

Paradygmaty programowania laboratorium

Lista 1 do wykładu 2

18 października 2018

!! Dozwolone jest tylko wykorzystywanie struktur omawianych na wykładzie

Zadanie 1 (Scala 1 pkt, OCaml 1 pkt)

Napisz funkcję wykorzystując rekursję ogonową, która będzie generowała listę liczb o zadanej na wejściu długości (nie wolno korzystać z gotowych funkcji dostarczonych w ramach bibliotek dla danego języka).

np. `makeList 100000 [] => [1, 2, 3,, 100000]`

funkcja testująca wygląda następująco (przykład w OCaml):

`let testMakeList f = assert(f = [1;2;3;4;5;6;7;8;9;10]);;`

Napisana funkcja ma przechodzić powyższy test.

Zadanie 2 (Scala 1 pkt, OCaml 1 pkt)

Napisz funkcję odwracającą zadaną listę od zadanego miejsca. Wykorzystaj w tym zadanie dopasowanie do wzorca oraz rekursję ogonową. Napisz funkcje testujące. Nie wolno korzystać z gotowych funkcji dostarczonych w ramach bibliotek dla danego języka.

np. `reverseFrom [1;2;3;4] 2 => [1;2;4;3]`

Zadanie 3 (Scala 3 pkt, OCaml 3 pkt)

Napisz funkcję, która wygeneruje funkcję (factory) grupującą wejściową listę liczb według podzielności przez zadane dwie liczby przekazane w parametrach funkcji fabrykującej. Należy użyć w tym zadaniu dopasowania do wzorca, rekursji ogonowej, rozwiniętej postaci funkcji oraz funkcji anonimowych. UWAGA!: każda z liczb w liście wejściowej może zostać tylko raz sprawdzona. Napisz funkcję testującą. Nie wolno korzystać z gotowych funkcji dostarczonych w ramach bibliotek dla danego języka.

(OCaml) np.

`let groupingFunction = generateGroupingFunctionFactory 3 5;;
groupingFunction [1;2;3;4;5;6;7;8;9;10] => ([3;6;9], [5;10], [1;2;4;7;8]);;`