

# Code smells

Elisabeth Forland

Developer at Bouvet

[elisabeth.forland@bouvet.no](mailto:elisabeth.forland@bouvet.no)



# Agenda

What is a code smell?

Why bother fixing them?

Example code smells

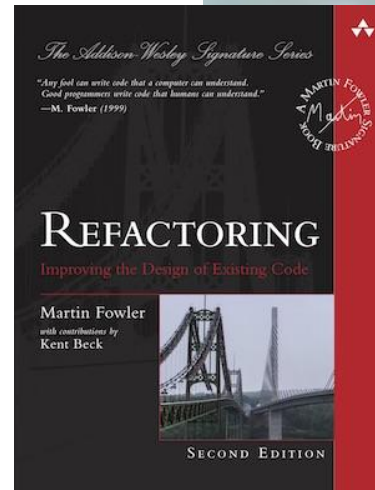
How to deal with some bad smells

When someone asks "What 's that smell?"



# What is a code smell?

- Surface indication of deeper problem in your system
- Sniffable – easy to spot
- Don't *always* indicate a problem, but gives you clues to where the *might* be design issues



# Why bother fixing them?

- Rigid
- Fragile
- Not reusable
- Complex design
- Environment



# Types of code smells

- Bloaters
- Object-Orientation Abusers
- Change Preventers
- Couplers

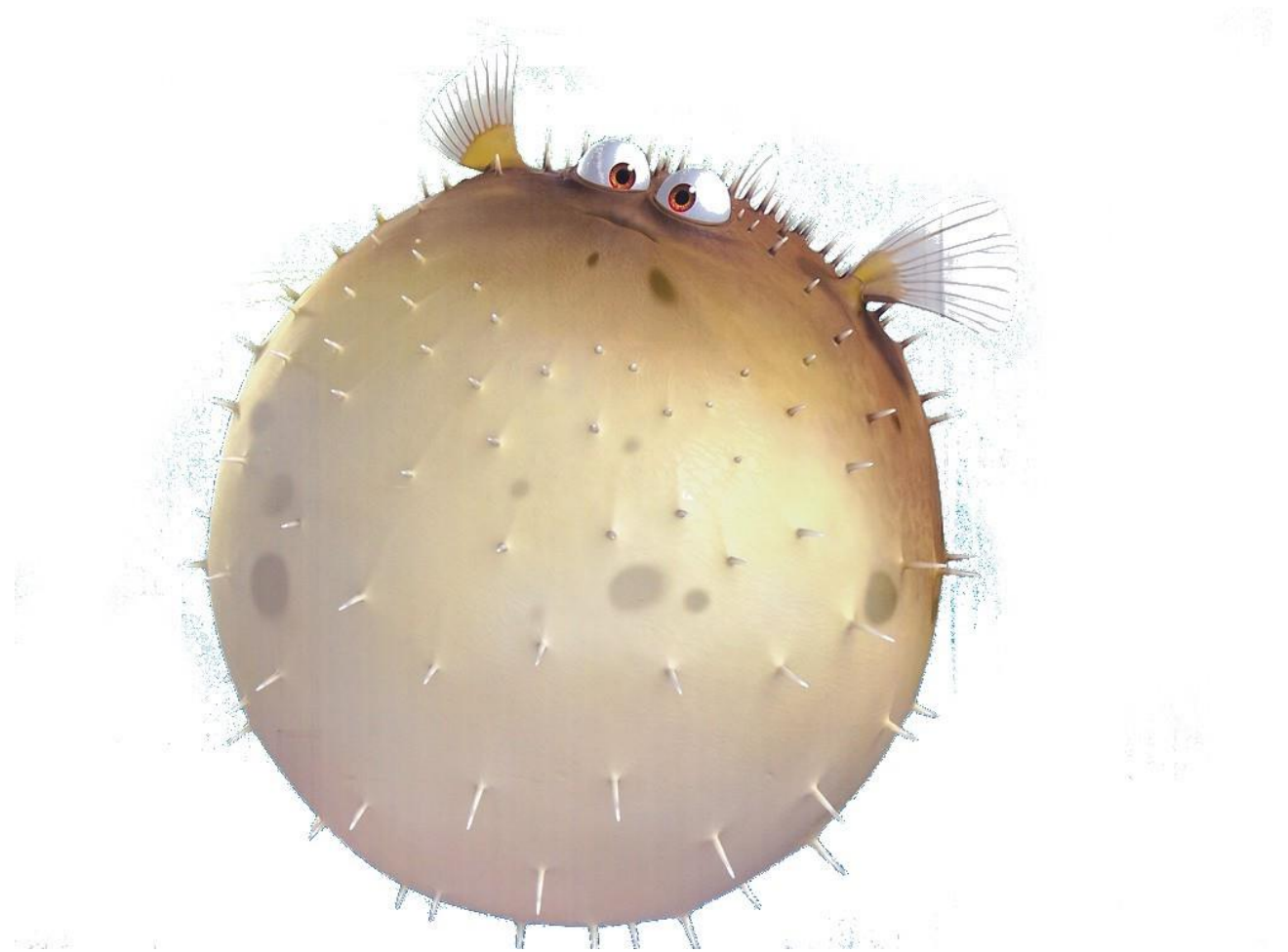




# Bloaters

Accumulate over time

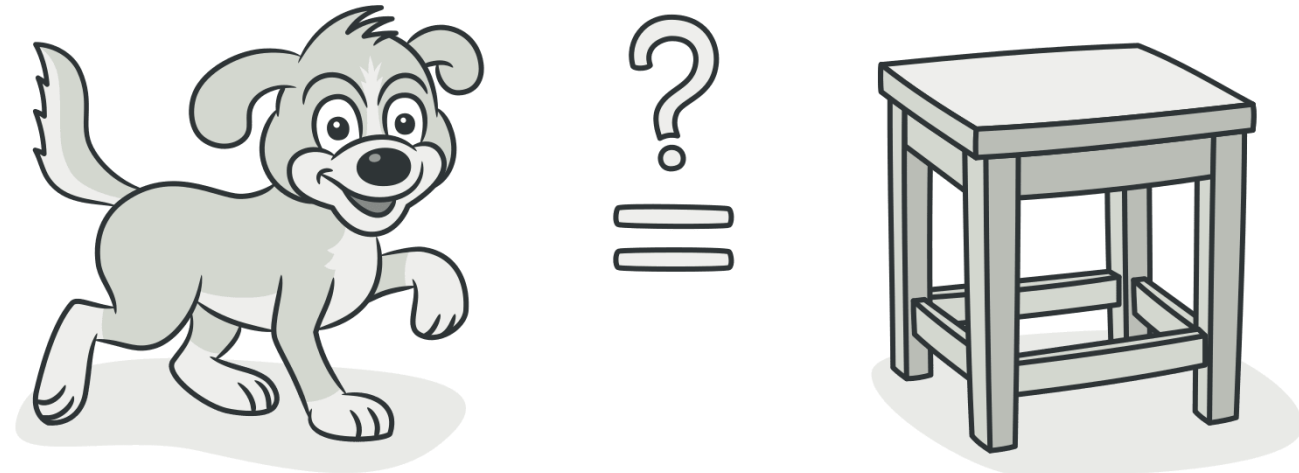
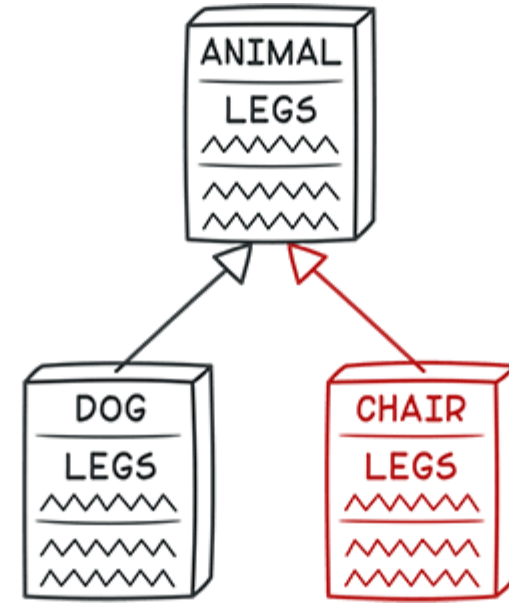
- Long methods
- Large classes
- Primitive obsession
- Long parameter list
- Data clumps



