

Programowanie funkcyjne — kolokwium nr 2, 22.01.2019

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.

- (a) Napisać bezpunktowo funkcję $\text{divBy3} :: \text{Int} \rightarrow \text{Bool}$, która sprawdza, czy liczba podana jako argument jest podzielna przez 3.
- (b) Korzystając z poprzedniej funkcji, napisać bezpunktowo funkcję $f :: [[\text{Int}]] \rightarrow \text{Int}$, która zwraca liczbę list, zawierających podzielną przez 3 liczbę liczb podzielnych przez 3. Na przykład $f [[1,3,7,9],[3,3,7,9],[1,2,4]] = 2$, bo tylko listy $[3,3,7,9]$ oraz $[1,2,4]$ mają podzielne przez 3 liczby elementów podzielnych przez 3 (pierwsza 3, druga 0).

W punkcie (b) należy koniecznie przedstawić kolejne kroki dochodzenia do rozwiązania. Uwaga: proszę pamiętać, że „f.g x” to nie to samo, co „(f.g) x”.

Zadanie 2. Napisać programik interaktywny (rozpoczynający się wierszem `main = do`), który wczytuje wiersz tekstu i wypisuje go w odwrotnej kolejności, i powtarza tę czynność aż do wczytania wiersza zawierającego jedynie kropkę.

Zadanie 3. Napisać moduł, który uruchomi trzy procesy: p_1 , p_2 i p_3 . Każdy z procesów ma odbierać liczbę, wypisywać ją na ekran, wykonywać na niej pewną operację i przysyłać następnemu procesowi zgodnie ze schematem: $p_1 \rightarrow p_2$, $p_2 \rightarrow p_3$ oraz $p_3 \rightarrow p_1$. Operacje wykonywane przez procesy p_1 , p_2 i p_3 to odpowiednio: dodanie 1, pomnożenie przez 2 i odjęcie 3. Moduł ma udostępniać funkcję $\text{start}(N)$, która uruchomi procesy w stosownym cyklu tak, by całość wypisywała wyniki działania procesów, poczynając od liczby N . Przykładowo, wywołanie $\text{start}(5)$ powinno spowodować wypisanie następujących liczb: 5, 6, 12, 9, 10, 20, 17, ...