Design Patterns in Java 8

•••

Contact Info

Ken Kousen

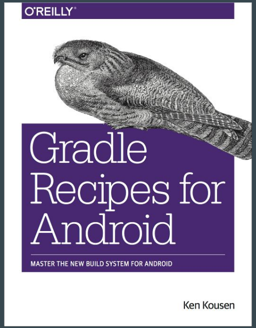
Kousen IT, Inc.

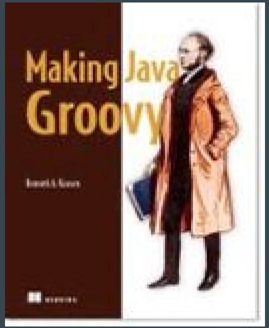
ken.kousen@kousenit.com

http://www.kousenit.com

http://kousenit.org (blog)

@kenkousen


O'REILLY
Gradle Recipes for Android
MASTER THE NEW BUILD SYSTEM FOR ANDROID
Ken Kousen



Videos

O'Reilly video courses: See Safari Books Online for details

- Groovy Programming Fundamentals
- Practical Groovy Programming
- Mastering Groovy Programming
- Learning Android
- Practical Android
- Gradle Fundamentals
- Gradle for Android
- Spring Framework Essentials
- Advanced Java Development

Modern Java Recipes

Source code:

https://github.com/kousen/java_8_recipes

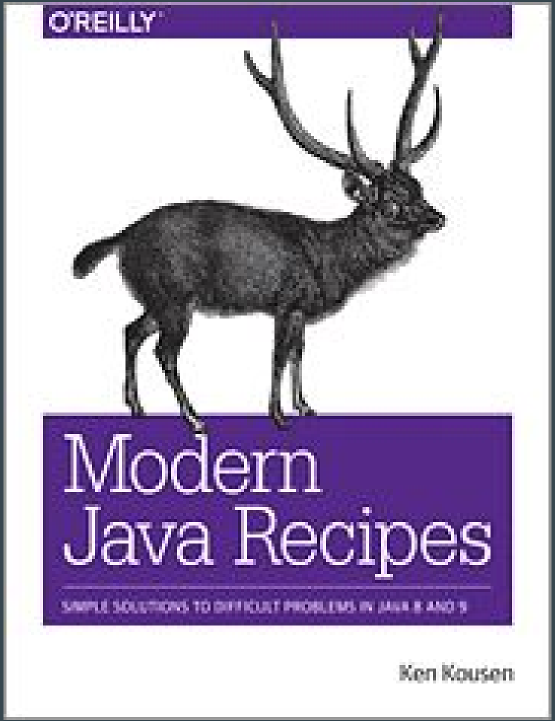
https://github.com/kousen/from-gof-to-lambda

which my fork of Mario Fusco's

https://github.com/mariofusco/from-gof-to-lambda

Also from Venkat Subramaniam's Agile Developer site

http://agiledeveloper.com/



Mario Fusco

Design Patterns posts:

https://www.voxxed.com/2016/04/gang-four-patterns-functional-light-part-1/

https://www.voxxed.com/2016/05/gang-four-patterns-functional-light-part-2/

https://www.voxxed.com/2016/05/gang-four-patterns-functional-light-part-3/

Me: Hey, I'm writing a talk on design patterns in Java

You: Dude, 1994 called and they want their talk back

Me: Yeah, well, 1998 called and they want their meme back

You: Tell them they can keep it

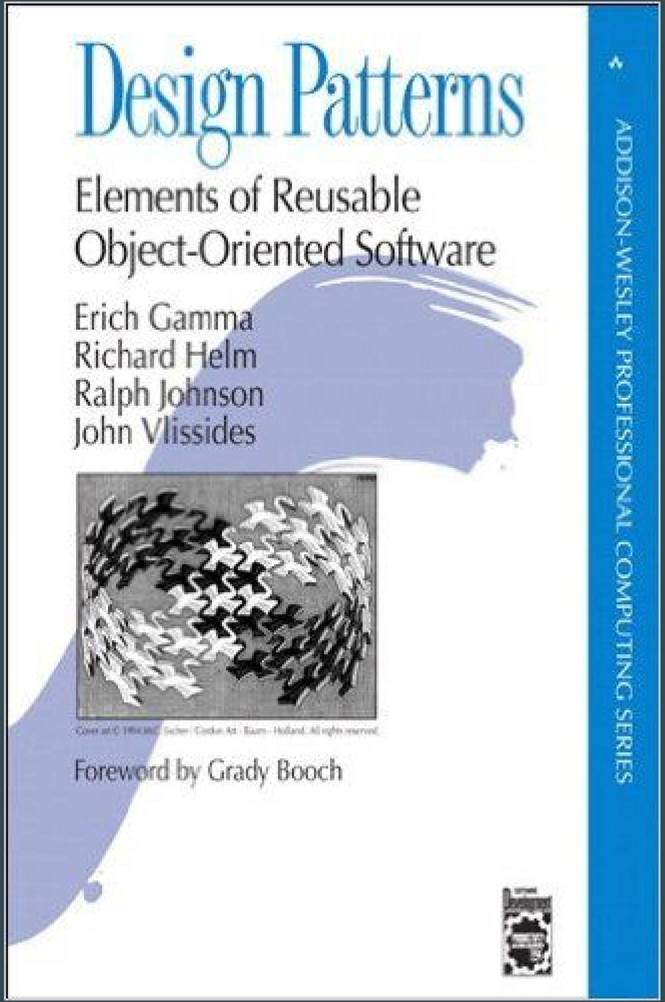
http://www.urbandictionary.com/define.php?term=X%20Called%2C%20they%20want%20their%20Y%20Back

Design Patterns:

Elements of Reusable
Object-Oriented Software

Gamma, Helm, Johnson, and Vlissides

Published October 1994 (!)