# Object-orientation Abusers

Incomplete or incorrect application of object-oriented programming principles

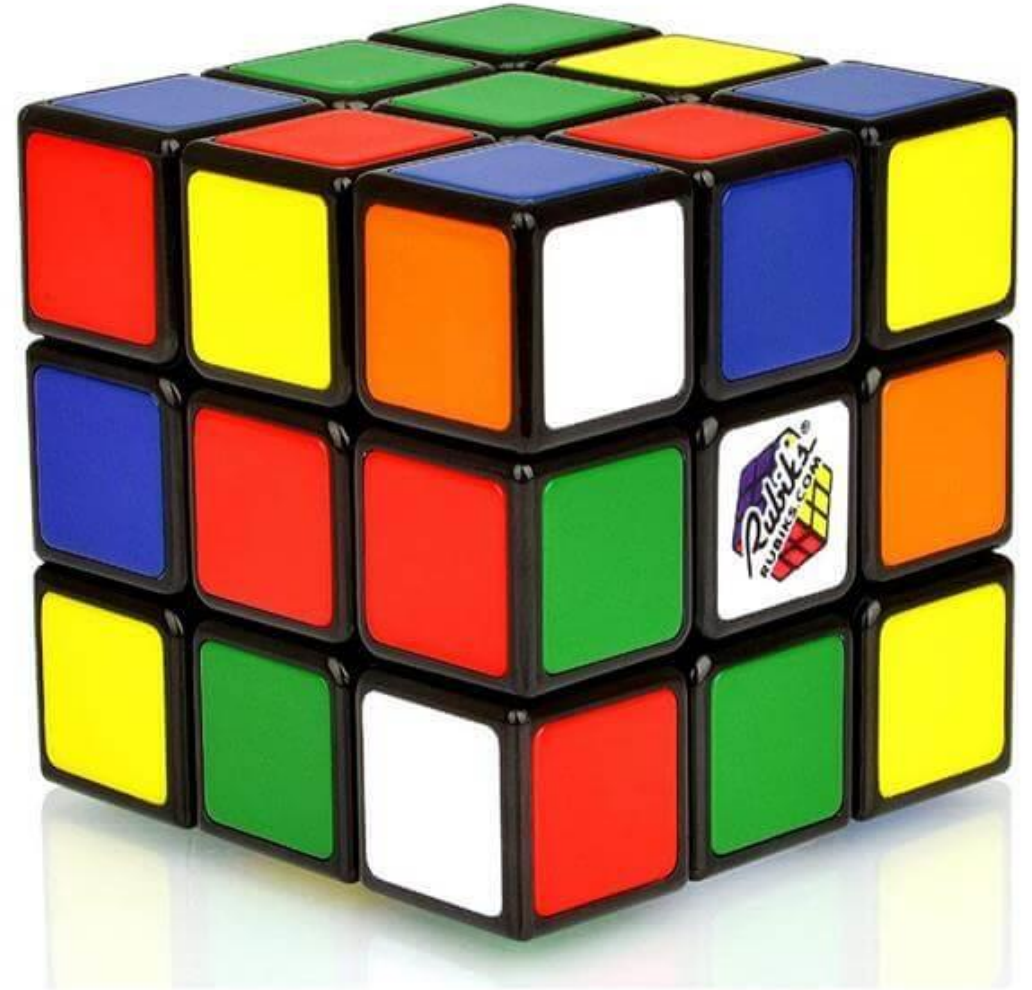
- Switch statements
- Temporary field
- Refused bequest
- Alternative classes with different interfaces



# Change Preventers

Change one thing leads to changing a lot of other things

- Divergent change
- Shotgun surgery
- Parallel inheritance hierarchies





# Dispensables

- Unneeded clutter
- Makes code cleaner when removed
- Comments
- Duplicate code
- Lazy class
- Data class
- Dead code
- Speculative generality



Source: imgflip.com

# Couplers

Excessive coupling or excessive delegation

- Feature envy
- Inappropriate intimacy
- Message chains
- Middle man



`Dog.Body.Tail.Wag()`

`Dog.ExpressHappiness()`

# Object Calisthenics and Code smells

Object calisthenics	→	Code smells
Only one level of indentation per method		Long Method
Don't use the ELSE keyword		Long Method / Duplicated Code
Wrap all primitives and strings		Primitive Obsession
First class collections		Divergent Change / Large Class
One dot per line		Message Chains
Keep all entities small		Large Class / Long Method / Long Parameter List
No classes with more than two instance variables		Large Class
No getters / setters / properties		Feature Envy
All classes must have state, no static methods, no utility classes		Lazy Class / Middle man / Feature envy



Thank you for  
your time!

Questions?



# References

- <https://sourcemaking.com/refactoring/smells>
- <https://martinfowler.com/bliki/CodeSmell.html>
- <https://deepsource.io/glossary/code-smell/>
- Alcor Academy – Running: <https://alcor.academy/training-programme>

Contact details:  
Elisabeth Forland  
Elisabeth.Forland@bouvet.no