

Part 7: Extracting Methods

Exercise Instructions

This part of the challenge is going to get you using a feature that pro's use all the time: and that's refactoring. In this exercise you're going to tidy up the code by using extracting methods (i.e. pulling code out into their own methods), and keep code tight by removing an unused variable.

Step 1: Update the Code

Extrct methods which will print the input parameters and print the results.

- Push the code which prints out the input variables into a method called `printInputs` .
- Push the code which does the actual calculation into a method called `calculateRepaymentAmount`
- Push the code which formats and prints the result out into a method called `printResult` .

The above actions are known as the Extract Method refactoring.

Note: To do the Extract Method refactoring, highlight just the lines to be pushed into a new method (by either clicking and dragging the text caret, or pressing SHIFT and using the arrow keys to do this), and then right-click on them and select **Refactor > Extract > Method...** and OK'ing through the prompts, entering the new method name. A new method should now appear which does just those lines, and a call to that method appears where those lines used to be.

- Inline the call to `calculateRepaymentAmount` in to the `printResult` method to remove the unused variable. (If you remember, "inlining" means to just call a method inside another method so the result can be used immediately, without having to have an intermediate variable to hold the result - this is used when nobody else needs the result besides that method that the call is inlined to).

Step 3: Run the program

- Run the program and check it all works as before. (It should work identically since refactoring is just changing the *structure* of the code without changing the functionality - i.e. what it does).