Logistics

Project proposals

- Elevator pitches on Thursday
  - Deadline for creating groups: 02/03/2016, 11:55pm
  - Deadline for submitting slides: 02/04/2016, 8:00am
  - 2 minutes (~2 slides, a single pdf).
  - Slides should clearly indicate:
    - 520 or 620 project (e.g., 520: My awesome project).
    - the current group members.
- Time to discuss ideas and (re-)group after presentations.

Recap

- Design patterns
  - What is a design pattern?
  - Categories of design patterns
    - Structural patterns
    - Creational patterns
    - Behavioral patterns
- Two design problems & potential solutions
  - Complex View -> composite pattern
  - Input streams -> decorator pattern
Today

More on design patterns
- Decorator pattern revisited
- Behavioral patterns
  - Template method
  - Strategy
  - ...

Decorator vs. composite

```
Decorator vs. composite

```

Decorator pattern revisited
```
InputStream is = new FileInputStream(...);
int b;
while((b = is.read()) != -1) {
    // do something
}
...
```

Suppose the client needs 1 byte at a time ...
```
FileInputStream
+ read():int
...
```

```
InputStream is =
new FileInputStream(...);
int b;
while((b = is.read()) != -1) {
    // do something
}
...
```

```
FileInputStream
+ read():int
...
```

```
FileInputStream
+ read():int
...
```
Decorator pattern revisited

```
InputStream is = new FileInputStream(...);
int b;
while((b=is.read()) != -1) {
    // do something
}
```

Only half the truth...

Problem: filesystem IO is expensive

Solution: use a buffer!

Why not simply implement the buffering in the client or subclass?
Decorator pattern revisited

```java
InputStream is = new BufferedInputStream(
    new FileInputStream(...));
int b;
while((b=is.read()) != -1) {
    // do something
}
```

Find the median in an array of doubles

Examples:
- median([1, 2, 3, 4, 5]) = 3
- median([1, 2, 3, 4]) = 2.5

Algorithm:
Input: array of length n   Output: median

1. Sort array
2. if n is odd return ((n+1)/2)th element
   otherwise return arithmetic mean of
   (n/2)th element and ((n/2)+1)th element

One possible solution: template method

```java
AbstractStats
+ median(a:double[]):double
  + sort(a:double[]):double
MyStats
+ sort(a:double[])
```

The template method (median) implements the algorithm but leaves the sorting of the array open.

The concrete subclass only needs to implement the actual sorting.
Another possible solution: strategy

MyStats delegates the sorting of the array to a “strategy”, which can be configured and changed at runtime.

Template method vs. strategy

Two solutions to the same problem

Template method
- Behavior selected at compile time.
- Don’t call us, we’ll call you.
- Template method is usually final.

Strategy
- Behavior selected at runtime.
- Composition over inheritance.