Refactoring

The word *refactor* is not in the online unabridged Merriam-Webster Unabridged Dictionary (as of Dec. 2017), but *refactoring* has been an important concept in programming since the 1980s and 1990s.

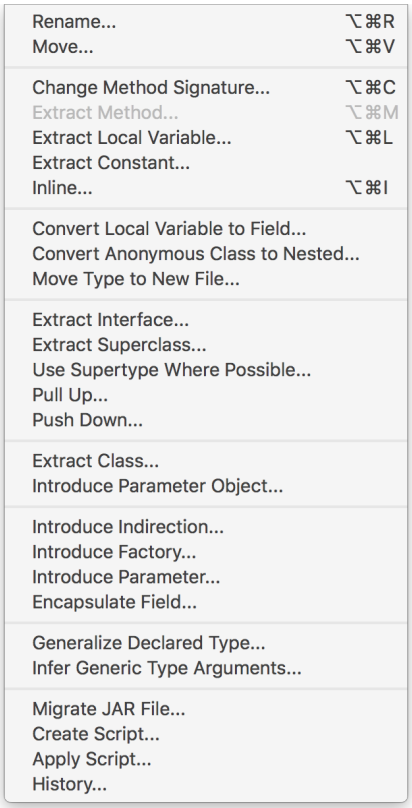
You know what *factoring* means in mathematics: finding the factors of an integer. *Factoring* is often generalized to mean the process of separating an equation, formula, cryptogram, etc. into its component parts (see Dictionary.com). In computer science it is also used as a synonym for *decomposition* —breaking a program or any complex system into parts that are easier to work with. See https://en.wikipedia.org/wiki/Decomposition_(computer_science).

> For programs, factoring refers to the decomposition of a program into parts —functions, procedures, etc. In OO programming, it would include the decomposition into classes and interfaces, methods within classes, and much more.

So, what does *refactoring* mean? Here is a definition: The process of modifying code without changing it behavior/correctness with an eye to improving it in some way, e.g. make it more readable, reduce complexity, make it easier to maintain, shorten it.

Eclipse refactoring tools

Eclipse provides several refactoring tools. They are listed in the window to the right, which appears when you use menu item *Refactor*. Some of the items may be greyed out, depending on where the cursor is in the program.



Here's another way to get a menu to help in refactoring; we use it more often than the main *Refactor* menu item. With the cursor positioned somewhere in a program, right-click your mouse to open a contextual menu; it will have *Refactor* in the middle of it. Slide the mouse to menu item *Refactor*, and a list of possible refactoring methods will appear.

We look at one extremely useful refactoring tool provided by Eclipse.

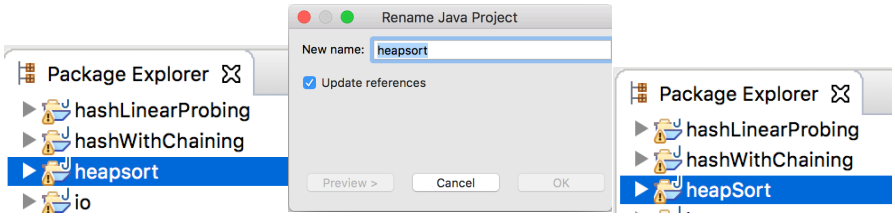
Renaming an identifier

Use menu item *Refactor -> Rename* to rename any identifier —a local variable, parameter, field, method, class, interface, even the name of the project. The neat point about this tool is that Eclipse will change the identifier wherever it is used in the project! Just:

(1) Place the cursor on the identifier,
(2) Select menu item *Refactor -> Rename*,
(3) Edit the identifier, and
(4) Hit the return/enter key.

We give three examples of renaming.

The image on the left below is part of the Package Explorer pane, with project heapsort selected. We select menu item *Refactor -> Rename*. The *Rename Java Project* window appears. Note that the default is to update references, so wherever this project is somehow referenced, the name *heapsort* in the other will be changed. We change the name to *heapSort*, and click button *OK*, resulting in the image to the right. Note that the name of the file for the project on your hard drive has been changed.



Refactoring

Changing a method name

To the right is procedure selection sort. It calls function *minimum*, whose header is at the bottom of the text box.

We decide to shorten the name of function *minimum* to *min*. To do this,

(1) Place the cursor within identifier *minimum* (or select the identifier). Note that it doesn't matter whether the cursor is placed on the call within *selectionSort* or on the name in the header of *minimum*.

(2) Select menu item *Refactor -> Rename*. A small rectangle encloses the identifier, and you can now edit it. We change it to *min*.

(3) Press key *Return/Enter*.

The name of the method has been changed, and all calls of the method have been changed, too, even calls that are in other classes. Neat!

```
/** Sort b */
public static void selectionSort(int[] b) {
   int j= 0;
   // inv P: b[0..j-1] is sorted and
   //        b[0..j-1] <= b[j..]}
   while (j != b.length) {
      int p= minimum(b, j, b.length-1);
      // {b[p] is min of b[j..b.length-1]}
      // Swap b[j] and b[p]
      int t= b[j]; b[j]= b[p]; b[p]= t;

      j= j+1;
   }
}

/** Return the position of minimum of b[h..k]
    Precondition: h < k. */
public static int minimum(int[] b, int h, int k)
```

Changing a variable name

Changing a variable name —parameter, local variable, field, or static variable— is similar. Select *any* occurrence of it, select menu item *Refactor -> Rename*, edit the name, and press key *Return/Enter*.