CS207.01 2013F, Class 15: Documentation and Javadoc

Overview

- Preliminaries
  - Admin.
  - Questions on HW4.
- Recent ideas.
- Documentation.
- Javadoc.
- Informal lab.

Admin

- Reading for Wednesday: An Introduction to Android Development
- I’ll distribute Android tablets and cases today. Please make sure your tablet is charged for class on Wednesday and that you’ve done the initial setup.
- Today’s lab is documented in the EBoard and Outline. (It’s informal.)
- EC Opportunities
  - Learning from Alumni: Jim Finnessy
  - CS Extras Thursday @ 4:30: Jennelle Nystrom on Microsoft
  - CS Table Friday (The Story of Mel, A Real Programmer)
  - G-Tones, Saturday, 3:30 at Herrick Chapel
  - Baseball Scrimmage Sunday 11 am, we think
  - Water Polo Saturday/Sunday
  - Others?
- Other things
  - Face the Bear! Go and see what’s at the Bear 7-9 Wednesday night
  - Swipe your P-Card at the Football game

HW4 Questions

*How do we test if the constructor throws an exception?*

```java
try {
    Fraction f = new Fraction(1,0);
    fail("Zero denominator");
} catch (Exception e) {
}
```

*Can we pay attention to precedence?*
Yes

Recent ideas

- You can have a variable of class A and initialize it with class B, provided class B is a subclass of A.
  
  ```java
  Counter c = new DoubleCounter(10);
  ```

- The activity of an object is determined by its actual type, not by the type you think it is.
- The method change is called "overriding" and should be marked with `@Override`
- Why declare something as the superclass?
  - Most frequently, we do this when writing a procedure.
  
  ```java
  public void logStudent(Counter c, Student s)
  ```

- In contrast, if we write `Super s = new Sub(...)`; We can only access the method names of Sub that are in Super

Documentation

- Help people understand code
- Audiences
  - Clients - Care about the *what* (and efficiency), not so much *how*
  - Users - Mostly about the UI
  - Fellow developers - Care about the implementation issues and overall structure
  - Yourself (as client, fellow developer, ...)
  - Maintainers/Supporters
- Other Audiences in a class environment
  - Grades
- Commenting early helps you think about problems
  - What’s the big picture - what pieces are there and how do they fit together?
  - What do I expect this procedure to do?
  - How do I achieve that goal?
- Our focus is clients

Javadoc

- Internal comments that provide nice documentation for clients (in HTML).
- Use slash-star-star ... star-slash
- Immediately precede class, interface, constant, method, field, etc. declarations

Informal lab.

- Start with your classes from the inheritance lab. Don’t add comments (yet).
- Use the Javadoc tool to generate Javadoc.
  - Project > Generate Javadoc
  - You can put the documentation in your project.
- But you may find it useful to put it elsewhere.
- Preview the Javadoc in Eclipse.
- Preview the Javadoc in your Web browser.
  - file:///home/username/path/to/project/doc/index.html
- Add some Javadoc-style comments and see their effects.
- Figure out how to get different levels of protection revealed (or not revealed)
- Add preconditions and postconditions.
  - tag pre:cm:"Preconditions"
  - tag post:cm:"Postconditions:"
- Play (e.g., add constants and other things.)

Copyright (c) 2013 Samuel A. Rebelsky.

This work is licensed under a [Creative Commons Attribution 3.0 Unported License](http://creativecommons.org/licenses/by/3.0/) To view a copy of this license, visit [http://creativecommons.org/licenses/by/3.0/](http://creativecommons.org/licenses/by/3.0/) or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.