Design Patterns
Elements of Reusable Object-Oriented Software
Erich Gamma
Richard Helm
Ralph Johnson
John Vlissides
Foreword by Grady Booch
ADDISON-WESLEY PROFESSIONAL COMPUTING SERIES

That book sold over 500,000 copies

That book sold over 500,000 copies

Mine haven't sold that many, or I wouldn't be here

That book sold over 500,000 copies

Mine haven't sold that many, or I wouldn't be here

(Maybe I would, but I'd be staying in a better hotel)

That book sold over 500,000 copies

Mine haven't sold that many, or I wouldn't be here

(Maybe I would, but I'd be staying in a better hotel)

(And I'd be driving a brand new Tesla)

That book sold over 500,000 copies

Mine haven't sold that many, or I wouldn't be here

(Maybe I would, but I'd be staying in a better hotel)

(And I'd be driving a brand new Tesla)

(And I would never, ever fly coach again. <Wistful sigh>)

Design Patterns book

Code examples were in what languages?

Design Patterns book

Code examples were in what languages?

Anybody?

Design Patterns book

Code examples were in what languages?

Anybody?

Yup: C++ and Smalltalk

GoF Patterns

Three categories:

- Creational
- Structural
- Behavioral

GoF Patterns

Wait, what? Behavioral? Why wrap operations inside classes?

https://steve-yegge.blogspot.com/2006/03/execution-in-kingdom-of-nouns.html

GoF Patterns

Wait, what? Behavioral? Why wrap operations inside classes?

