Functional Programming in the Wild

Functional Miners

21.05.2019

Monoid

Algebraic structure with a single associative binary operation and an identity element

$(M, e, \odot)$: $\mathrm{ob}(M) = \{\cdot\}$, $\mathrm{hom}(M) = M$, $\circ = \odot$

Monad

A *monad* in *X* is just a **monoid** in the category of *endofunctors* of *X*, with product × replaced by composition of endofunctors and unit set by the identity endofunctor

Thank you!

Questions?

https://commons.wikimedia.org/wiki/File:ET_.jpg

Functional Programming in the Wild

Functional Miners

21.05.2019

~ # whoami

(afronski)

- ✓ *Co-founder* and *Cloud Architect* at [Pattern Match]

  Erlang, ***Elixir***, F#, JavaScript, Python DevOps, **AWS** (3/5 *certificates*)

- ✓ Co-organizer of ***Functional Miners*** and ***Natywna Chmura***

PATTERN MATCH

Why **FP**?

Why **FP**?

Unique Selling Propositions

~~Function as a First-class Citizen~~

Polyglot Programming

Learn at least **one new language every year.**

Different languages solve the same
problems in different ways.

By learning several different approaches, you can help
broaden your thinking and avoid getter struck in a rut.

Andy Hunt, Dave Thomas
(The Pragmatic Programmer)

Declarativeness

```
// Imperative:
for (let i = 0; i < 10; ++i) {
    if (i % 2 === 0) {
      action(array[i]);
    }
}

// Declarative:
array.filter((_v, i) => i % 2).forEach(action);
```

```
// Imperative:
for (let i = 0; i < 10; i += 2) {
  action(array[i]);
}

// Declarative:
array.filter(onlyEvenIndex).forEach(action);
```

```
// Declarative (pattern matching):
<<A:32, B:32, C:32>> = crypto:rand_bytes(12),
random:seed({A,B,C})

// Declarative with macros and compile time generation:
```

elixir/unicode/unicode.ex

Correctness

Immutability

Referential Transparency

Advanced Type Systems

Concurrency

Myths

OOP is a standard

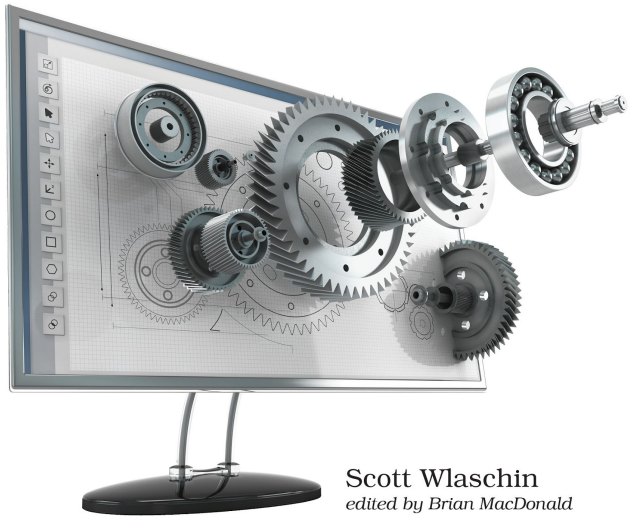
Alan Kay’s Definition of Object Oriented

1. Everything Is An Object.
2. Objects communicate by sending and receiving messages (in terms of objects).
3. Objects have their own memory (in terms of objects).
4. Every object is an instance of a class (which must be an object).
5. The class holds the shared behavior for its instances
   (in the form of objects in a program list)
6. To eval a program list, control is passed to the first object and
   the remainder is treated as its message.

The Pragmatic Programmers

Domain Modeling Made Functional

Tackle Software Complexity with Domain-Driven Design and F#



Scott Wlaschin

edited by Brian MacDonald

Ode to Erlang!

ERLANG

https://pragprog.com/book/swddf/domain-modeling-made-functional

SOLID

FP is hard

```
// Imperative:
for (let i = 0; i < 10; i += 2) {
  action(array[i]);
}

// Declarative:
array.filter(onlyEvenIndex).forEach(action);
```

FP is slow, ...


ERLANG


OCaml

FP is not pragmatic

https://commons.wikimedia.org/wiki/File:ET_.jpg

Start?

1. At first, pick your favorite language and apply concepts from functional paradigm to small pieces of your daily work.
    a. Gradually increase the difficulty level.
    b. Be patient. **It will be difficult at the beginning**.
2. Pick a functional language and learn syntax and functional thinking on small examples.
    a. What language? *It is a secondary choice*.
    b. *Exercism.io*, *HackerRank*, ...
3. Build small projects in functional language.
    a. Brainfuck - *interpreter* and *compiler*.
    b. RFC 862 - *Echo Protocol*.
    c. URL shortener.

Do not stop at first obstacle!

Look and ask for help.

KatoTech Slack - #functional_miners

Thank you!

Questions?

1. **Our company** - **Pattern Match** and my talks.
2. **Natywna Chmura** (facebook, website).
3. **Functional Miners** (facebook, twitter, github, email, website).
4. Alan Kay's Definition Of Object Oriented.
5. Ode to Erlang.
6. **Book**: The Pragmatic Programmer.
7. **Book**: Domain Modelling Made Functional (*F#*).
8. **eBook**: Learn You A Haskell For Great Good!
9. **eBook**: Learn You Some Erlang For Great Good!
10. KatoTech Slack - #functional_miners.

References

PATTERN MATCH