

Programowanie funkcyjne — kolokwium nr 2, 22.01.2020

Instrukcja: Rozwiązania należy przesłać do godziny 9:45, w jednym mailu, na adres kolokwium.pf@gmail.com. Każde zadanie należy przesłać w oddzielnym pliku: zadanie1.hs, zadanie2.hs i zadanie3.erl. Plików nie należy zipować. Rozwiązania muszą się poprawnie kompilować. W rozwiązaniach nie można korzystać z modułów innych niż standardowe; niedozwolone jest użycie polecenia import. Zadania 1 i 2 napisać w Haskellu, zadanie 3 — w Erlangu. Rozwiązania nie spełniające powyższych wymogów nie będą oceniane. Punktacja: 10 punktów za każde zadanie. Uwaga: korzystanie z internetu poza wyznaczonym czasem skutkuje automatycznym otrzymaniem 0 punktów.

Zadanie 1. Napisać bezpunktowo funkcję $ce :: [[Int]] \rightarrow [Int]$, która konkatenuje wyłączenie listy o parzystej sumie. Na przykład

$ce \ [[1,2,3],[2,1],[5,1,1],[],[4,1,1]] = [1,2,3,4,1,1]$.

W komentarzach należy przedstawić uzasadnienie rozwiązania. Wskazówka: (a) Należy bezwzględnie pisać sygnatury, aby ustrzec się błędów kompilacji. (b) Przydatny może być operator $\$$ o następującym działaniu $f \$ g \ x = f(g \ x)$.

Zadanie 2. Napisać programik interaktywny (rozpoczynający się wierszem `main = do`), który wczytuje wiersz tekstu i wypisuje dla niego rozkład częstości znaków, i powtarza tę czynność aż do wczytania wiersza zawierającego jedynie kropkę. Rozkład częstości znaków to zbiór par postaci (a, n) , gdzie a oznacza znak, zaś n — liczbę jego wystąpień w wierszu. W rozkładzie należy każdorazowo uwzględnić tylko te znaki, które pojawiają się we wczytanym wierszu.

Zadanie 3. Napisać moduł uruchamiający N procesów, które roboczo numerujemy od 1 do N , przy czym $N \geq 3$. Proces o numerze i odbiera liczbę x , wypisuje na ekran parę (i, x) i przesyła następnemu procesowi zmodyfikowaną liczbę x . Modyfikacja polega na pomnożeniu przez 2 przez procesy o numerach parzystych i odjęciu 1 przez procesy o numerach nieparzystych. Przez następny proces rozumiemy proces o numerze $i \bmod N + 1$.

Moduł ma udostępniać funkcję `start(N, X)`, która uruchomi procesy w stosownym cyklu tak, by całość wypisywała wyniki działania procesów, poczynając od procesu o numerze 1, któremu zostanie przekazana liczba X . Przykładowo, wywołanie `start(5, 3)` powinno spowodować wypisanie następujących liczb: (1, 3), (2, 2), (3, 4), (4, 3), (5, 6), (1, 5), (2, 4), (3, 8), ...