https://steve-yegge.blogspot.com/2006/03/execution-in-kingdom-of-nouns.html

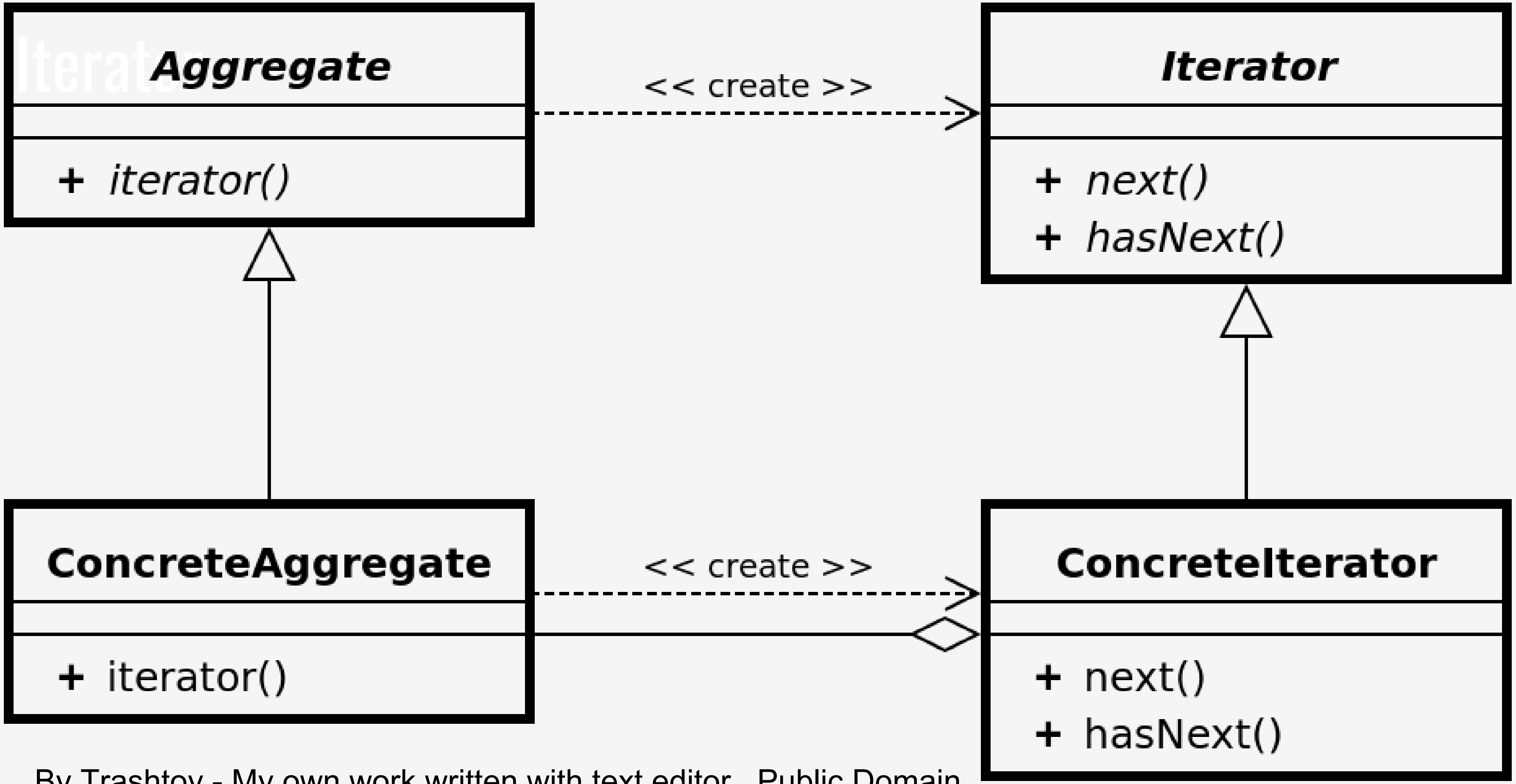
But now we have lambda expressions

TRIGGER WARNING: *UML Diagrams ahead*

Iterator

Okay, this one's easy. As of Java 1.2:

```
java.util.Iterator

    -   boolean hasNext()
    -   E next()
    -   default void remove()
    -   default void forEachRemaining(Consumer)
```



By Trashtoy - My own work written with text editor., Public Domain, https://commons.wikimedia.org/w/index.php?curid=1698830

Iterator

But there's more (Java 1.5)

```
java.util.Iterable

    -  default void forEach(Consumer)
    -  Iterator<T> iterator()
    -  default Spliterator<T> spliterator()
```

Used in for-each loops

Iterator

- `java.util.Collection extends Iterable`
  - Returns an iterator via the `iterator()` method
  - Concrete classes implement as an inner class

Iterator

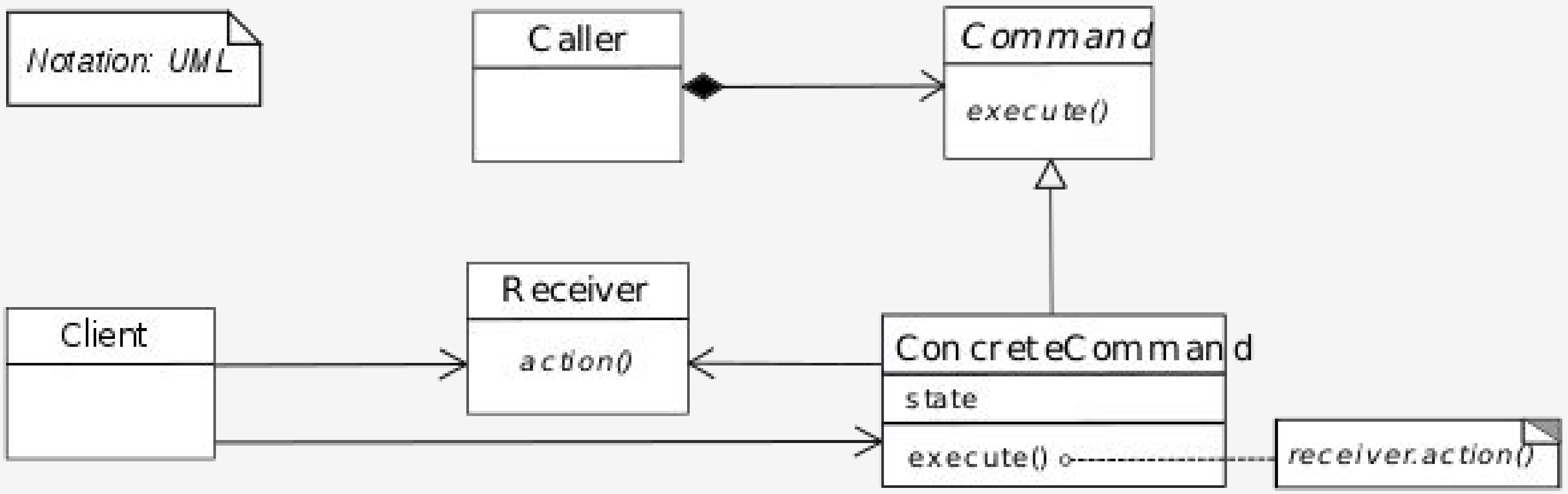
```
public Iterator<E> iterator() {
    return new Itr();
}

private class Itr implements Iterator<E> {
    …
}
```

Iterator

```java
@Override
public void forEach(Consumer<? super E> action) {
    Objects.requireNonNull(action);
    for (E e : a) {
        action.accept(e);
    }
}
```

Command

Creates objects to encapsulate actions and parameters


Notation: UML
Caller
Command
execute()
Receiver
action()
Client
ConcreteCommand
state
execute()
receiver.action()

Command

Define a Command interface, usually with a single method

Create classes that implement that method

Create an "executor"

- Takes a collection of Command objects
- Executes the interface method on each one

Command

- Java 8 adds lambda expressions
    - Now you don't need an object wrapper
    - All you need is to pick a functional interface
        - `Runnable` is frequently a good choice

Command

This all sounds familiar

Remember ~~Jakarta~~ Apache Struts ~~1~~ 2?

