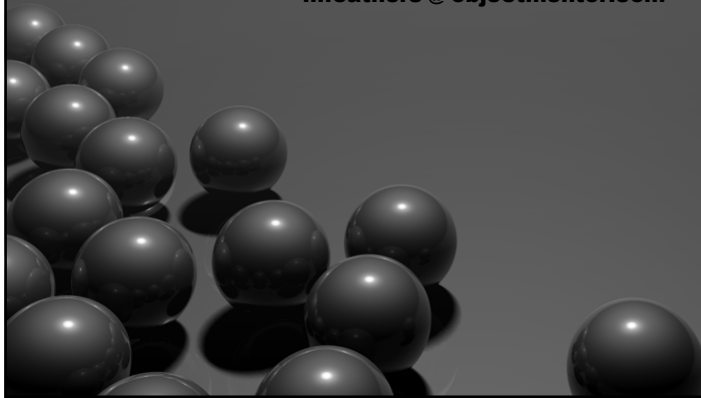


Working Effectively with Legacy Code

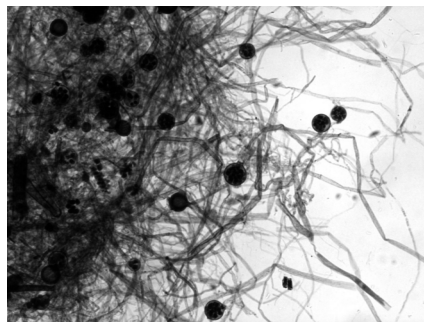
Michael Feathers
mfeathers@objectmentor.com



Topics



- What is this about?
- Breaking Dependencies
- *In Situ* Refactoring Strategies





A code base that bites back

© 2008 Object Mentor Incorporated. All rights reserved. / Page 3



The Refactoring Dilemma

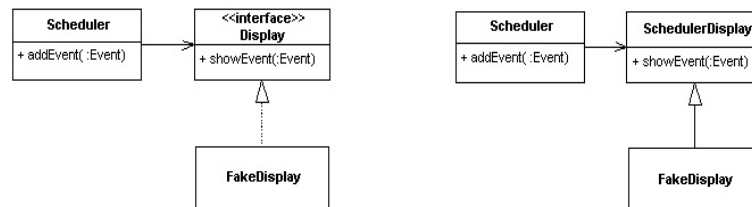
To refactor, we should have tests. To put tests in place, we have to refactor.

© 2008 Object Mentor Incorporated. All rights reserved. / Page 4

Choices



- Why choose one over another?

