

Lista 7

OCaml: Moduły, funktory

Moduły ([doc](#))

1. Korzystając z opracowanej implementacji *punktu* w przestrzeni trójwymiarowej o współrzędnych x, y, z , skonstruuj sygnaturę oraz strukturę modułu reprezentującego punkt. Zadbaj o to, aby moduł dokonywał enkapsulacji swojej wartości, a także funkcji odległości. Postaraj się zadbać o możliwość polimorfizacji typu reprezentującego współrzędną, (np. Int, Float).
2. Analogicznie, utwórz moduł reprezentujący *odcinek*, który dokona enkapsulacji dwóch punktów. Dodatkowo, niech moduł zawiera funkcję zwracającą długość odcinka.
3. Z wykorzystaniem modułów skonstruuj w języku OCaml sygnaturę oraz strukturę modułu, który enkapsuluje drzewo binarne oparte o wybrany sposób reprezentacji (np. listy Cons/Nil) wraz z operacjami:
 - a. dodania węzła do drzewa,
 - b. usunięcia węzła z drzewa,
 - c. przejścia po drzewie preorder/inorder lub postorder,
 - d. zwrócenia listy liści drzewa.

Funktory ([doc](#))

4. W języku OCaml skonstruuj funktory, które będą realizowały opisane poniżej przekształcenia:
 - a. funktor `Make_Point` tworzący punkt danego typu (np. Int, Float),
 - b. funktor `Make_Segment` dokonujący odwzorowania pomiędzy dwoma punktami a odcinkiem, którego te punkty stanowią krańce,
 - c. `Translate_Point` oraz `Translate_Segment` które łącznie posłużą do realizacji morfizmu pomiędzy modułami reprezentującego operację [geometrycznej translacji](#), gdzie modułem wejściowym będzie punkt/odcinek przed przesunięciem, a modułem wyjściowym punkt/odcinek po przesunięciu. Aby reprezentować przesunięcie, należy wykonać moduł `Translation`, który może dokonywać enkapsulacji wartości przesunięcia w konkretnych wymiarach.