- A Model 2 framework

Command

- This all sounds familiar
- Remember ~~Jakarta~~ Apache Struts ~~1~~ 2?
    - A Model 2 framework
    - Controllers are classes that extend Action

Command

This all sounds familiar

Remember ~~Jakarta~~ Apache Struts ~~1~~ 2?

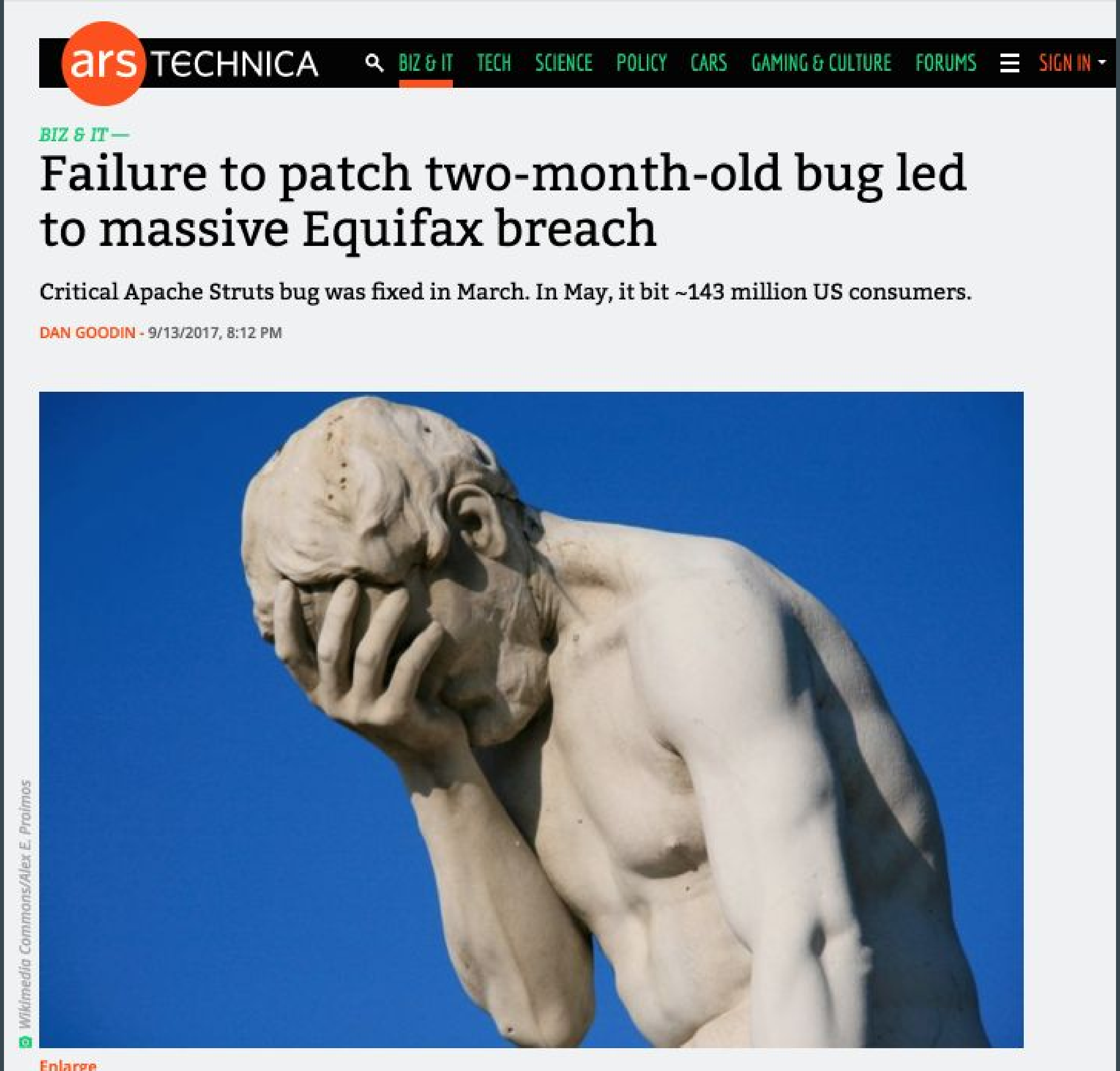
- A Model 2 framework
- Controllers are classes that extend Action
- Action has a single method called `execute()`

Command

This all sounds familiar

Remember ~~Jakarta~~ Apache Struts ~~1~~ 2?

