FP vs. OOP: Beyond the Bike shed

Colin Jones
@trptcolin
https://8thlight.com
background
bikesheds
C. Northcote Parkinson
Parkinson’s Law of Triviality
bikesheds
10 lines of code = 10 issues.

500 lines of code = "looks fine."

Code reviews.

1:58 AM - 5 Nov 2013
Why Should I Care What Color the Bikeshed Is?

"The really, really short answer is that you should not. The somewhat longer answer is that just because you are capable of building a bikeshed does not mean you should stop others from building one just because you do not like the color they plan to paint it. This is a metaphor indicating that you need not argue about every little feature just because you know enough to do so. Some people have commented that the amount of noise generated by a change is inversely proportional to the complexity of the change."

(If you don't like the way we painted this bikeshed, try bikeshedding.io, or the minimalist shed.bike. Or set up your own? That's the spirit!)
http://bikeshed.com/

Why Should I Care What Color the Bikeshed Is?

From freebsd.org/doc/en_US.ISO8859-1/books/faq/misc.html#bikeshed-painting; see also phk.freebsd.dk/sagas/bikeshed.html for historical background.

"The really, really short answer is that you should not. The somewhat longer answer is that just because you are capable of building a bikeshed does not mean you should stop others from building one just because you do not like the color they plan to paint it. This is a metaphor indicating that you need not argue about every little feature just because you know enough to do so. Some people have commented that the amount of noise generated by a change is inversely proportional to the complexity of the change."

(If you don't like the way we painted this bikeshed, try bikeshedding.io, or the minimalist shed.bike. Or set up your own? That's the spirit!)

Subject: A bike shed (any colour will do) on greener grass...
From: Poul-Henning Kamp <phk@freebsd.org>
Date: Sat, 02 Oct 1999 16:14:10 +0200
Message-ID: <18238.938873650@critter.freebsd.dk>
Sender: phk@critter.freebsd.dk
Bcc: Blind Distribution List; 
MIME-Version: 1.0

[bcc'ed to committers, hackers]
Why Should I Care What Color the Bikeshed Is?

From freebsd.org/doc/en_US.ISO8859-1/books/fag/misc.html#bikeshed-painting; see also phk.freebsd.dk/sagas/bikeshed.html for historical background.

"The really, really short answer is that you should not. The somewhat longer answer is that just because you are capable of building a bikeshed does not mean you should stop others from building one just because you do not like the color they plan to paint it. This is a metaphor indicating that you need not argue about every little feature just because you know enough to do so. Some people have commented that the amount of noise generated by a change is inversely proportional to the complexity of the change."

(If you don't like the way we painted this bikeshed, try bikeshedding.io, or the minimalist shed bike. Or set up your own! That's the spirit!)

Subject: A bike shed (any colour will do) on greener grass...
From: Poul-Henning Kamp <phk@freebsd.org>
Date: Sat, 02 Oct 1999 16:14:10 +0200
Message-ID: <18238.938076650@critter.freebsd.dk>
Sender: phk@critter.freebsd.dk
Bcc: Blind Distribution List: ;
MIME-Version: 1.0

[bcc'd to committers, hackers]
Why Should I Care What Color the Bikeshed Is?

From freebsd.org/doc/en_US.ISO8859-1/books/mlsc.html#bikeshed-painting; see also phk.freebsd.dk/sagas/bikeshed.html for historical background.

"The really, really short answer is that you should not. The somewhat longer answer is that just because you are capable of building a bikeshed does not mean you should stop others from building one just because you do not like the color they plan to paint it. This is a metaphor indicating that you need not argue about every little feature just because you know enough to do so. Some people have commented that the amount of noise generated by a change is inversely proportional to the complexity of the change."

(If you don't like the way we painted this bikeshed, try bikeshedding.to, or the minimalist shedbike. Or set up your own? That's the spirit!)
FP vs. OOP
axes of antipathy
dynamic vs. static
threads vs. processes
vertical scaling vs. horizontal scaling
synchronous vs. asynchronous
convention vs. configuration
easy vs. simple
we could go on...
what’s to be done?
FP vs. OOP
FP’ers say...
FP’ers say:

OOP is overcomplicated: dozens of classes for the simplest task
FP’ers say:

OOP means
a big ball of mud
FP’ers say:

It’s impossible to use OOP well
FP’ers say:

FP is about making systems easier to understand
FP’ers say:

FP is for everybody
OOP’ers say...
OOP’ers say:

FP is concrete dependencies everywhere, needlessly convoluted, callback hell
FP is concrete dependencies everywhere, needlessly convoluted, callback hell
OOP’ers say:

FP is concrete dependencies everywhere, needlessly convoluted, callback hell
OOP’ers say:

You have to be really smart to use FP well
OOP’ers say:

OOP is about making systems easier to understand
OOP’ers say:

OOP is for everybody
whew
definitions
OOP
classes and objects
inheritance, polymorphism, and encapsulation
"OOP to me means only messaging, local retention and protection and hiding of state-process, and extreme late-binding of all things."

- Alan Kay
"OOP to me means only messaging, local retention and protection and hiding of state-process, and extreme late-binding of all things."

- Alan Kay
"OOP to me means only messaging, local retention and protection and hiding of state-process, and extreme late-binding of all things."
- Alan Kay
"OOP to me means only messaging, local retention and protection and hiding of state-process, and extreme late-binding of all things."

- Alan Kay
programming with mathematical functions
first-class functions
avoiding side causes
avoiding side effects
immutability
languages
features
FP features
first-class functions
first-class functions
laziness
laziness
persistent data structures / immutability
persistent data structures / immutability
OOP features
classes
namespacing
namespacing
data containment
data containment
messaging
messaging
polymorphism
polymorphism
encapsulation
encapsulation
state management
state management
inheritance
inheritance
inheritance
unique features
pattern matching
reified effects?
tail recursion optimization
1-letter variable names?
1-letter variable names?
classes and objects
3000-line files?
3000-line files?
features?
constraints
avoid side effects
avoid side causes
avoid doing all the work
avoid exposing internals
avoid concrete dependencies
conclusions
emphasis
messaging
encapsulation
polymorphism
explicit causes & explicit effects
FP and OOP are mostly orthogonal
plurality
extrapolating
precision
charity
Parkinson’s Law of Triviality: https://www.farnamstreetblog.com/2013/12/parkinsons-law/

How the term “bikeshed” got into software: http://bikeshed.com

Alan Kay’s definition of OOP: http://www.purl.org/stefan_ram/pub/doc_kay_oop_en


Gamma/Helm/Johnson/Vlissides (Gang of Four), *Design Patterns*: [http://a.co/6OdEq4h](http://a.co/6OdEq4h)

Chris Okasaki, *Purely Functional Data Structures*: [http://a.co/isBRerv](http://a.co/isBRerv)


FP vs. OOP: Beyond the Bikeshed

Colin Jones
@trptcolin
https://8thlight.com