

Extract Method Object

The **Extract Method Object** refactoring moves method into a new class, converting all the local variables to its fields, allowing you to decompose the method into other methods on the same object. It is an alternative to the [Extract Method](#), and can be used when you have multiple return values in an extracted method.

Example

Before	After
<pre>class Account { int gamma (int val1, ...) { //some computations return c-2*a; } }</pre>	<pre>class Account { int gamma (int val1, ...) { Calculations calculations = new Calculations(val1, ...); int c = calculations.getC(); int a = calculations.getA(); return c-2*a; } private class Calculations { private int val1; ... private int a; private int c; public Calculations(int val1, ...) { this.val1 = val1; ... } public int getA() {return a;} public int getC() {return c;} public Calculations invoke() { ...//computations return this; } } }</pre>

To extract a method object

1. In the editor, select the method code block to be extracted into the object.
2. On the main menu, or from the context menu of the selection, choose **Refactor | Extract | Method Object**.
3. Select whether you want to create inner class, or anonymous class.

You cannot extract the method object into an anonymous class, if the selected method code block contains local variables that should be accessed individually somewhere else. In this case method object can be extracted into inner class, that will contain needed getters.

4. If you want to create an inner class, you need to specify the name for the class and the visibility scope. You can also make the class static, if needed.
5. If you want to create an anonymous class, you should specify method's name.
6. In the **Parameters** area select the variables that will be used as a parameters.
7. Review **Signature Preview** and click **OK**.

See Also

Reference:

- [Extract Method Object Dialog](#)

Web Resources:

- [Developer Community](#) 