Software Dev

Clean Object oriented programming

Admin

- Midterm

Caveat

- Everything said in this lecture is true
  - Or so I'm claiming
  - But some languages (python) don't give you the tools to do it right.
  - Principles still hold

Abstraction

- The whole point of Object Oriented Programming/Design:
  - Abstraction through encapsulation
  - Encapsulation: hiding object state through private instance variables
  - Abstraction: hiding the details from users – it just works
  - Example from 94 in clean code
    - Note the jab at java.

What does that gain us?

- What does the abstraction of a point buy us?

What does that gain us?

- What does the abstraction of a point buy us?
  - Now we can use whatever point representation is efficient/convenient without the client code needing to know
  - We can change the implementation without anyone knowing/caring

Accessors and Mutators

- We use accessors and mutators all the time
  - Why?

Accessors and Mutators

- We use accessors and mutators all the time
  - And why are we letting clients access our private data?
  - Accessors maybe
  - Mutators?!?!?!
  - Why do we use accessors even? Is there a better way?
  - (I Know some languages love them)
  - “Richard” gives us another reason not to abuse mutators
  - http://mcfunley.com/from-the-annals-of-dubious-achievement

Data Structures vs Objects

- Book:
  - Objects hide their data and expose operations
  - Data structures expose their data and have no operations.
    - (think an array or a linked list)
  - Now we can try to make object oriented data structures
    - but there is a reason for c++ struct
  - How about a Java 'class' that is really a data structure?
  - Python?

Law of Demeter

- Law of Demeter
  - Ian Holland 1987 Northeastern Univ
  - Don't talk to strangers only talk to friends (in the c++ sense of the word)
  - AKA: principle of least knowledge
  - Book version:
    - A module should not know about the innards of the objects it uses
  -

Law Of Demeter

- Law of Demeter Specifics:
  - A class C with a method M, M can only call:
    - Methods in class C
    - Methods in an object created by M
    - Methods from objects passed as parameters to M
    - Methods from an instance variable of C
    - Do not call methods on objects returned by any of the above
  - Buys us automatically reduced coupling
  - By which I mean?
-

Train Wrecks

- Often referred to as optional part or optional corollary of Law of Demeter: avoid train wrecks
- Any one know what train wrecks are?

Train Wrecks

- Often referred to as optional part or optional corollary of Law of Demeter: avoid train wrecks
- Any one know what train wrecks are?
- this.configuration.getLocation().toString().toUpperCase().equals(otherString)
- Wow!1?
  - Perfectly syntactically correct
  - I’ve done a lesser version myself.
- Law of Demeter violation?

Train Wrecks

- Often referred to as optional part or optional corollary of Law of Demeter: avoid train wrecks
- Any one know what train wrecks are?
- this.configuration.getLocation().toString().toUpperCase().equals(otherString)
- Wow!1?
  - Perfectly syntactically correct
  - I’ve done a lesser version myself.
- Law of Demeter violation?
  - No – but how long did it take you to tell?

Train Wrecks

- Main problem with train wrecks is that it is hard to tell if there is a law of Demeter violation.
- How about this:
  - self.priceLabel.text = self.media.ad.price.value;
  - Is this a Law of Demeter violation?

Train Wrecks

- Main problem with train wrecks is that it is hard to tell if there is a law of Demeter violation.
- How about this:
  - self.priceLabel.text = self.media.ad.price.value;
  - Is this a Law of Demeter violation?
  - No – if you can access the data members directly it is a data structure – not a class
  - So no Law of Demeter Violation.
  - Difficulty in languages like java and frameworks like java beans that demand all data structures use private instance variables and accessors.

Data Transfer Objects

- Data Transfer Objects (DTO)
  - Pure data structures
  - Public instance variables, no methods
  - structs from C++
  - Named tuples from python
-

Next Step

- The Java Bean:
  - 'quasi-encapsulation'
  - Private instance variables
  - Accessors and mutators for all
  - Robert Martin (Uncle Bob) refers to this as:
  - “to make some OO purists feel better but usually provides no other benefit”
  - Were everywhere 5-10 years ago (in heyday of java)
    - Seem to be less widely used these days
    - Lots of legacy code to support.

Active Records

- Refers to a 'special type of DTO
- DTO with 'navigation methods' like save and find
- Designed by Martin Fowler
- Don't add other methods
  - Like business rules.
- These days almost synonymous with ruby on rails
  - Which wasn’t a thing when Clean Code was written.

Assignment

- Read Robert Martin chapter 6