Pragmatic Functional Programming in Java

Grzegorz Piwowarek

@pivovarit

Happy families are all alike; every unhappy family is unhappy in its own way.

—Tolstoy: Anna Karenina

What's Object-Oriented Programming?

OOP Pillars:

- Encapsulation
- Inheritance
- Polymorphism
- Abstraction

What's Functional Programming?

FP Pillars:

- Immutability
- Purity (Referential Transparency)
- Type-Driven Development
- Declarative Programming
- Function Composition
- Equational Reasoning

Immutability

It's impossible to change object's state after its creation

- Simplified Reasoning
- Thread Safety
- Reduced Number of Moving Parts and Invalid States

Purity

Functions/Methods are pure when they have no dependencies on internal/external state, and don't do any side-effects

- Simplified Reasoning
- Simplified Testing
- Memoizable (cacheable)

Type-Driven Development

A strong type system can not only prevent errors, but also guide you and provide feedback in your design process.

- Reduced Number of Moving Parts and Invalid States
- Self-discoverable APIs
- Fewer bugs

Function Composition

- Low-scale modularity

Equational Reasoning

- The power of substitution

OOP/FP

Do these really go against each other?

What's wrong with Java?

- Omnipresent mutability
- Omnipresent side-effects
- Spurious control flows (exceptions)
- Magic frameworks abusing reflection, proxies, classpath scanning, implicit configurations

Judging by its name and signature, what do you think that method does?

```
List<String> transform(List<String> list);
```

```
List<String> transform(List<String> list) {
    list.add(this.state.get(0));
    list.add(SOME_CONSTANT);

    this.orderPizza();

    if (MOON.getLunarPhase() == FULL) list.remove(7);
    this.moreState.addAll(list);
    OtherClass.mutableStaticField = list;
    return list;
}
```

A dirty and imperative program

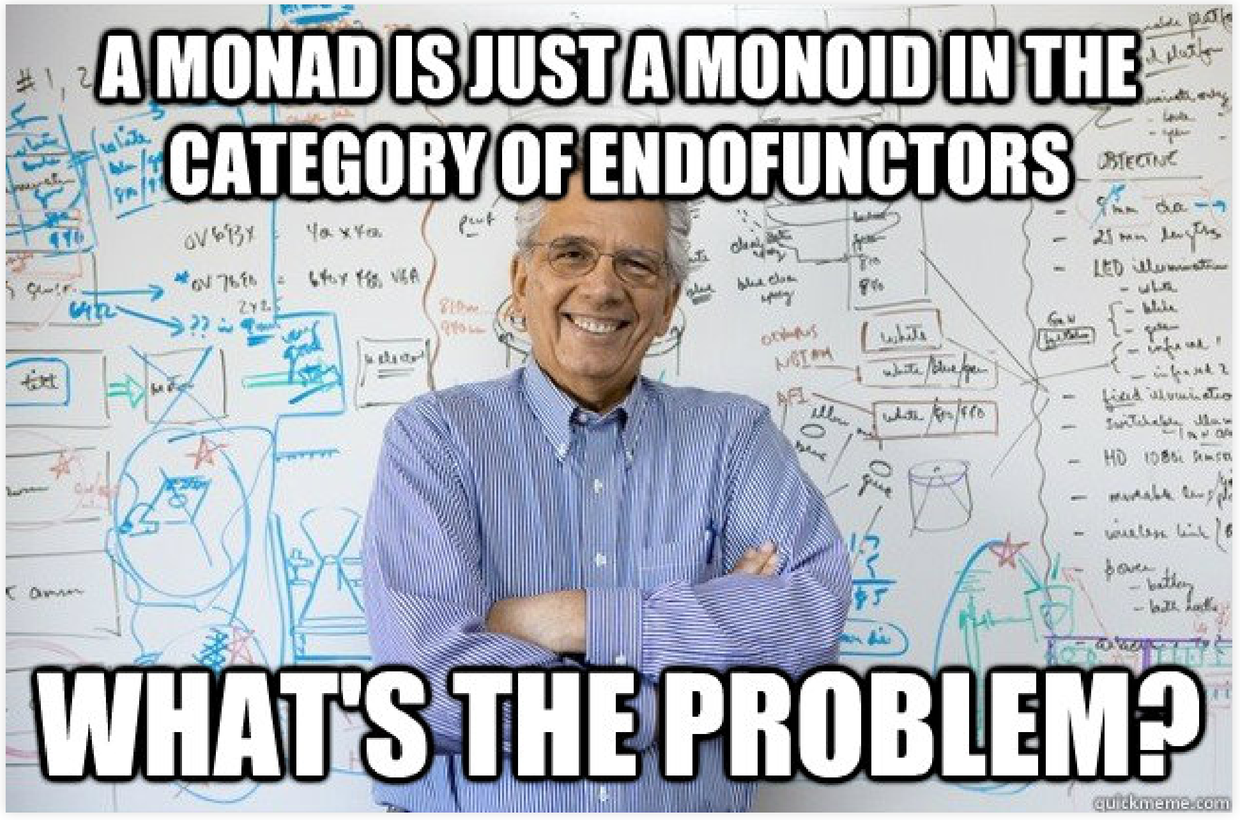
```
println("hello!")
callSomeExternalService()

program.unsafeRunSync()
```

A ”purely functional” declarative program

```
val program: IO[Unit] = for {
  _ <- IO { println("hello!") }
  _ <- IO { callSomeExternalService }
} yield ()

program.unsafeRunSync()
```


A MONAD IS JUST A MONOID IN THE CATEGORY OF ENDOFUNCTORS
WHAT'S THE PROBLEM?
quickmeme.com

```
//TODO: insert some livecoding here
```