

Refactoring - CS 2340

...

Shishir Bhat

What's the point?

- We will all encounter bad code/design
- Most of us will be not be working on new projects. More likely to be placed on existing product.
- Get rid of technical debt from early in the project

Refactoring

- Changing existing code without changing behavior
- Used to improve design of already working/written code
- Works well with properly-written unit tests

Code Smells

- Something in source code that could indicate a deeper problem.
- Not necessarily something that needs to be removed
- Subjective to the programmer

Common Code Smells

- **Duplicated Code** - When the same exact code appears in more than 1/2 locations
 - Solution: Extract common code into a separate method
 - Keep code DRY (Don't Repeat Yourself)

Common Code Smells

- **Long Method** - When your method is more than 10-15 lines long
 - Solution: Extract out code into a separate method
 - Especially in Scala and other non-verbose languages, really long methods are a huge code smell

Common Code Smells

- **Long Class** - When your class has too many methods and behaviors
 - Solution: Split up class into separate, more-specific classes
 - Remember the Single Responsibility Principle

Common Code Smells

- **Long Parameter List** - When methods take in too many parameters
 - **Ex:** `createWindow(null, 5, 6, 10, null, 540, null)`
 - **Solution:** Create parameter object, named parameters
 - **Ex:** `createWindow(Dimension(5, 6, 10), res=540)`

Common Code Smells

- **null** - Using `null` in your code
 - Solution: Use a more descriptive data type to describe that conditional lack of value (Ex: `Option[T]`)
 - The “billion dollar mistake” - Tony Hoare

Common Code Smells

- **Shotgun Surgery** - When a small change to your application requires changes in many different parts of the source code
 - Solution: Extract out a method/class

Common Code Smells

- **Cyclomatic Complexity** - When the code is too complex and can go down too many branches
 - Solution: Extract out methods and try to eliminate nested branches

Other Code Smells

- `instanceof`
- Comments
- Primitive obsession
- Magic numbers
- Downcasting
- Catching any Exception
- Overuse of inheritance