

CS314. Функциональное программирование

Направление «Фундаментальная информатика и информационные технологии»
Институт математики, механики и компьютерных наук имени И. И. Воровича
Южный федеральный университет

27 октября 2017 г.

```
{-
```

Написать функцию, которая по заданному списку строк возвращает сумму длин всех строк.

```
-}
```

```
totalLength :: [String] -> Int
```

```
totalLength = foldl (\acc x -> acc + (length x)) 0
```

```
totalLength = sum . map length
```

```
{-
```

Написать функцию, которая по заданному символу и целому числу n строит список строк, содержащих $1, 2, \dots, n$ повторений символа. Функция должна возвращать `Nothing`, если $n=0$.

```
-}
```

```
step :: Char -> Int -> [String] -> [String]
```

```
step c 0 xs = xs
```

```
step c n xs = step c (n - 1) ((take n (repeat c)):[] ++ xs)
```

```
build1 :: Char -> Int -> Maybe [String]
```

```
build1 c 0 = Nothing
```

```
build1 c n = Just (step c n [])
```

- `take n.repeat = replicate n`
- `inits :: [a] -> [[a]]`

```
build1 :: Char -> Int -> Maybe [String]
build1 c 0 = Nothing
build1 c n = Just (tail $ inits $ replicate n c)
```

{-

Написать функцию, аналогичную по возможностям функции `build1`, но возвращающую при этом значение типа `Either String [String]`, в котором значение слева должно свидетельствовать об одной из следующих особых ситуаций:

(*) $n=0$;

(*) $n > 100$;

(*) Роскомнадзор запрещает создавать строки из символа 'x'.

-}

```
build2 :: Char -> Int -> Either String [String]
```

```
build2 c n = if (n == 0) then Left "N is 0" else if (n > 100) then Left "N is too big"
            else if (c == 'x') then Left "Roskomnadzor forbids 'x'" else Right []
```

```
build2 :: Char -> Int -> Either String [String]
build2 _ 0 = Left "N is 0"
build2 c n
  | n > 100 = Left "N is greater than 100"
  | c == 'x' = Left "Roskomnadzor ..."
  | otherwise = Right (tail $ inits $ replicate n c)
```

```
totalLength :: [String] -> Int
totalLength = sum . map length

stringsBuilder :: Int -> Char -> [String]
stringsBuilder n c = tail $ inits $ replicate n c

build1 :: Char -> Int -> Maybe [String]
build1 c 0 = Nothing
build1 c n = Just (stringsBuilder n c)

build2 :: Char -> Int -> Either String [String]
build2 _ 0 = Left "N is 0"
build2 c n
  | n > 100 = Left "N is greater than 100"
  | c == 'x' = Left "Roskomnadzor ..."
  | otherwise = Right (stringsBuilder n c)
```

{-

Параметрами командной строки являются имя файла, символ, целое число.

1) Пользуясь функцией `totalLength` и возможностями `IO` как функтора, подсчитать и вывести общую длину строк, переданных программе в качестве аргументов командной строки.

2) Пользуясь функцией `totalLength` и возможностями `IO` как функтора, подсчитать и вывести общую длину строк, содержащихся в заданном текстовом файле (результат `readFile` должен быть предварительно преобразован к списку строк).

3) Пользуясь функцией `totalLength`, подсчитать общую длину строк для значений в контекстах, сформированных функциями `build1` и `build2` (в решении следует пользоваться возможностями `Maybe` и `Either String` как функторов).

-}


```
main = do
  args <- getArgs
  str <- readFile (head args)
  print $ totalLength args
  print $ totalLength (lines str)
  print $ if (last args == "0")
    then length("Nothing") + totalLength (concat (rights
    else
      totalLength (fromJust (build1 (read $ head (tail
      totalLength (concat (rights ((build2 (read $ head
      totalLength (lefts ((build2 (read $ head (tail ar
```

```
main = do
  l1 <- totalLength <$> getArgs
  print l1
  [fname, c', n'] <- getArgs
  l2 <- totalLength.lines <$> readFile fname
  print l2

let
  c = head c'
  n = read n'
print $ totalLength <$> build1 c n
print $ totalLength <$> build2 c n
```

- Проблема: нужно найти сумму двух результатов.

```
processResults :: Maybe Int -> Either String Int -> IO ()
processResults Nothing _ = print "error: b1 is Nothing"
processResults _ (Left err) = print $ "error: " ++ err
processResults (Just l1) (Right l2) =
    print $ "result: " ++ show (l1 + l2)

main = do
    ...
    let
        b1 = totalLength <$> build1 c n
        b2 = totalLength <$> build2 c n
    processResults b1 b2
```