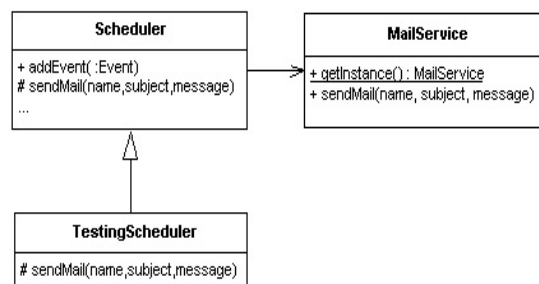


© 2008 Object Mentor Incorporated. All rights reserved. / Page 5

Extract and Override



- Breaking dependency with a *testing subclass*

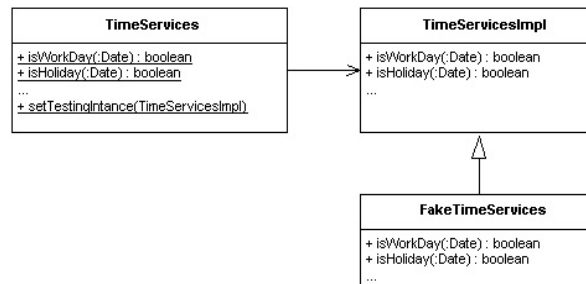


© 2008 Object Mentor Incorporated. All rights reserved. / Page 6

Delegating Instance



- Maintaining the facade

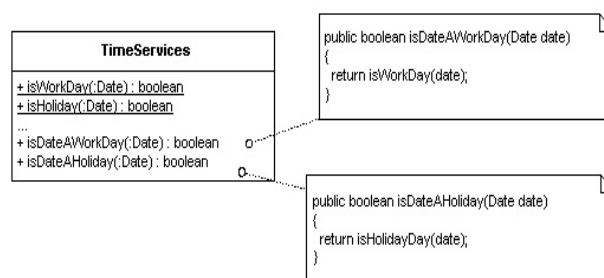


© 2008 Object Mentor Incorporated. All rights reserved. / Page 7

Instance Delegator



- Investing in a move away from globals



© 2008 Object Mentor Incorporated. All rights reserved. / Page 8

Finding the Scope



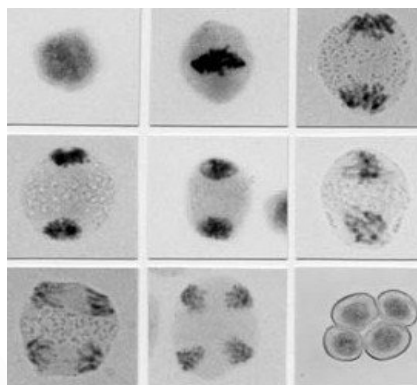
- Testing *near* versus testing *far*
 - Near – tests don't cover much, but cause of failures is obvious
 - Far – tests can cover longer paths but when they fail you have to debug
- Be pragmatic
 - Break dependencies where it is easy
 - Component-level scope may be too much
- You only have to do it once

© 2008 Object Mentor Incorporated. All rights reserved. / Page 9

Classes are like cells..

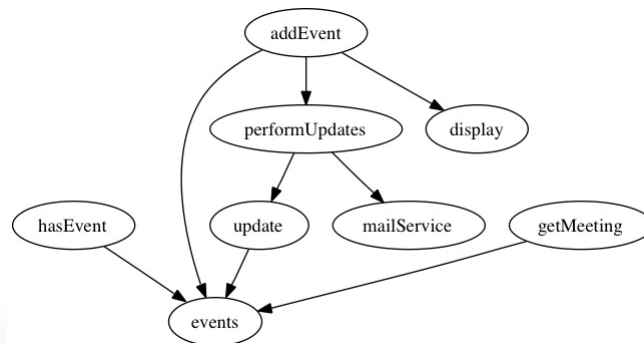


- Most designs need some meiosis



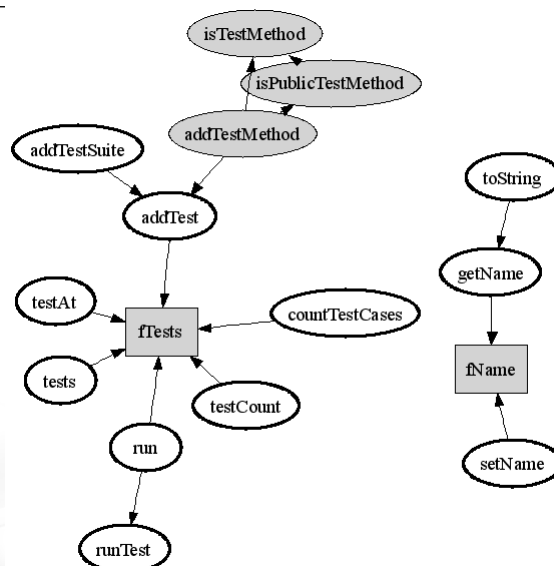
© 2008 Object Mentor Incorporated. All rights reserved. / Page 10

Scheduler – Internal Structure



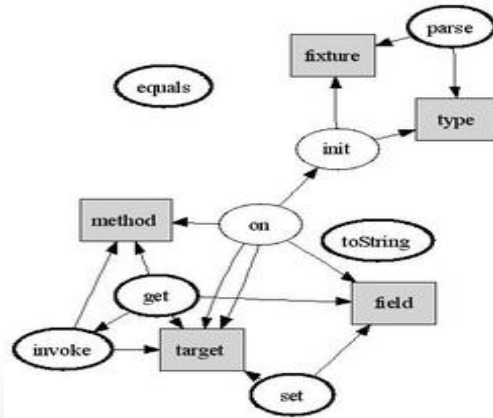
© 2008 Object Mentor Incorporated. All rights reserved. / Page 11

JUnit – TestSuite



© 2008 Object Mentor Incorporated. All rights reserved. / Page 12

FIT - TypeAdapter



© 2008 Object Mentor Incorporated. All rights reserved. / Page 13

Refactoring for Understanding



- Extract effectful and side-effect free methods separately
 - Look for possible places to split classes
- Consider extracting Method Objects
- Normalize Scopes in Long methods
- Do Scratch Refactorings

© 2008 Object Mentor Incorporated. All rights reserved. / Page 14

Refactoring Forward



- Dependency Injection is nice, but..
- Document *Constructor Sets* and look for patterns
- Remember: there is no such thing as a Global Variable

© 2008 Object Mentor Incorporated. All rights reserved. / Page 15

Know When To Say *No*



© 2008 Object Mentor Incorporated. All rights reserved. / Page 16