

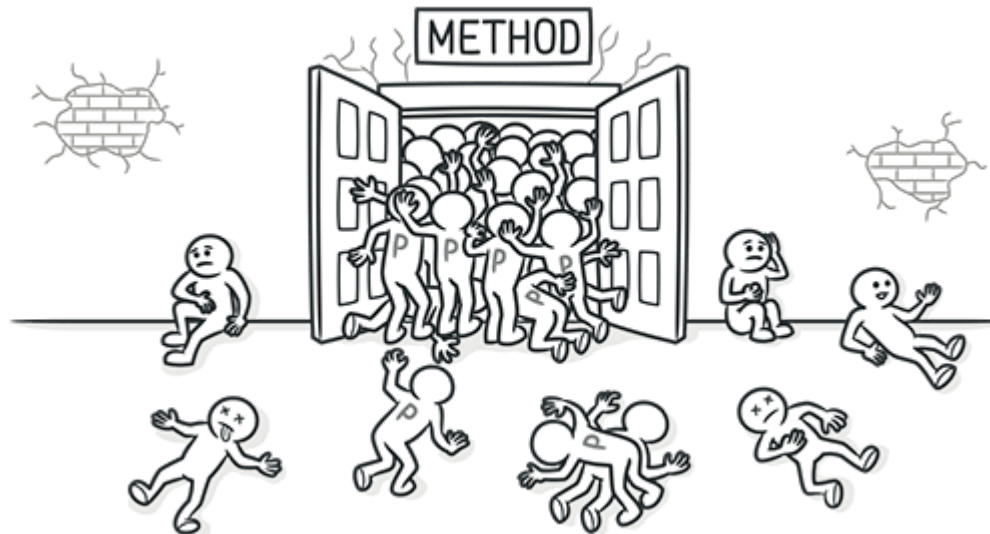


[Home](#) / [Refactoring](#) / [Code Smells](#) / [Bloaters](#)

Long Parameter List

Signs and Symptoms

More than three or four parameters for a method.



Reasons for the Problem

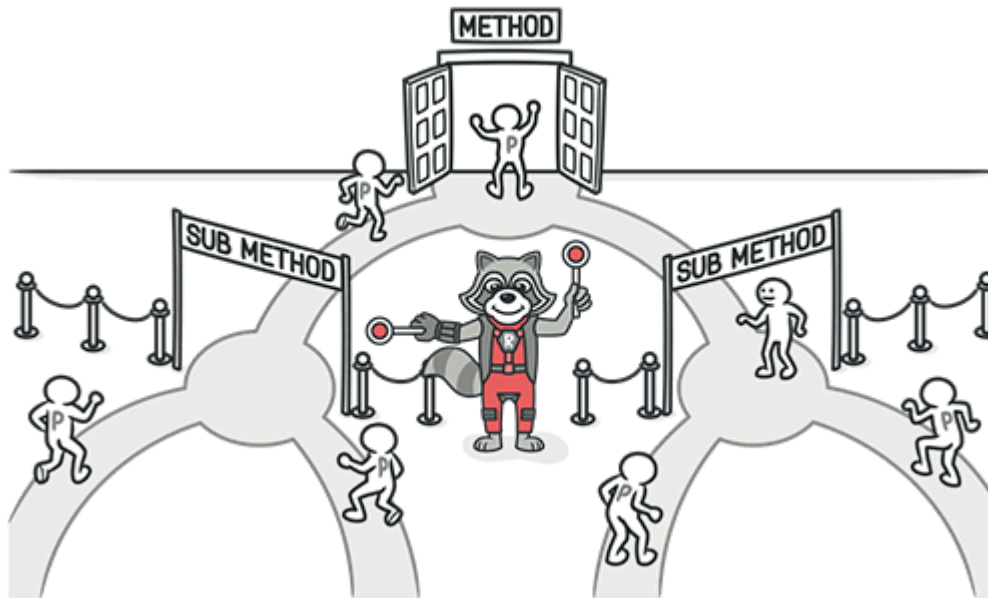
A long list of parameters might happen after several types of algorithms are merged in a single method. A long list may have been created to control which algorithm will be run and how.

Long parameter lists may also be the byproduct of efforts to make classes more independent of each other. For example, the code for creating specific objects needed in a method was moved from the method to the code for calling the method, but the created objects are passed to the method as parameters. Thus the original class no longer knows about the relationships between objects, and dependency has decreased. But if several of these objects are created, each of them will require its own parameter, which means a longer parameter list.

It's hard to understand such lists, which become contradictory and hard to use as they grow longer. Instead of a long list of parameters, a method can use the data of its own object. If the current object doesn't contain all necessary data, another object (which will get the necessary data) can be passed as a method parameter.

Treatment

- Check what values are passed to parameters. If some of the arguments are just results of method calls of another object, use **Replace Parameter with Method Call**. This object can be placed in the field of its own class or passed as a method parameter.
- Instead of passing a group of data received from another object as parameters, pass the object itself to the method, by using **Preserve Whole Object**.
- If there are several unrelated data elements, sometimes you can merge them into a single parameter object via **Introduce Parameter Object**.



Payoff

- More readable, shorter code.
- Refactoring may reveal previously unnoticed duplicate code.

When to Ignore

Don't get rid of parameters if doing so would cause unwanted dependency between classes.