CS 520
Theory and Practice of Software Engineering
Fall 2017

OO design and Java 8 features

October 03, 2017
Today

- MAP postmortem.
- Another MVC (source code) example.
- A primer on lambdas and method references (Java 8).
- A first paper/poster review.
Course overview: the big picture

● **Software architecture and design**
  ○ Software modelling and UML crash course.
  ○ Best practices and OO design principles.
  ○ Architecture and Design patterns.

● **Empirical Software Engineering**
  ○ Reasoning about experimental designs and studies.
  ○ Understanding and reasoning about threats to validity.

● **Software testing, debugging, and repair**
  ○ Learning about cutting-edge research.
  ○ Hands-on experience, using testing and debugging techniques.

● **Class project**
  ○ Design, development, and testing of a research prototype, etc.
Course overview: the big picture

● **Software architecture and design**
  ○ Software modelling and UML crash course.
  ○ Best practices and OO design principles.
  ○ Architecture and Design patterns.

● **Empirical Software Engineering**
  ○ Reasoning about experimental designs and studies.
  ○ Understanding and reasoning about threats to validity.

● **Software testing, debugging, and repair**
  ○ Learning about cutting-edge research.
  ○ Hands-on experience, using testing and debugging techniques.

● **Class project**
  ○ Design, development, and testing of a research prototype, etc.
MAP: Postmortem

Overview

● Organization and logistics
● Material
● Grading and workload
MVC revisited

Design patterns in an MVC architecture

- View
- Controller
- Model
- Client

- Composite
- Strategy
- Observer
A primer on Java 8 features: Lambdas and method references
Recall our Median implementation (strategy pattern)

```
<<interface>>
Median
  +median(a:double[]):double

<<interface>>
Sorter
  +sort(array:double[])

StrategyMedian
  -sortStrategy:Sorter
    +median(a:double[]):double
    +setSorter(s:Sorter)

HeapSort
  +sort(...) 

QuickSort
  +sort(...) 
```

Is this a reasonable solution, given the simplicity of the problem?
// The Sorter interface.
public interface Sorter {
    public void sort(double[] numbers);
}

// A concrete Sorter implementation.
public class QuickSort implements Sorter {
    public void sort(double[] numbers) {
        java.util.Arrays.sort(numbers);
    }
}

@test
public void testQuickSort() {
    Median m = new Median();
    m.setSorter(new QuickSort());
    double actual = m.median(1, 2, 3, 4);
    assertEquals(2.5, actual, EPS);
}
A lot of code and new classes for just calling sort

```java
// The Sorter interface.
public interface Sorter {
    public void sort(double[] numbers);
}

// A concrete Sorter implementation.
public class QuickSort implements Sorter {
    public void sort(double[] numbers) {
        java.util.Arrays.sort(numbers);
    }
}

@Test
public void testQuickSort() {
    Median m = new Median();
    m.setSorter(new QuickSort());
    double actual = m.median(1, 2, 3, 4);
    assertEquals(2.5, actual, EPS);
}
```

What are alternatives to creating a new class to wrap QuickSort?
Live coding: lambdas and method references

See code example (online)

- **lambda**
  - Test class *AllTests* provides four test cases.
  - Four possibilities to test the same behavior of *StrategyMedian*.
    - Using a concrete class *QuickSort*.
    - Using an *anonymous inner class*.
    - Using a *lambda expression*.
    - Using a *method reference* (Method references simplify lambda expressions that call a single method).

Use **ant** to compile and test the code:

```
$ant -p       => list all targets
$ant compile  => compile the code
$ant test     => run all four tests, which test the same behavior
```

Key concepts: target typing and type inference.
A first paper/poster review
What’s this?
What’s this?
Studying brain activity using fMRI scans
Now, what’s this?
There is an obvious experiment to conduct...

A first paper/poster review

A first paper/poster review

Important questions to answer:

- What is the **motivation** of the paper?
- What are the **main objectives** of the paper?
- What are **strengths** and **weaknesses** of the paper?
- Is the **methodology** and **presentation** convincing and adequate?
- What are **open questions** or **controversial statements**?
A first paper/poster review

Important questions to answer:

- What is the **motivation** of the paper?
- What are the **main objectives** of the paper?
- What are **strengths** and **weaknesses** of the paper?
- Is the **methodology** and **presentation** convincing and adequate?
- What are **open questions** or **controversial statements**?

Generally, when interpreting results and statistics, check for red herrings and dead salmons :)