



[Home](#) / [Refactoring](#) / [Code Smells](#) / [Object-Orientation Abusers](#)

# Alternative Classes with Different Interfaces

## Signs and Symptoms

Two classes perform identical functions but have different method names.



## Reasons for the Problem

The programmer who created one of the classes probably didn't know that a functionally equivalent class already existed.

## Treatment

Try to put the interface of classes in terms of a common denominator:

- **Rename Methods** to make them identical in all alternative classes.
- **Move Method**, **Add Parameter** and **Parameterize Method** to make the signature and implementation of methods the same.
- If only part of the functionality of the classes is duplicated, try using **Extract Superclass**. In this case, the existing classes will become subclasses.
- After you have determined which treatment method to use and implemented it, you may be able to delete one of the classes.

## Payoff

- You get rid of unnecessary duplicated code, making the resulting code less bulky.
- Code becomes more readable and understandable (you no longer have to guess the reason for creation of a second class performing the exact same functions as the first one).



## When to Ignore

Sometimes merging classes is impossible or so difficult as to be pointless. One example is when the alternative classes are in different libraries that each have their own version of the class.

