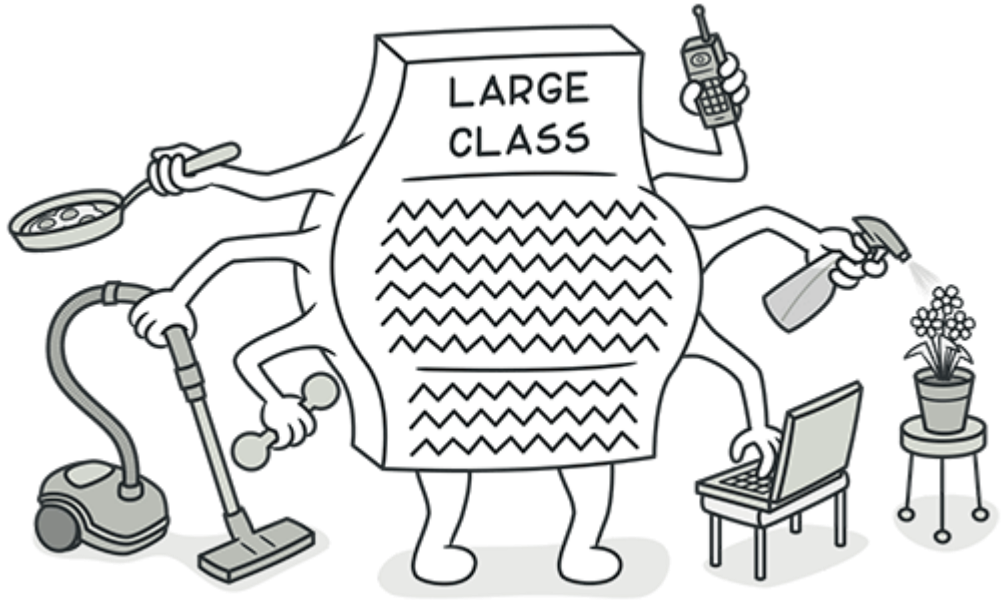
Code Smells

THE COMMON CODE SMELLS
THAT I ~~USED TO~~ MAKE

Bloaters

- Large pieces of code accumulated over time



Large classes + Long Methods

How to observe?

- Class too long (50+ lines)
- Method too long (15+ lines)

Solution

- Extract methods into smaller classes based on responsibility
- Remove duplication
- Extract parts into smaller methods
- Decompose conditionals

Primitive Obsession

How to observe?

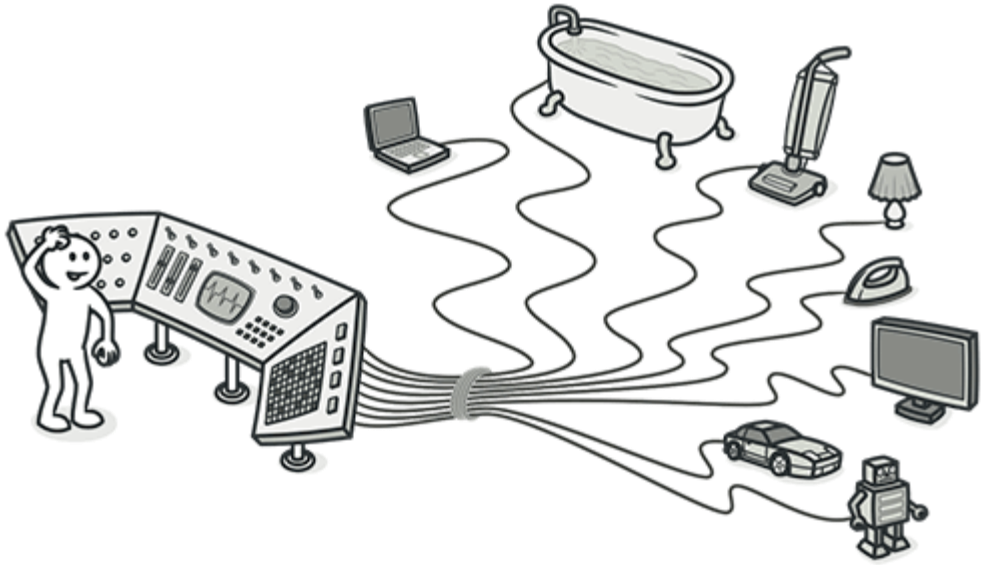
- Use of many primitive values instead of objects

Solution

- Move the primitives into their own class
- Keep the associated behavior separated in that class

Object-Orientation Abusers

- Incorrect application of object-oriented programming



Switch Statements

How to observe?

- Long Switch operator
- Multiple If statements

Solution

- Create interface/abstract class with subclasses that match each type/property that appears in the switch
- Instead of conditional, use polymorphism to call the method from the right subclass

Temporary Field

How to observe?

- Value used only in some circumstances
- Otherwise always null/unused

Solution

- Variable and code using extracted in separate class

Refused Bequest

How to observe?

- Subclass uses only some of the inherited methods

Solution

- Split up interface/parent class (interface segregation)
- Get rid of inheritance, create parent class object inside subclass and use needed methods

Alternative Classes with Different Interfaces

How to observe?

