

Extract Method

When the *Extract Method* refactoring is invoked, IntelliJ IDEA analyses the selected block of code and detects variables that are the input for the selected code fragment and the variables that are output for it.

If there is exactly one output variable, it is used as a return value for the extracted method. In case there are multiple output variables, the *Extract Method* refactoring may not be applied, and the error message appears.

There are several workarounds to allow *Extract Method* work in this case. For example, you may introduce a special data-class that contains all output values.

- [Example](#)
- [Extracting a method](#)

The Extract Method refactoring has the following limitations:

- Refactoring does not work with multiple output values in automatic mode. You have to change your code before applying the refactoring.
- Refactoring does not work for a code fragment which conditionally returns from the containing method and is not placed at the end of it.

Example

| Before | After |
|---|--|
| <pre>public void method() { int a=1; int b=2; int c=a+b; int d=a+c; }</pre> | <pre>public void method() { int a=1; int b=2; int c=add(a,b); int d=add(a,c); } ... private int add(int a, int b) { return a+b; }</pre> |
| <pre>public ArrayList method() { String[] strings = {"a","b","c"}; ArrayList list = new ArrayList(); for (int i=0; i < strings.length; i++) {list.add(strings[i]);} return list; }</pre> | <pre>public ArrayList method() { String[] strings = {"a","ab","abc"}; ArrayList list=add(strings); return list; } private ArrayList add(String[] strings) { ArrayList list = new ArrayList(); for (int i=0; i < strings.length; i++) {list.add(strings[i]);} return list; }</pre> |

To extract a method

1. In the editor, select a block of code to be transformed into a method or a function.

The code fragment to form the method does not necessarily have to be a set of statements. It may also be an expression used somewhere in the code.

2. On the main menu or on the context menu of the selection, choose **Refactor | Extract | Method** or press **Ctrl+Alt+M**.
3. In the **Extract Method** dialog box that opens, specify the name of the new method.
4. To create a static method, select the **Declare Static** check box.
5. In the **Parameters** area, do the following:
 - Specify the variables to be passed as method parameters, by selecting/clearing the corresponding check boxes; if a parameter is disabled, a local variable of the corresponding type, with the initial value ... will be created in the extracted method, so that you will have to enter the initializer with an appropriate value manually.
 - Rename the desired parameters, by double-clicking the corresponding parameter lines and entering new names.
6. In the **Visibility** area define the method's visibility scope.
7. Check the result in the **Signature Preview** pane and click **OK** to create the method. The selected code fragment will be replaced with a method call. Additionally, IntelliJ IDEA will propose to replace any similar code fragments found within the current class.

You can extract methods from the repetitive code fragments, which IntelliJ IDEA finds in course of the [duplicates analysis](#). The encountered duplicates display in the [Duplicates tool window](#), where you can try to replace them with method calls.

See Also

Reference:

- [Extract Method Dialog](#)

External Links:

- [Replace conditional logic with strategy pattern](#) 

Web Resources:

- [Developer Community](#) 