



🏠 / Refactoring

Code Smells

- What? How can code "smell"??
- Well it doesn't have a nose... but it definitely can stink!



Bloaters

Bloaters are code, methods and classes that have increased to such gargantuan proportions that they are hard to work with. Usually these smells do not crop up right away, rather they accumulate over time as the program evolves (and especially when nobody makes an effort to eradicate them).

- Long Method
- Large Class
- Primitive Obsession

- Long Parameter List
- Data Clumps

Object-Orientation Abusers

- Switch Statements
- Temporary Field
- Refused Bequest
- Alternative Classes with Different Interfaces



All these smells are incomplete or incorrect application of object-oriented programming principles.



Change Preventers

These smells mean that if you need to change something in one place in your code, you have to make many changes in other places too. Program development becomes much more complicated and expensive as a result.



Dispensables

A dispensable is something pointless and unneeded whose absence would make the code cleaner, more efficient and easier to understand.

Couplers

All the smells in this group contribute to excessive coupling between classes or show what happens if coupling is

- Divergent Change
- Shotgun Surgery

- Parallel Inheritance Hierarchies

- Comments
- Duplicate Code
- Lazy Class

- Data Class
- Dead Code
- Speculative Generality

- Feature Envy
- Inappropriate Intimacy
- Message Chains
- Middle Man

- Incomplete Library Class



replaced by excessive delegation.

Reading is boring

Aren't you bored of reading so much? Try out our new interactive learning course on refactoring. It has more content and much more fun.



♡ Learn more

READ NEXT

Bloaters



RETURN

Design Patterns
AntiPatterns
Refactoring
UML

My account
Forum
Contact us
About us

© 2007-2019 SourceMaking.com
All rights reserved.

[Terms / Privacy policy](#)