

Objectives

- Code Critique
- Refactoring for extensibility

Oct 31, 2008

Sprenkle - CS209

1

Reflection on Assignment 10

- How did you make design decisions?
- Were there any particularly difficult design decisions?
 - What were the tradeoffs?
- Did anybody consider making a FileData or File class?

Oct 31, 2008

Sprenkle - CS209

2

Assignment 10 Lessons

- Code should be soft
 - Eclipse makes code easier to change
 - The Refactor menu is a great resource
- Keep asking yourself
 - Is this understandable?
 - Will other people know what this code means?
 - Maintaining code and bug fixes are done much more than writing new code
 - How is this code most likely to change?
 - Does this code have a funny smell?
 - Literals, long methods, large classes, ...

Oct 31, 2008

Sprenkle - CS209

3

Review: Library System

- Write a program to simulate a library system
 - Borrowing items
 - Returning borrowed items
 - Paying fees
- A student with > 7 borrowed items, any one late item, or late fees greater than \$25.00 may not borrow anything new
- Late fees and due dates:

Library Item	Period	Late Fee
Book	14	\$0.50 per day
Video	2	\$5.00 one day late plus \$1.50 for each additional day late
CD	7	\$2.50 per day

Oct 29, 2008

Sprenkle - CS209

4

Review: Lendables

- Refactored code
 - Adjusted methods, comments, names, constructors
 - Order of parameters in constructors
 - Added appropriate calls to super class in subclass
 - Removed checkoutAnyLendable method
 - Move code into checkSelfOut
- Compare metrics
 - Number of attributes, methods, lines of code decreased

Oct 31, 2008

Sprenkle - CS209

5

Lendables: Testing after Refactoring

- Check our coverage
- Check our output for Video tests
 - Approaches to fix?

Oct 31, 2008

Sprenkle - CS209

6

Lendables Discussion

- Check our tests for Video
 - Wrong computation of late fee
- Approaches to fix:
 - Override Video's `getLateFee`
 - Add `firstDayLateFee` field to `Lendable`
 - Adjust `getLateFee`
- Which approach is more flexible/maintainable?

Oct 31, 2008

Sprenkle - CS209

7

Lendables Discussion

- `toString()` method
- `my_availability`

Oct 31, 2008

Sprenkle - CS209

8

Contrasting Lendables to Bins

Bins

- Readability, code smells

Lendables

- Extensibility

Oct 31, 2008

Sprenkle - CS209

9

Simulating a Roulette Game

[Roulette Specification](#)

Oct 31, 2008

Sprenkle - CS209

10

Get a Solution

- Import Existing Project
 - `/home/courses/cs209/handouts/roulette.tar`
- Enable Metrics

Oct 31, 2008

Sprenkle - CS209

11

Understanding Code

- Execute the code
 - What is the main driver for this project?
- What are each class's responsibilities?
- What does `test.RouletteTestSuite` do?

Oct 31, 2008

Sprenkle - CS209

12

Bug in the Code

- Determining if Odd/Even Bet was won is incorrect

Oct 31, 2008

Sprenkle - CS209

13

Understanding Code

- Focus: how open is the code to adding new kinds of bets and how closed it is to modification?
 - How many classes know about the Bet class?
 - What code would need to be added to Game to allow the user to make another kind of bet that paid one to one odds and was based on whether the number spun was high (between 19 and 36) or low (between 1 and 18)?

Oct 31, 2008

Sprenkle - CS209

14

Assignment 11

- Understanding code
- Refactoring for Extensibility
- Adding the remaining 3 bets
 - Compare new code to old code

Oct 31, 2008

Sprenkle - CS209

15