Yeti

ML

JVM

ML + JVM = ❤️

Example

```
ack m n =
    if m == 0 then
        n + 1
    elif n == 0 then
        ack (m - 1) 1
    else
        ack (m - 1) (ack m (n - 1))
    fi;

println "ack 3 8 = \(ack 3 8)"
```

Structural typing

```
brian = {
    name = "Brian McKenna",
    age = 21
}; // is {age is number, name is string}

pet = {
    name = "Monkey",
    type = "Bird"
}; // is {name is string, type is string}

printName o = println o.name; // is {.name is 'a} -> ()

printName brian; // Brian McKenna
printName pet // Monkey
```

Variant types

```
isLeft e = case e of
    Left _: true;
    Right _: false
esac; // is Left 'a | Right 'b -> boolean
```

```
println (isLeft (Left 100)); // true

swap e = case e of
    Left l: Right l;
    Right r: Left r
esac; // is Left 'a | Right 'b -> Left 'b | Right 'a

println (swap (Right "a")) // Left "a"
```

JVM Interop

```
module Interop;

import HelloClojure;

class HelloYeti
    static void printHello()
        HelloClojure#printHello()
end
```

Tools

- Yeb
- Yebspec
- Yetidoc
- Yeti-Maven-Plugin
- Yeti modes for Emacs, Notepad++, Vim, Netbeans

Future

- Built-in doc generator
- Macros
- Use typedefs to shorten errors

Summary

- ML - Nice & easy
- JVM - Very interoperable
- Feels dynamic - Inferred, structural, variant

Thanks

- Yeti
  - http://mth.github.com/yeti/
  - http://github.com/mth/yeti
- Me
  - brian@brianmckenna.org
  - http://twitter.com/puffnfresh