

Specyfikacje formalne i programy funkcyjne

Zadanie projektowe z języka Haskell

Zima 2020

1 Opis zadania

W ramach zadania należy napisać aplikację rozwiązującą łamigłówkę „Mozaika”. Łamigłówka polega na zamalowywaniu pól planszy według podanych cyfr. Cyfra (0-9) na danym polu oznacza ile sąsiednich pól, włącznie z polem z cyfrą, powinno być zamalowanych. Przykładową łamigłówkę z rozwiązaniem pokazuje rysunek 1:

		5				5	4	
	5			6			5	
4		2		5			4	4
	4					1		
			1			1	3	5
			3			3	6	
	6	7	6		4			
	3			7	7			3
	1	3		8				1
						3		

Rysunek 1: Przykładowa łamigłówka¹

Algorytm poszukiwania rozwiązania łamigłówki jest częścią zadania. Po uruchomieniu aplikacja powinna spytać użytkownika o nazwę pliku wejściowego z łamigłówką. Taki plik zawiera opis łamigłówki w postaci:

```
["pola 1-go wiersza", "pola 2-go wiersza", ..., "pola n-tego wiersza"]
```

Zawartość kolejnych pól wiersza jest przedstawiona napisem złożonym z cyfr (0-9) i symbolu kropki, gdzie kropka oznacza pole puste. Plik wejściowy dla pokazanej wyżej łamigłówki mógłby mieć postać:

```
["..5....54.",  
 ".5..6..5..",  
 "4.2.5...44",  
 ".4....1...",  
 "...1..13.5",  
 "...3..36..",  
 ".676.4....",  
 ".3..77..31",  
 ".13.8...1.",  
 ".....3..."]
```

¹Źródło: <https://www.wydawnictwologi.pl/mozaiki-zasady>

Można założyć poprawność wczytywanego pliku. Nie ma ograniczeń na wielkość planszy. Po znalezieniu rozwiązania, aplikacja powinna je wyświetlić w formie tekstowej na ekranie.

Uwaga: w rozwiązywaniu proszę nie korzystać z innych bibliotek niż Prelude i System.IO.

2 Wczytywanie pliku

Wczytanie pliku z łamigłówką i utworzenie na jego podstawie wartości typu `[String]` pokazuje poniższy przykład:

```
import System.IO (readFile)

-- wczytaj łamigłówkę z pliku o podanej nazwie
readPuzzle :: String -> IO [String]
readPuzzle filename = do
    contents <- readFile filename           -- odczytaj całą zawartość pliku
    let puzzle = read contents :: [String]  -- utwórz listę napisów (zob. klasa typów Read)
    return puzzle

main = do puzzle <- readPuzzle "puzzle.txt"
        putStrLn (show $ length $ puzzle)  -- wyświetl liczbę wierszy łamigłówki
```

3 Termin i forma oddania zadania

Zadanie realizowane jest w **dwuosobowych** zespołach. Źródła programu należy wysłać w formie załącznika na adres m.szlenk@elka.pw.edu.pl do dnia **18 stycznia 2021 r.** (włącznie). Mail powinien być zatytułowany „[SPOP Projekt] Nazwisko1 Nazwisko2”. Dodatkowo należy również dołączyć dokument (.pdf) zawierający jedną (i tylko jedną) stronę opisu do źródeł. Opis ten może mieć charakter pół-formalny (odpowiednik osobistego prezentowania źródeł prowadzącemu). Należy w nim założyć, że czytelnik wie do czego program służy, natomiast chce zorientować się, co i gdzie jest w kodzie. Poziom szczegółowości opisu wyznacza jego maksymalny rozmiar równy jednej stronie.

4 Kryteria oceny

Projekt jest oceniany w skali 0–10 punktów. Przy ocenie będą brane pod uwagę:

- zachowanie programu (czy jest zgodne z oczekiwaniami?);
- zwięzłość i czytelność kodu, w tym jakość komentarzy;
- złożoność czasowa i pamięciowa (czy program jest użyteczny?).