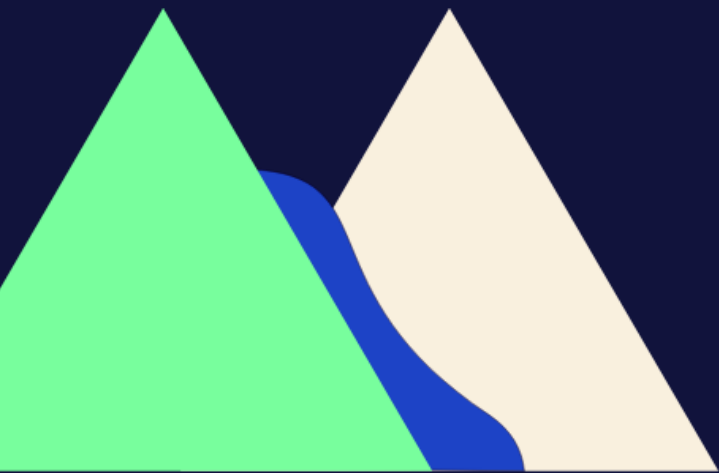


Refactoring

Code that will get you a Nobel prize in litterateur

Anders Lilleheim



Focus on Readability

Refactor readability first
-> then design

Object Calisthenics rules are your friends. Listen to your friends.

Do not break it

Refactoring should not break anything

- > stay in the green

Do not change public api

- > do parallel change

- > expand, migrate, contract

Code smells

- Bloaters
- Object-orientation abusers
- Change preventers
- Dispensables
- Couplers

Code smells

Bloaters

- Long method
- Large class
- Primitive obsession
- Long parameter list
- Data clumps





Code smells

Object-orientation abusers

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



Code smells

Change preventers

- Divergent change
- Shotgun surgery
- Parallel inheritance hierarchies

Code smells

Dispensables

- Comments
- Duplicate code
- Lazy class
- Data class
- Dead code
- Speculative generality

Code smells

Couplers

- Feature envy
- Inappropriate intimacy
- Message chains
- Middle man
- Incomplete library class



What can we achieve

- Maximize cohesion
 - Minimize coupling
 - Optimize connascence
-
- Modular code that scales and can be changed without the whole system breaking

Thank you

Anders Lilleheim

Email: Anders.lilleheim@bouvet.no

For more information check out:

<https://alcor.academy/>

