

# **Refactoring**

**Improving the design of Existing Code**

**Mark Windholtz**

**ObjectWind Software Ltd**



## **Who Am I**

- Mark Windholtz
- ObjectWind Software Ltd
- 17 years Building Professional Software
- Business, Retail, Medical, Military, Financial, etc.
- First contact with XP in Oct 1996
- Started Six XP teams
- Consulted for Nine other XP teams

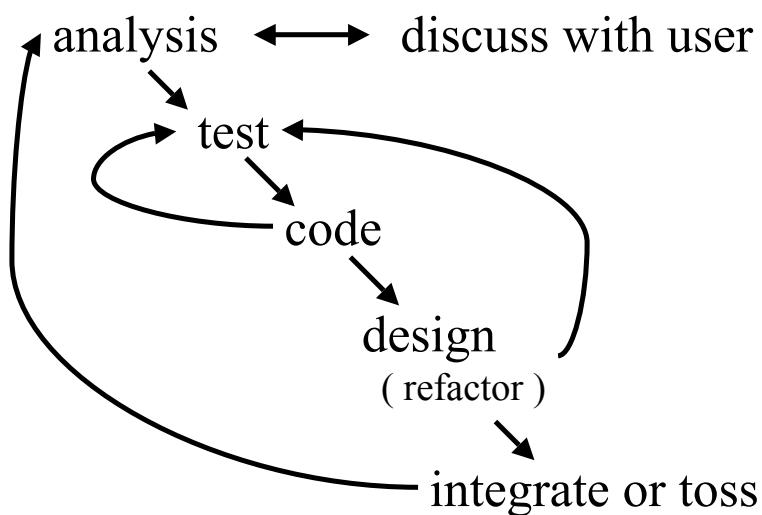


## Time Scale

- Product: Years
- Release: Month(s)
- Iteration: Week(s)
- Acceptance Test: 1-3 Days
- Episode: 1-3 hours then integrate
- Test: 1 to 5 minutes
- Code: 0 to 20 minutes to satisfy test case



## XP Episode Cycle



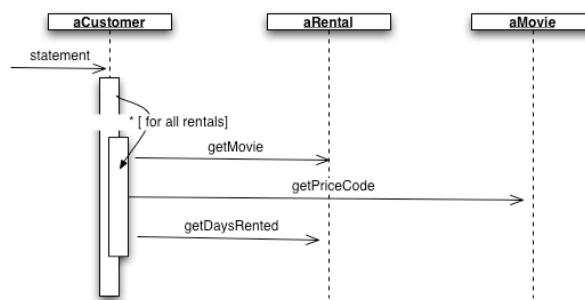
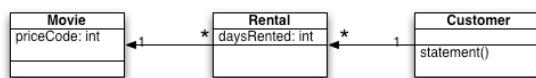


## Movie Rental

- Enter which movies a customer rented and for how long
- Program then calculates:
  - **Charges** which depend on
    - ✓ How long the movie is rented
    - ✓ Type of Movie: Regular, Children's, or New release
  - **Frequent Renter Points**
    - ✓ Varies depending on if film is new release or not



## Initial Design [7] (and Code Review)



Example from: Fowler, Martin.  
Refactoring: Improving the Design of Existing Code.  
Addison-Wesley, 2000



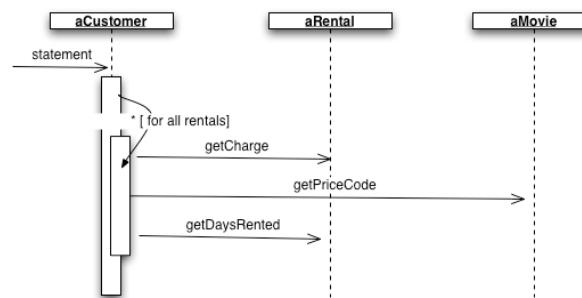
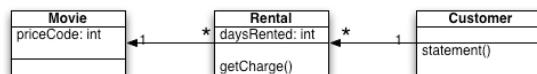
## New Customer Request

1. Customer wants to also print statements in HTML
  - Consider the impact.
  - Simple to copy and paste a whole new method
  - What happens then when the rules change?
2. Want to change how movies are classified
  - Don't know what kind of classifications yet
  - Changes could effect charge calculations
  - and/or frequent renter points.

(let's look at the statement method)

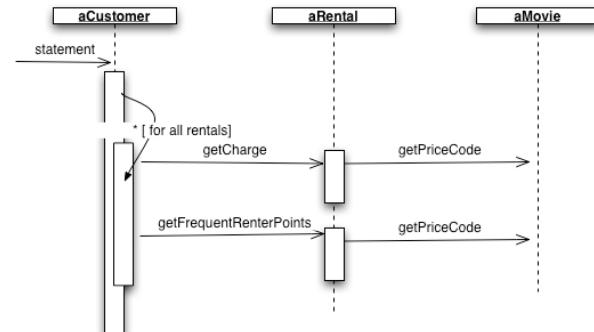
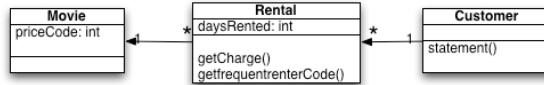