- A Model 2 framework
- Controllers are classes that extend Action
- Action has a single method called `execute()`
- Framework collected Action instances and invoked execute as needed

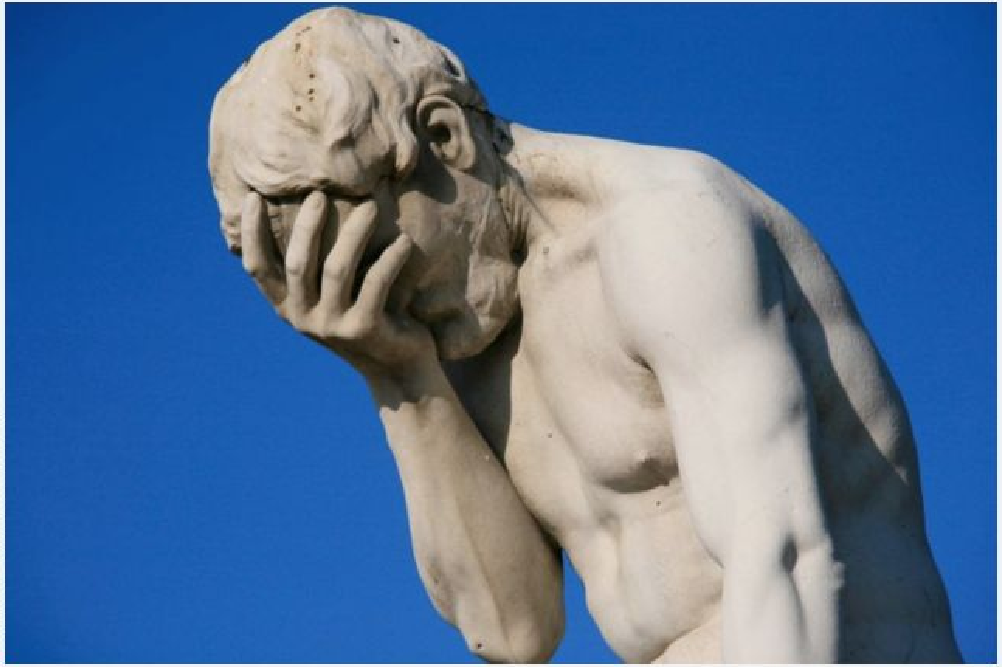


BIZ & IT —

Failure to patch two-month-old bug led to massive Equifax breach

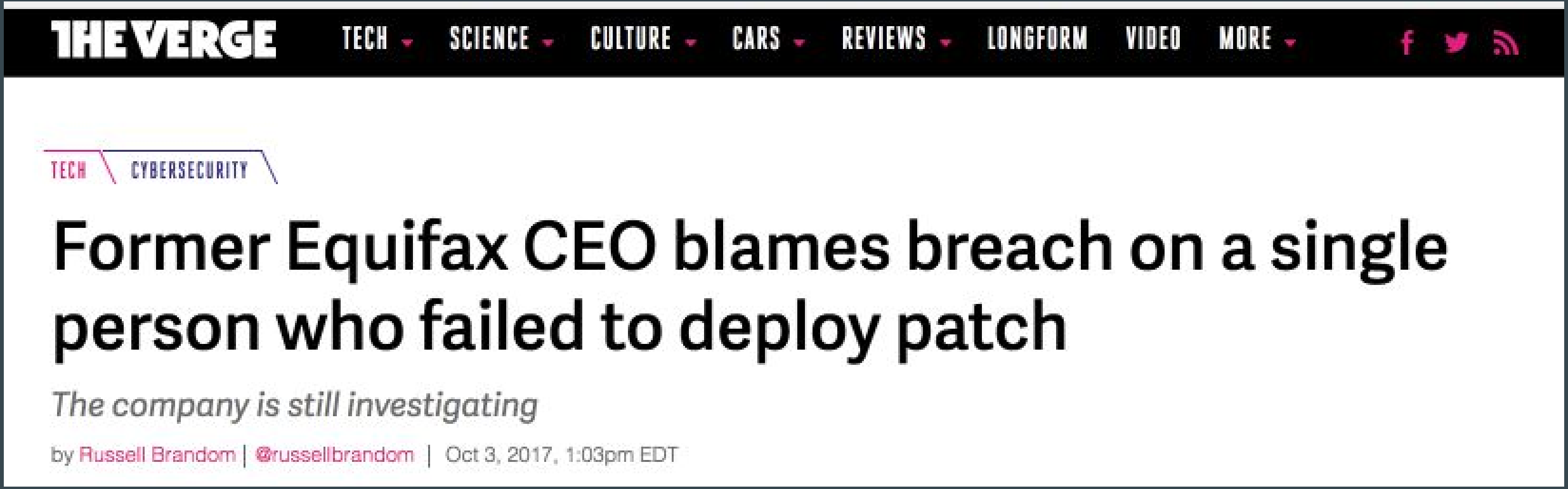
Critical Apache Struts bug was fixed in March. In May, it bit ~143 million US consumers.

