

Lista 7

OCaml: Moduły, funktory

Zapoznaj się z ideą modułów i funktorów w języku OCaml.

Moduły ([doc](#))

1. Punkt w przestrzeni trójwymiarowej:
 - a. Utwórz sygnaturę modułu reprezentującego punkt w przestrzeni trójwymiarowej.
 - b. Moduł powinien enkapsulować typ rekordowy reprezentujący punkt, funkcję tworzącą punkt oraz funkcję obliczającą odległość między dwoma punktami.
 - c. Utwórz strukturę tego modułu.
2. Odcinek
 - a. Analogicznie, wykonaj moduł reprezentujący odcinek, który enkapsuluje dwie instancje punktu.
 - b. Moduł powinien zawierać funkcję zwracającą długość odcinka.
3. Drzewo binarne:
 - a. Utwórz moduł, który enkapsuluje drzewo binarne oparte o wybrany sposób reprezentacji (np. listy Cons/Nil).
 - b. Zaimplementuj operacje:
 - i. Dodawania węzła do drzewa,
 - ii. Usuwania węzła z drzewa,
 - iii. Przechodzenia po drzewie w preorder/inorder/postorder,
 - iv. Zwracania listy liści drzewa.

Funktory ([doc](#))

4. Zadbaj o możliwość abstrakcji nad typem reprezentującego współrzędną (np. int, float). Czy jest taka możliwość? Rozważ implementację punktu poprzez utworzenie modułu (sygnatury) enkapsulującego typ współrzędnej a następnie wykonanie funktora `MakePoint`, który odwzorowuje moduł reprezentujący współrzędną w moduł reprezentujący punkt.
5. Utwórz moduł `Translation`, który enkapsuluje wartości przesunięcia w konkretnych wymiarach.

6. Następnie skonstruuj funktory `Translate_Point` oraz `Translate_Segment`, które realizują morfizm pomiędzy modułami reprezentującymi operację [geometrycznej translacji](#).
 - a. Modułem wejściowym dla funktorów będzie punkt/odcinek przed przesunięciem, a modułem wyjściowym punkt/odcinek po przesunięciu.