Refactoring: Extract a local variable

Sometimes, we notice that an expression is repeated several times, and in order to simplify the program and make it more efficient, we want to introduce a local variable to contain the value of the expression and evaluate the expression only once. Eclipse has a refactoring tool that makes this easy to do.

In nonsense method \texttt{m} to the right, the expression \texttt{b*c} appears three times. So, we:

1. Select or highlight one of them.
2. Choose menu item \textit{Refactor -> Extract local variable}. This causes this window to pop up:

![Extract Local Variable Window]

When we hit button \textit{OK}, method \texttt{m} is changed, as shown on the right. The assignment of \texttt{b*c} to \texttt{lc} was inserted and all occurrences of \texttt{b*c} were replaced by \texttt{lc}.

Be careful!

Procedure \texttt{m}, to the right, is the same as the one above except that the statement shown by the arrow has been added. Because the statement before it changes the value of \texttt{b}, the expressions \texttt{b*c} in these two statements have different values. When using the \textit{Extract local variable} tool as above, all the occurrences of \texttt{b*c}, including the one marked by the arrow, will be replaced by variable \texttt{lc}. So, the final values of \texttt{b} before and after refactoring are different.

The moral of the story is that the result of refactoring a method in any way must be checked carefully to make sure the intended effect was achieved. Hopefully, in a real setting, a Junit testing procedure was written to test the method, and retesting will show any errors.

©David Gries, 2018