DAN GOODIN - 9/13/2017, 8:12 PM



Wikimedia Commons/Alex E. Proimos

Enlarge


THE VERGE TECH SCIENCE CULTURE CARS REVIEWS LONGFORM VIDEO MORE

TECH CYBERSECURITY

Former Equifax CEO blames breach on a single person who failed to deploy patch

The company is still investigating

by Russell Brandom | @russellbrandom | Oct 3, 2017, 1:03pm EDT

I SAY WE TAKE OFF AND NUKE THE ENTIRE SITE FROM ORBIT...
COMPANY
IT'S THE ONLY WAY TO BE SURE...
imgflip.com

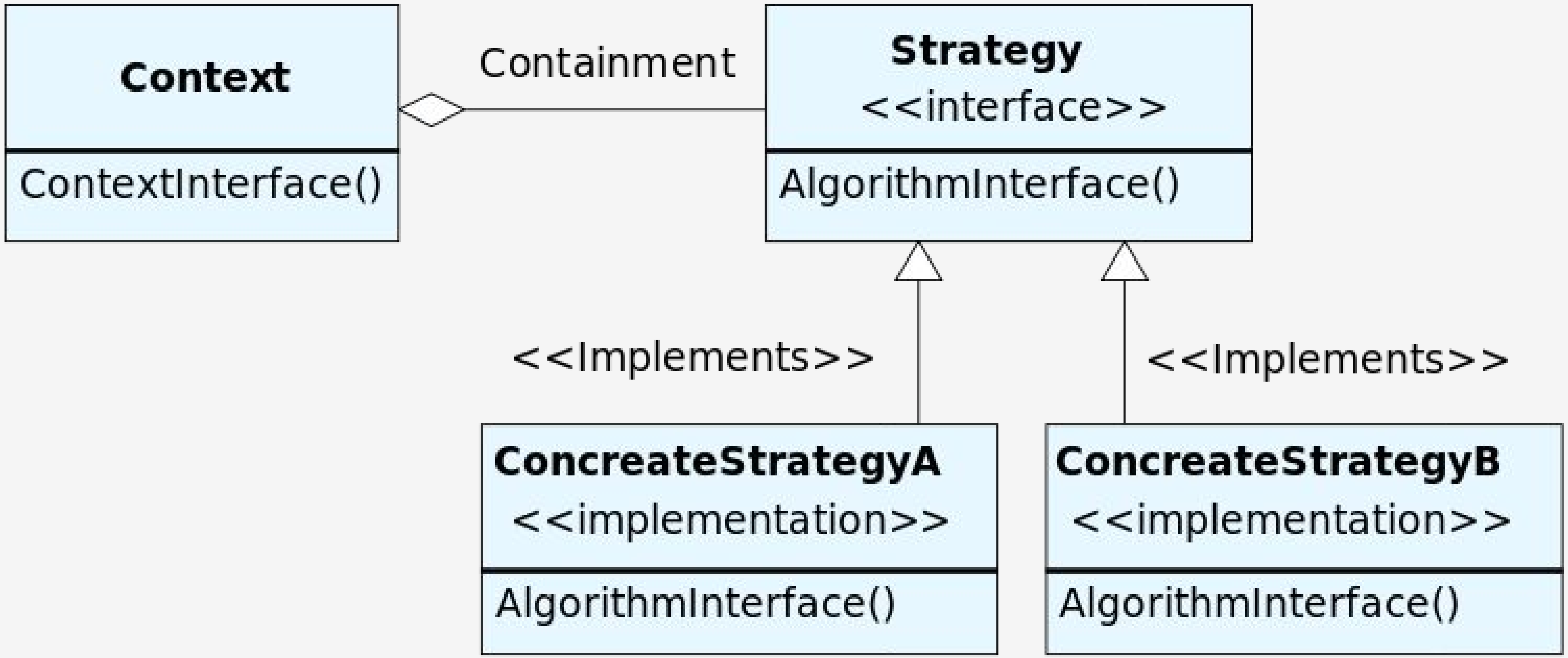
Strategy

Select one of a family of algorithms at run time

Bunny Hugged

Merrie Melodies (*Bugs Bunny*) series

Bugs using a little "stragety" to gain the upper hand on the Crusher.


Context
ContextInterface()
Containment
Strategy
<<interface>>
AlgorithmInterface()
<<Implements>>
<<Implements>>
ConcreateStrategyA
<<implementation>>
AlgorithmInterface()
ConcreateStrategyB
<<implementation>>
AlgorithmInterface()

Strategy

- Container delegates to a "Strategy" object
    - Include a setter method to select the Strategy
    - Create multiple classes that all implement the Strategy
    - User chooses strategy and executes method on Container

Strategy

How do lambdas change that?

- Each strategy implements a functional interface
- Container methods take lambdas as argument(s)

What else happened in 1994?

