

Lista 5

Ewaluacja leniwa

W poniższych zadaniach **nie wolno** wykorzystywać funkcji wbudowanych obliczających **długość listy**, **odwracających listę** oraz **łączących dwie listy**.

Każde zadanie, poza implementacją funkcji, musi posiadać **kompletny zestaw testów**.

Do wykonania zadań należy wykorzystać mechanizmy poznane na wykładzie nr 5.

- 1) Wykorzystując funkcje typu: *unit* -> 'a zamodeluj poniższą sytuację:
„Na stole ustawiono w rzędzie losy na loterię. Każdy z losów można odkryć, zyskując w ten sposób ukrytą pod nim nagrodę. Próba odkrycia odkrytego losu nie zmienia stanu na stole.”

Potraktuj stół jako zwykłą listę zawierającą losy. Zdefiniuj funkcję *buyTicket*, która dla danego stołu oraz numeru losu dokonuje jego odsłonięcia (zwracając nową listę). (OCaml) (10 pkt.)

Przykładowo:

[?; ?; ?; ?; ?] ---- 3 ----> [?; ?; „Komputer”; ?; ?], gdzie „?” oznacza nieodkryty los.

Zwróć uwagę, że powyższy przykład nie przedstawia dokładnie tego, co interpreter będzie wyświetlał na ekranie.

- 2) Zdefiniuj funkcję *sublistsL*, która dla danej listy leniwej zwraca nową listę leniwą zawierającą kolejne podlisty (typu List) listy wejściowej. (Scala) (10 pkt.)

Przykładowo:

[x₁; x₂; x₃; x₄; ...] -----> [[x₁]; [x₁; x₂]; [x₁; x₂; x₃]; ...]