# Лабораторная работа t-13-2 (вариант 4)

## Условие

Построить парсер для данных из файла в формате:

Корректное задание массива в JavaScript (пустого или из целых чисел)

m1 = new Array()

m2 = []

m3 = [1,2,3,6]

m4 = new Array(2,3,4,5)

m5 = [2, 0, 2, 1]

new Array(3, 4, 1)

## Код

{-

ПИ19-3 Деменчук Георгий

Построить парсер для данных из файла в формате:

Корректное задание массива в JavaScript (пустого или из целых чисел)

-}

-- Делает разделение данных по определенному паттерну

-- Разделение происходит только по 1 элементу паттерна

-- print(splitDataOnPattern "," "1,2,3,4") -> ("1","2,3,4")

splitDataOnPattern :: String -> String -> (String, String)

splitDataOnPattern sub str = let

splitdataOnPatternA' [] s = ([], s)

splitdataOnPatternA' (y:ys) (x:xs)

| x == y = let (xs', xs'') = splitdataOnPatternA' ys xs in (xs', xs'')

| otherwise = let (xs', xs'') = splitdataOnPatternA' sub xs in (x:xs', xs'')

splitdataOnPatternA' \_ \_ = ([], [])

in splitdataOnPatternA' sub str

-- dataMatchItems делает сопоставление данных по определенному паттерну

-- print(dataMatchItems "new Array" "new Array(1,2,3)") -> True

-- print(dataMatchItems "=" "new Array(1,2,3)") -> False

dataMatchItems :: Eq a => [a] -> [a] -> Bool

dataMatchItems [] \_ = True

dataMatchItems (x:xs) (y:ys) = x == y && dataMatchItems xs ys

dataMatchItems \_ \_ = False

-- findArray обрабатывает целиком все, что считано с файла и ищет массивы

findArray :: (Read a) => String -> [[a]]

findArray s = let

m = "new Array("

find [] acc = acc

find c@(x:xs) acc

| dataMatchItems m c = let

(a, t) = splitDataOnPattern ")" (drop (length m) c)

in

find t (acc ++ [read ("[" ++ a ++ "]")])

| x == '[' = let

(a, t) = splitDataOnPattern "]" xs

in

find t (acc ++ [read ("[" ++ a ++ "]")])

| otherwise = find xs acc

in

find s []

-- Точка входа в программу

main :: IO ()

main = do

putStrLn "Введите путь до файла для парсинга:"

inputFilePath <- getLine

print (splitDataOnPattern "." inputFilePath)

let (fName, fExt) = splitDataOnPattern "." inputFilePath

let fileOutPath = fName ++ ".hs"

-- Если не .js

if fExt /= "js"

then error "Файл не поддерживается, нужен .js!"

-- Если js

else do

content <- readFile inputFilePath

let parsedArrays = findArray content

print (parsedArrays :: [[Int]])

putStrLn "\nТест всех функций по отдельности"

print(dataMatchItems "new Array" "new Array(1,2,3)")

print(dataMatchItems "=" "new Array(1,2,3)")

print(splitDataOnPattern "," "1,2,3,4")

## Скриншоты

Выполнение программы