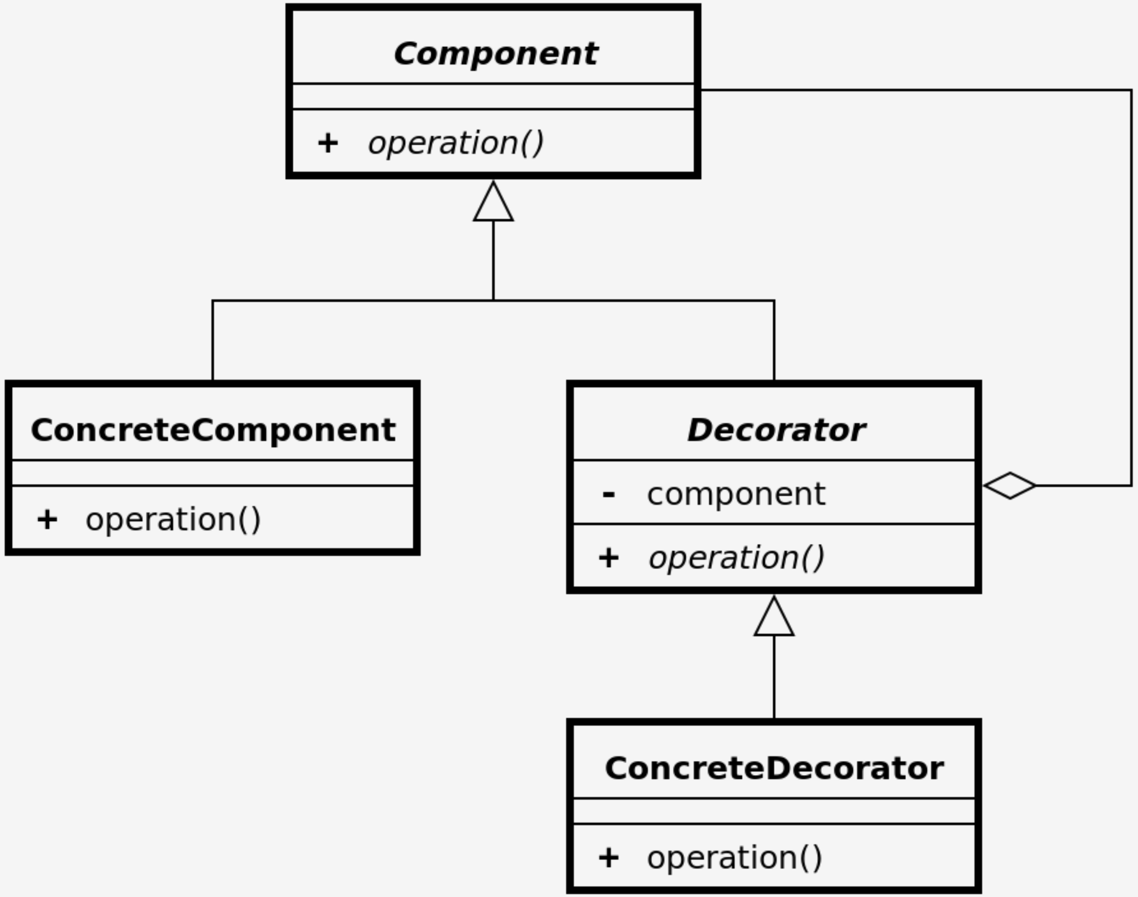
- O.J. Simpson drives a Ford Bronco on the highway

Howard, you have no idea what' s coming...

Decorator (Wrapper)

Dynamically adds/overrides behavior on an object

Replaces inheritance with composition



By Trashtoy - My own work, written with text editor., Public Domain, https://commons.wikimedia.org/w/index.php?curid=1797712

Decorator

Concrete classes and decorators both implement same interface

Concrete classes can stand alone; decorators can not

Decorators wrap concrete classes and each other

Method call passes through the series of objects and back

Decorator

How do lambdas change that?

Closure composition

Predicate:	`and, or, negate`
Consumer:	`andThen`
Function:	`andThen, compose`

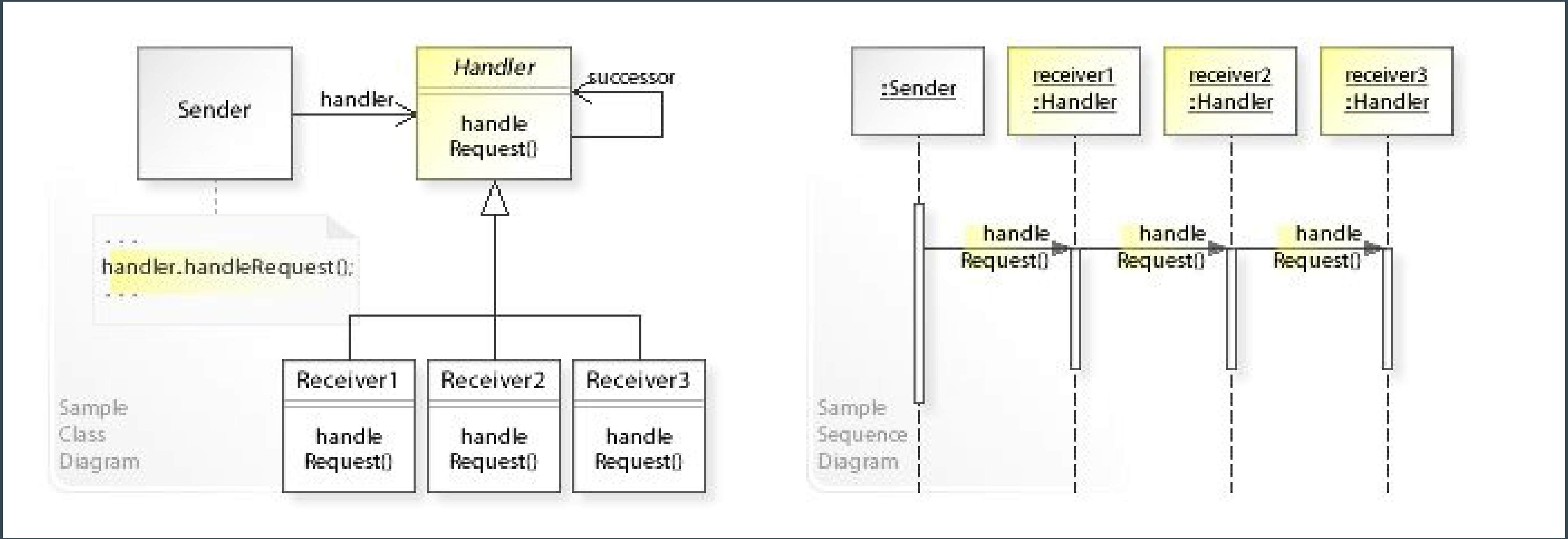
`reduce(BinaryOperator)` method builds complete closure

