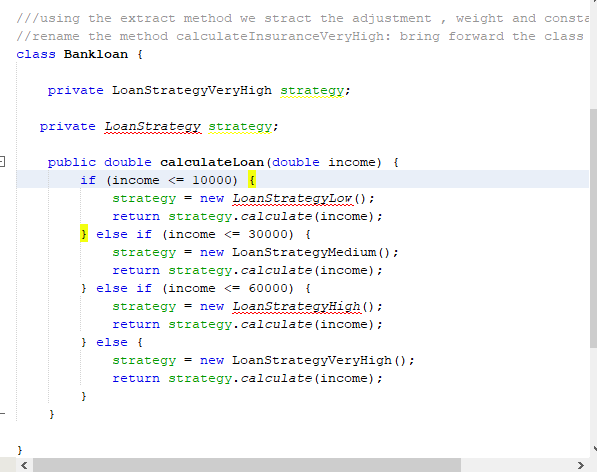
From the original data we can see that the algorithm involves

(income - adjustment) \* weight + constant

**Code after**

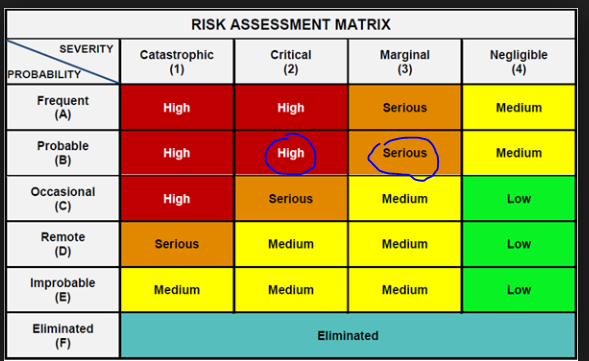
**///using the extract method we extract the adjustment , weight and constant**





**Risk**

The refactoring risk in actually low however, this is a bank system that involved finance and financial system are always at risk because it involved many. If the system functionality was inaccurate then the bank could lose many and client make be inherent high interest.



Testing

The use of unit testing can be used as per attachment below to make sure that system in functional before and after.

