



Zadanie projektowe z języka Haskell

Semestr: lato 2017

Opis zadania

Zadanie realizowane jest w dwuosobowych zespołach. W ramach zadania należy napisać aplikację rozwiązującą łamigłówkę "Plaster miodu". Łamigłówka polega na uzupełnieniu diagramu w kształcie plastra miodu literami od **A** do **G** tak, aby dla każdego pola włącznie z jego otoczeniem żadna z liter od **A** do **G** się nie powtórzyła. Reguła dotyczy również skrajnych pól.

Poniżej znajduje się przykładowa łamigłówka:



Powyższy przykład i inne można znaleźć pod adresem:

www.lamiglowkimix.pl/modules/honeycomb/honeycomb.php.

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:

Plaster ["*poła 1-go wiersza*", "*poła 2-go wiersza*", ..., "*poła n-tego wiersza*"]

Zawartość kolejnych pól wiersza jest przedstawiona napisem złożonym z liter (A-G) i symbolu kropki, gdzie kropka oznacza pole puste. Plik wejściowy dla pokazanej wyżej łamigłówki ma postać:

Plaster ["BD..", ".GA.D", ".FEG", "ABDCF", "E..."]

Dla łamigłówki o n -wierszach, kolejne wiersze (napisy) będą miały długość: $(n-1)$, n , $(n-1)$, ..., $(n-1)$. Należy sprawdzić poprawność wczytywanego pliku. Nie ma ograniczeń na wielkość planszy. Po znalezieniu rozwiązania, aplikacja powinna je wyświetlić w formie tekstowej na ekranie.

Uwaga: w rozwiązaniu wolno korzystać wyłącznie z tych funkcji i typów standardowych, które pojawiły się na wykładzie lub laboratorium.

Kryteria oceny

Projekt jest oceniany w skali 0-20 punktów. Przy ocenie będą brane pod uwagę:

- zakres funkcjonalny programu (czy program udostępnia wymienione w opisie zadania funkcje?),
- zachowanie programu (czy jest zgodne z oczekiwaniami?),
- czytelność i intuicyjność interfejsu,
- zwięzłość i czytelność kodu, w tym jakość komentarzy,
- złożoność czasowa i pamięciowa (czy nie jest "absurdalnie" wysoka?).