1994: Tonya Harding and Nancy Kerrigan kerfuffle

Chain of Responsibility

Delegates commands to a chain of processing objects

Chain of Responsibility



Chain of Responsibility

Handlers form a chain, which is essentially a linked list

Each handler either handles the argument, or passes it on

Chain of Responsibility

- What happens with lambdas?
    - Decisions using `filter(Predicate)`
    - Handle using `map(Function)`
- Note: Java 9 adds `stream` method to `Optional`

Also from 1994

The Lion King

Remember: before it all worked out,

Scar became King

The Lion King

(along with his kids

Scar, Jr, Eric, and Ivanka)

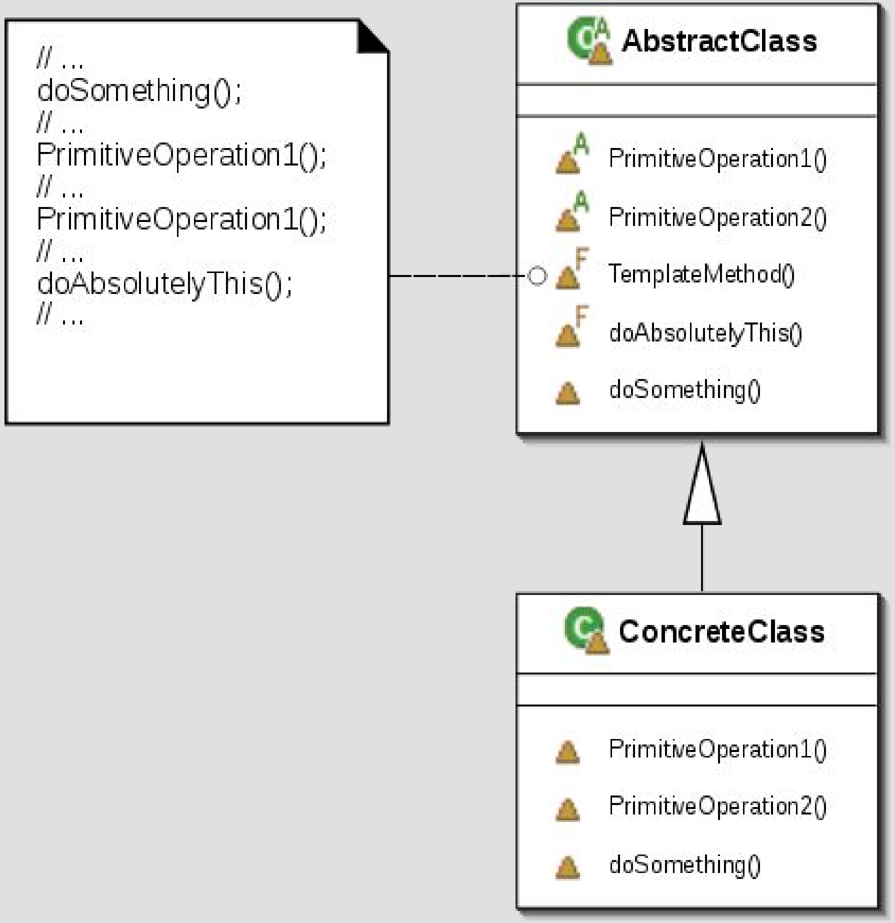
Execute Around Method

Not GoF, but related to Template Method

Scala people call it Loan

Execute Around Method

- Put resource management in the infrastructure
    - Allocation, cleanup
    - Client provides a `Consumer` that uses the Resource



```
// ...
doSomething();
// ...
PrimitiveOperation1();
// ...
PrimitiveOperation1();
// ...
doAbsolutelyThis();
// ...
```

Template Method

Create a (final) method that delegates to others

Client overrides the other methods

Template method calls the others in the proper order

Template Method

How do lambdas help?

Each method is a `Function`

Function maps from arg to `Optional`

Make a stream of them

Filter on `Optional.isPresent()`

Use `findFirst()` to get the actual result

Conclusions

- Design Patterns wrapped methods in classes
- Lambdas let us unwrap them
- Composition helps

Remember:

"a comic says funny things; a comedian says things funny"

Conclusions

- Design Patterns wrapped methods in classes
- Lambdas let us unwrap them
- Composition helps

Remember:

"a comic says funny things; a comedian says things funny"

That makes me a ... software developer.