## Extracted getCharge()

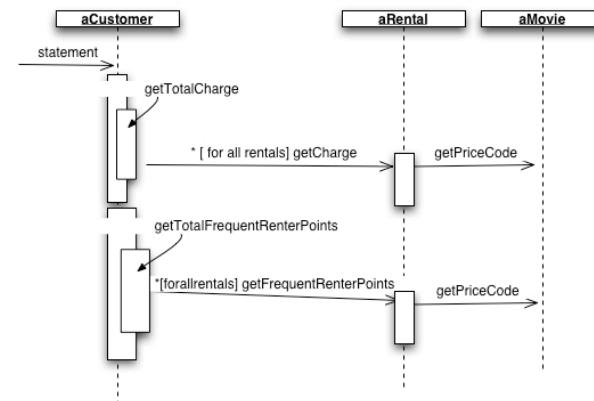
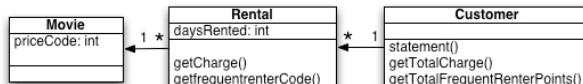




## Extract, move frequent renter points 6,7



## Extract totals 10,11





## public String **htmlStatement()** {

```

String result = "<H1>Rental Record for <EM>"  

+ getName() + "</EM></H1><P>\n";  

Enumeration rentalsEnum = rentals.elements();  

while (rentalsEnum.hasMoreElements())  

{  

    Rental each = (Rental) rentalsEnum.nextElement();  

// Show figures for this rental  

    result += each.getMovie().getTitle()  

    + ":" "  

    + each.getCharge()  

    + "<BR>\n";  

}  

// Add footer lines  

result += "<P>You owe <EM>"  

+ getTotalCharges() + "</EM></P>\n";  

result += "On this rental you earned <EM>"  

+ getTotalFrequentRenterPoints()  

+ "</EM> frequent renter points<P>";  

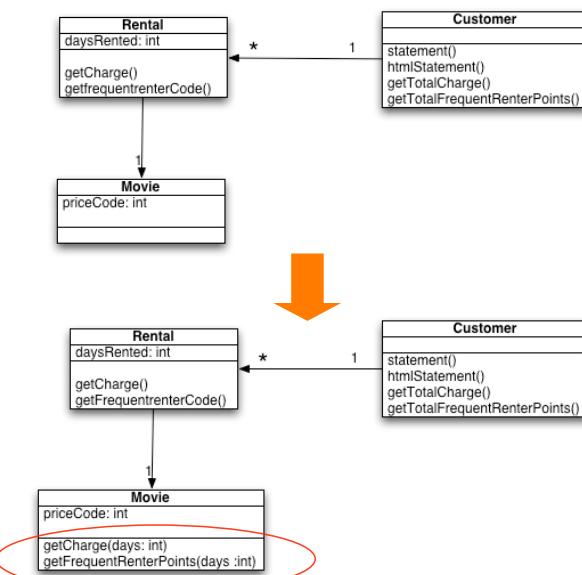
return result; }
```

www.ObjectWind.com

11



## Moving methods to Movie 12,13

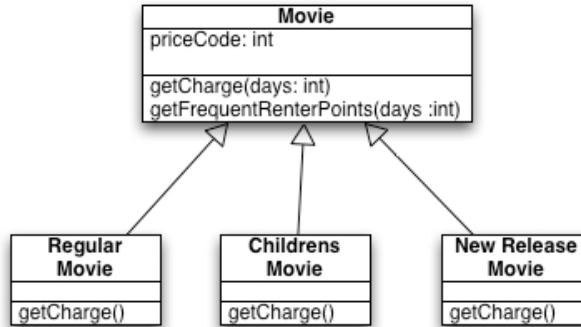


www.ObjectWind.com

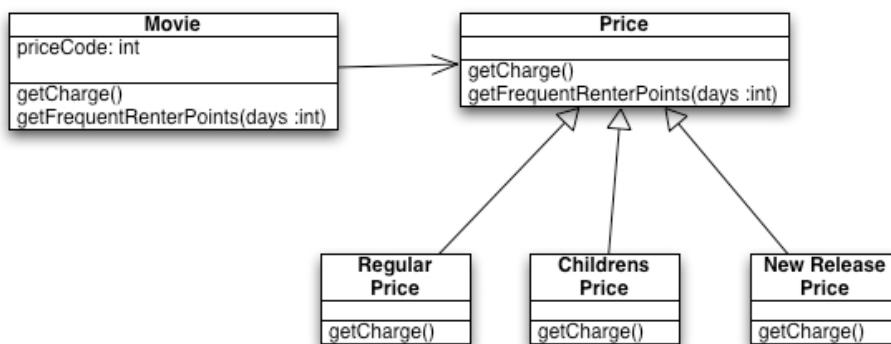
12



## Using Inheritance on Movie 14



## Using the State Pattern on Movie 15





## Was it worth it?

- Now it's much easier to ...
  - Change price behavior
  - Add new price
  - Add extra price dependent behavior
- In a larger (more realistic system) the number of price codes would be far greater and these changes would make a big difference
- Better distribution of responsibilities
- Easier to maintain
- Small change, test, small change, test, etc...



*There are 10 kinds of people in this world. Those who read binary numbers and those who don't...*



## Further Reading - 1

1. [www.junit.org](http://www.junit.org) - testing frameworks
2. [www.refactoring.org](http://www.refactoring.org)
3. Fowler, Martin. **Refactoring: Improving the Design of Existing Code.** Addison-Wesley, 2000
4. [www.pairprogramming.com](http://www.pairprogramming.com) - resources
5. [www.extremeprogramming.org](http://www.extremeprogramming.org) - on-line methodology manual
6. [www.xprogramming.org](http://www.xprogramming.org) - many articles
7. [www.objectWind.com](http://www.objectWind.com) - articles on Lean Software Development and XP