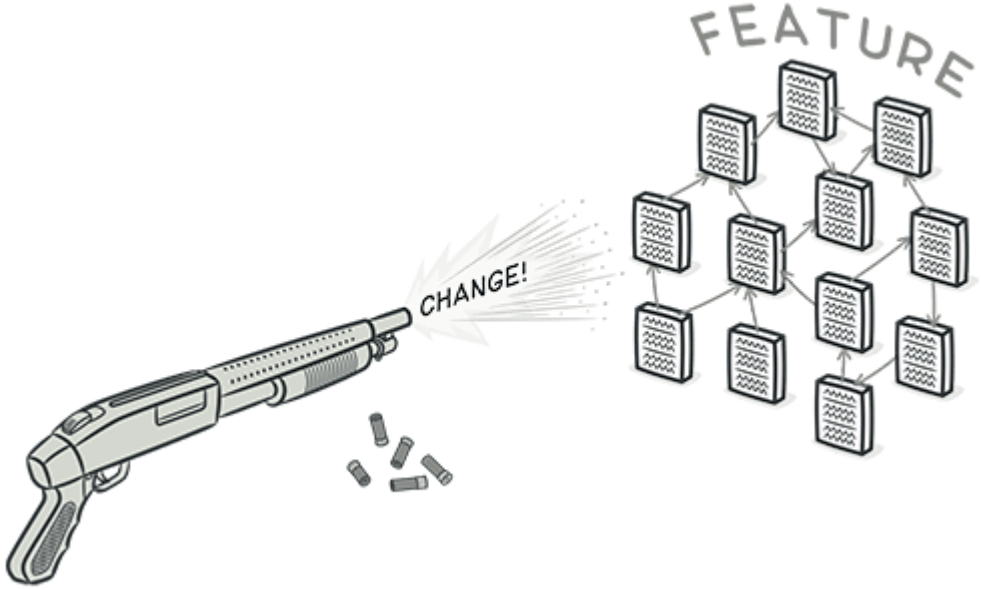
- 2 classes have identical functionality, but different names and methods

Solution

- Know code base
- Get rid of duplication, move methods into one class, get rid of other
- If only partially identical, extract common methods into superclass, a make the existing classes its subclasses

Change Preventers

- If you make one change, you must make many changes in other places too



Divergent Change vs Shotgun Surgery

How to observe?

- Divergent change: Must change many unrelated methods when you make a change to a class

VS

- Shotgun surgery: If you change one thing, then you are required to make changes in other classes too

Solution

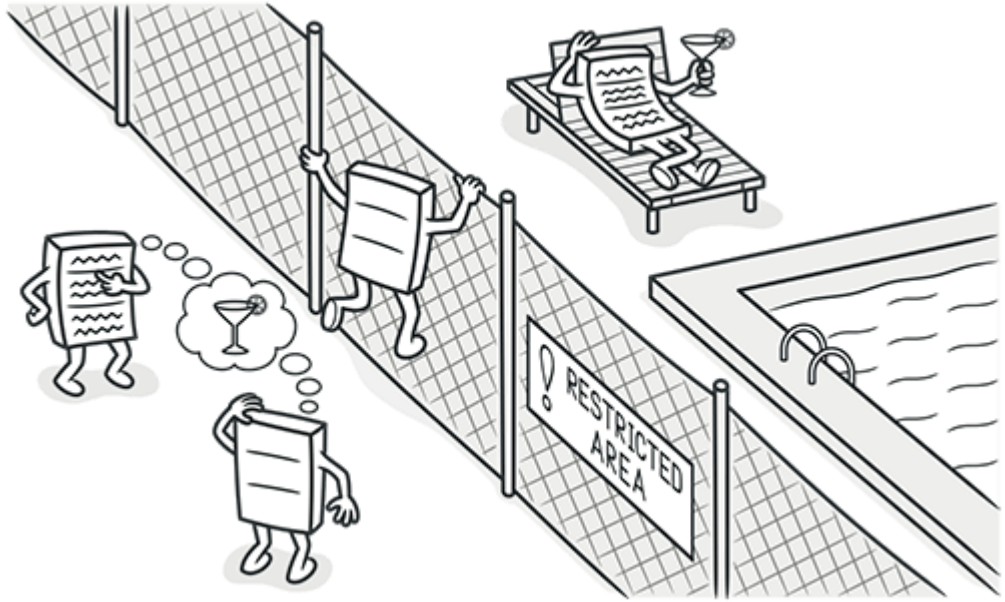
- Divergent change: Split up unrelated parts into different classes, combine related parts with inheritance

VS

- Shotgun surgery: Move responsibility to the same class

Couplers

- Code that creates excessive coupling between classes



Feature Envy + Inappropiate Intimacy

How to observe?

- Feature Envy: Method accesses the public fields and methods of another class more than its own

VS

- Inappropiate Intimacy: Method accesses the internal workings of the class

Solution

- Feature Envy: Method better suited to other class, should be moved

VS

- Inappropiate Intimacy: Make the methods and fields private, force our class to work only with accessible methods

What did I learn?

- Lots of different smells
- Many things to remember
- Many are obvious but a few are difficult to recognize

Sources

- Alcor Academy
- https://en.wikipedia.org/wiki/Code_smell
- https://sourcemaking.com/refactoring/smells
- https://www.shutterstock.com/ (photos)

Thank you for your attention !

Alexandru Chevul

LINKEDIN: www.linkedin.com/in/alexandru-chevul-458697185

EMAIL: alexandru.chevul@betterask.erni