Группа 2225

и Функция-полукровка

Завьялов А.А.

5 декабря 2022 г.

Кафедра систем информатики ФИТ НГУ

Проблемы с вводом-выводом

- Все функции в Haskell – чистые
- Хотим работать с вводом-выводом (потенциально "грязными операциями)
- `getCharFromConsole :: Char` должна **всегда** возвращать одно и то же

Проблемы с вводом-выводом

- Все функции в Haskell – чистые
- Хотим работать с вводом-выводом (потенциально "грязными операциями)
- `getCharFromConsole :: Char` должна **всегда** возвращать одно и то же
- `getCharFromConsole :: RealWorld -> (RealWorld, Char)` всегда даёт одинаковый ответ при одинаковом состоянии `RealWorld`
- Окружающий мир непрерывно изменяет свое состояние
- Невозможно или тяжело отследить и локализовать изменения
- Нужно ограничивать доступ к `RealWorld`

Проблемы с вводом-выводом

- Все функции в Haskell – чистые
- Хотим работать с вводом-выводом (потенциально "грязными операциями)
- `getCharFromConsole :: Char` должна **всегда** возвращать одно и то же
- `getCharFromConsole :: RealWorld -> (RealWorld, Char)` всегда даёт одинаковый ответ при одинаковом состоянии `RealWorld`
- Окружающий мир непрерывно изменяет свое состояние
- Невозможно или тяжело отследить и локализовать изменения
- Нужно ограничивать доступ к `RealWorld`
- `getChar :: IO Char`.

Проблемы с вводом-выводом

- Все функции в Haskell – чистые
- Хотим работать с вводом-выводом (потенциально "грязными операциями)
- `getCharFromConsole :: Char` должна **всегда** возвращать одно и то же
- `getCharFromConsole :: RealWorld -> (RealWorld, Char)` всегда даёт одинаковый ответ при одинаковом состоянии `RealWorld`
- Окружающий мир непрерывно изменяет свое состояние
- Невозможно или тяжело отследить и локализовать изменения
- Нужно ограничивать доступ к `RealWorld`
- `getChar :: IO Char`. `IO`?

Что такое `IO`?

- Тип `IO a` говорит нам – "во время вычисления значения типа `a` может случиться операция ввода-вывода"
- `IO` приблизительно определяется как:
  ```
  newtype IO a = IO (RealWorld -> (RealWorld, a))
  ```
- Документация говорит: "RealWorld is **deeply magical**"
- Значениями `RealWorld` нельзя манипулировать, он существует только на уровне системы типов

Что такое `IO`?

- Тип `IO a` говорит нам – "во время вычисления значения типа `a` может случиться операция ввода-вывода"
- `IO` приблизительно определяется как:
  ```
  newtype IO a = IO (RealWorld -> (RealWorld, a))
  ```
- Документация говорит: "RealWorld is **deeply magical**"
- Значениями `RealWorld` нельзя манипулировать, он существует только на уровне системы типов
- Говорят, что выражения с `IO a` это не функции

Что такое `IO`?

- Тип `IO a` говорит нам – "во время вычисления значения типа `a` может случиться операция ввода-вывода"
- `IO` приблизительно определяется как:
  ```
  newtype IO a = IO (RealWorld -> (RealWorld, a))
  ```
- Документация говорит: "RealWorld is **deeply magical**"
- Значениями `RealWorld` нельзя манипулировать, он существует только на уровне системы типов
- Говорят, что выражения с `IO a` это не функции, а **действия ввода вывода** (IO actions)

Монада IO

IO можно примерно определить как

```
instance Monad IO where
  return x = IO $ \w -> (w, x)
  m >>= k = IO $ \w ->
    case m w of
      (w', a) -> k a w'
```

Что важно

- Нельзя достать значения из IO
- Побочный эффект (операция ввода-вывода) действия происходит один раз
- Побочные эффекты происходят в заданном порядке

Основные функции консольного ввода-вывода

- Ввод
  - `getChar :: IO Char`
  - `getLine :: IO String`
  - `getContents :: IO String`
- Вывод
  - `putChar :: Char -> IO ()`
  - `putStr, putStrLn :: String -> IO ()`
  - `print :: Show a => a -> IO ()`
- Ввод-вывод
  - `interact :: (String -> String) -> IO ()`

Модуль `System.Random`

- Вообще-то, генераторы псевдослучайных чисел не зависят от *окружающего мира*

Модуль `System.Random`

- Вообще-то, генераторы псевдослучайных чисел не зависят от *окружающего мира*
- Но могут...

Модуль `System.Random`

- Вообще-то, генераторы псевдослучайных чисел не зависят от *окружающего мира*
- Но могут...
- Это делает *некоторые* генераторы действиями ввода-вывода

Модуль `System.Random`

- Вообще-то, генераторы псевдослучайных чисел не зависят от *окружающего мира*
- Но могут...
- Это делает *некоторые* генераторы действиями ввода-вывода

Установка `System.Random`

```
$ cabal install --lib random
```

Модуль `System.Random`

- Вообще-то, генераторы псевдослучайных чисел не зависят от *окружающего мира*
- Но могут...
- Это делает *некоторые* генераторы действиями ввода-вывода

Установка `System.Random`

```
$ cabal install --lib random
```

Полезные функции

- `randomIO :: (Random a, MonadIO m) => m a`
- `randomRIO :: (Random a,MonadIO m)=>(a,a)->m a`

Полезные функции для работы с монадами

Эти функции определены для **любых** монад, не только `IO`

- ```
  mapM :: (Traversable t, Monad m) =>
  (a -> m b) -> t a -> m (t b)
  ```
- ```
  forM :: (Traversable t, Monad m) =>
  t a -> (a -> m b) -> m (t b)
  ```
- ```
  sequence :: (Traversable t, Monad m)
  => t (m a) -> m (t a)
  ```

Есть версии, возвращающие `m ()`: `mapM_`, `forM_`, `sequence_`
  ```

Тёмные искусства – изменяемые ссылки

- Не повторяйте этого дома
- `import Data.IORef (newIORef,readIORef,writeIORef)`

Темнейшее волшебство

- Не стоит вам этим пользоваться. Никогда.

Темнейшее волшебство

- Не стоит вам этим пользоваться. Никогда.
- `import System.IO.Unsafe`

Q&A