# Avoiding overuse of method overloading [closed]

Asked 13 years, 7 months ago  Modified 5 years, 7 months ago  Viewed 2k times

3

**Closed**. This question needs details or clarity. It is not currently accepting answers.

**Want to improve this question?** Add details and clarify the problem by editing this post.

Closed 9 years ago.

Improve this question

I've come across a Java utility class that has four methods for processing data. All methods process and then write the parameters to the same file, and each method accepts four different sets of inputs:

```
util::process(String data1)
util::process(String data1, Object1 data2)
util::process(String data1, String data3)
util::process(String data1, Object2 data4)
```

I'm seeing what feels like a code smell because every time a new combination of data is introduced, a new method is also added. The method overloading feels like it is masking a design or object construction issue.

Is it possible to refactor this and avoid method overloading? How?

java design-patterns refactoring overloading

Share

Improve this question

Follow

edited Apr 30, 2019 at 2:28

Bhargav Rao

52k ● 29 ● 126 ● 141

asked May 28, 2011 at 18:32

TERACytE

7,833 ● 15 ● 79 ● 112

give us type information for data1-4. – Charlie Martin May 28, 2011 at 18:36

2 What language are we talking about here? Different languages have different constructions for dealing with this type of situation. – Trott May 28, 2011 at 18:38

Is each combination handled by the same code? Does the code vary for how many arguments there are? I see one function takes one argument and the others take two. – erisco May 28, 2011 at 18:43

Updated the question to include the language (i.e. Java) and the parameter types. – TERACytE May 28, 2011 at 18:44

Erisco, I'm not clear on what you are asking. The methods in this utility class are called by many other class in the project. Depending on the context, those classes will call the appropriate method in the utility class. Data 1 thru 4 may be available to those classes, but only two of them need to be processed by the utility class. – TERACytE May 28, 2011 at 18:50

## 3 Answers

Sorted by: Highest score (default)

3

This is an interesting question because there are not enough details to really say definitively. I do believe there is an odor coming off this code, and it's the stench of procedural trying to look like object-oriented programming. It's funny that the method is entitled 'process' and the types are either strings or generic object like names. What is doing the processing? Surely not a controller object, I am guessing. At some point, you want to have objects that have responsibilities. So let's assume that the base of this was the concept of an Application (e.g. an application for a job). You might want to pull the method process up into that type, and then use inheritance to augment functionality by introducing different Application types.

Another option would be to introduce a Chain of Responsibility pattern. The problem I see with that is you seem to be saying that there are only 2 links in the potential chain. The advantage of the chain approach is that the handlers don't have to know anything about each other.

A third option would be to make the successors implement an interface. This would be something like the Command Pattern, then the actual 'processing' is really just a trigger and each thing is responsible for its own behavior.

Share Improve this answer Follow

answered May 28, 2011 at 22:01

Rob
11.7k ● 7 ● 42 ● 59

2

IMHO it is not smelly because of overloading. If these weren't overloads, but different methods, it would be code smell anyway. From what you write, it seems that the code does not follow the open/closed principle. If don't provide more details on what you are trying to accomplish with the code, we cannot suggest any solution.

Share Improve this answer Follow

answered May 28, 2011 at 18:48

Steves
3,214 ● 22 ● 25

I've added more details in the comments above. Do you need more? If so, what would you need? – TERACytE May 28, 2011 at 18:55

1

I can think of a couple of possibilities that explain this kind of overloading quite rationally:

1. Only one of these functions actually does the work, and the others are merely presented for convenience and internally call the 'worker' method. It could even be that none of these methods do any work and they all call an internal (private) method that does the heavy lifting. I don't see any problem with that pattern and use it myself, there is no replication.
2. It could be that what the overloaded methods are trying to achieve is just not possible generically due to syntactical limitations or design decisions.

As has been previously mentioned, you don't provide information on what these methods are trying to achieve. You may well be able to refactor and eliminate all but one variation of those methods but before I can recommend whether or not you should do it or how you would do it you need to specify what the purposes of the methods are and how they differ from each-other. The name 'process' doesn't tell us much about them.

Share Improve this answer Follow

answered May 28, 2011 at 22:30

Gyan

12.4k ● 2 ● 25 ● 26