COMP3141

Software System Design and Implementation

Effects and IO Monad Practice

Curtis Millar

CSE, UNSW (and Data61)
11 July 2020

The IO Type

A procedure that performs some side effects, returning a result of type a is written as
`IO a`

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

The IO Type

A procedure that performs some side effects, returning a result of type a is written as
```
IO a
```

World interpretation

IO a is an abstract type. But we can think of it as a function:

```
RealWorld -> (RealWorld, a)
```

(that's how it's implemented in GHC)

The IO Type

A procedure that performs some side effects, returning a result of type a is written as
IO a.

World interpretation

IO a is an abstract type. But we can think of it as a function:

RealWorld -> (RealWorld, a)

(that's how it's implemented in GHC)

```
(>>=) :: IO a -> (a -> IO b) -> IO b
pure  :: a -> IO a

getChar :: IO Char
readLine :: IO String
putStrLn :: String -> IO ()
```

Two-player Tic-Tac-Toe

Assignment Project Exam Help

Example

Simple two-player Tic-Tac-Toe game

https://powcoder.com

Done in editor

Add WeChat powcoder

State Monads

Assignment Project Exam Help

```
newtype State s a = State (s -> (s, a))
```

https://powcoder.com

Add WeChat powcoder

State Monads

Assignment Project Exam Help

```haskell
newtype State s a = State (s -> (s, a))
```

State Monad

```haskell
get :: State s s        https://powcoder.com
put :: s -> State s ()
modify :: (s -> s) -> State s ()
```

Add WeChat powcoder

State Monads

```haskell
newtype State s a = State (s -> (s, a))
```

State Monad

```haskell
get :: State s s
put :: s -> State s ()
modify :: (s -> s) -> State s ()
```

Here we use a monadic interface to simplify the passing of our state around, so that we don't need to manually plumb data around.

Tic-Tac-Toe A.I

Assignment Project Exam Help

> **Example**
>
> Adding A.Is for Tic-Tac-Toe
>
> Done in editor

https://powcoder.com

Add WeChat powcoder

QuickChecking Monads

Assignment Project Exam Help

QuickCheck lets us test `IO` (and `ST`) using this special property monad interface:

```
monadicIO :: PropertyM IO () -> Property
pre       :: Bool -> PropertyM IO ()
assert    :: Bool -> PropertyM IO ()
run       :: IO a -> PropertyM IO a
```

https://powcoder.com

Add WeChat powcoder

Testing a Tic-Tac-Toe A.I

Assignment Project Exam Help

> **Example**
>
> Testing A.Is for Tic-Tac-Toe
>
> https://powcoder.com
>
> Done in editor

Add WeChat powcoder

Homework

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Homework

Assignment Project Exam Help

1. Next week is flexibility week

https://powcoder.com

Add WeChat powcoder

Homework

1. Next week is flexibility week
2. Last week's quiz is due on Friday. Make sure you submit your answers.

Homework

Assignment Project Exam Help

1. Next week is flexibility week
2. Last week's quiz is due on Friday. Make sure you submit your answers.
3. The fourth programming exercise is due by the start if my next lecture (in 14 days).

https://powcoder.com

Add WeChat powcoder

Homework

1. Next week is flexibility week
2. Last week's quiz is due on Friday. Make sure you submit your answers.
3. The fourth programming exercise is due by the start if my next lecture (in 14 days).
4. This week's quiz is also up; it's due Friday week (in 16 days).

Consultations

Assignment Project Exam Help

- Poll on Piazza to register interest. Will not run if there are no votes.

https://powcoder.com

Add WeChat powcoder

Consultations

- Poll on Piazza to register interest. Will not run if there are no votes.
- Tomorrow, 9am to 11am on Blackboard Collaborate.

Consultations

- Poll on Piazza to register interest. Will not run if there are no votes.
- Tomorrow, 9am to 11am on Blackboard Collaborate.
- Link on course website & Piazza.

Consultations

- Poll on Piazza to register interest. Will not run if there are no votes.
- Tomorrow, 9am to 11am on Blackboard Collaborate.
- Link on course website & Piazza.
- Make sure to join the queue on Hopper. Be ready to share your screen with REPL (`ghci` or `stack repl`) and editor set up.