

## Przegląd kodu

Projekt zespołu w składzie: Grzegorz Frejek oraz Kacper Janda  
umieszczony pod adresem <https://github.com/Kacpro/Haskell-projekt>

## Wrażenia ogólne

Kod napisany czytelnie, opisy funkcji najczęściej zwięzłe i jasne, przypadki graniczne obsługane (np. nie można pomnożyć dwóch macierzy o złych wymiarach), wykorzystywane funkcje pomocnicze.

Duża ilość testów, zarówno HUnit, jak i QuickCheck, większość testów nawet udokumentowana. Stworzona instancja QuickCheck.Arbitrary dla własnego typu BufMatrix

## Uwagi

- Polecenie ``stack build`` rzuca błąd przy imporcie modułu Matrix (nazwa pliku małą literą zamiast wielką)
- Funkcja ``inverse`` zwraca niepoprawny wynik dla macierzy osobliwych, podczas gdy powinna rzucić błędem
- Moduł nazywa się "Tree", natomiast przedstawia drzewo BST a nie dowolne drzewo, jak sugerowałaby nazwa, brak też informacji o tym w dokumentacji
  - funkcja `treeContainsValue` opiera się na wyszukiwaniu w drzewie BST
- W dokumentacji opis funkcji `flattenTreeInOrder` oraz `flattenTreePostOrder` jest taki sam: "Returns list that is representation of given Tree's elements but in different order"
- Funkcja `flattenTreeInOrder` wykonuje na drzewie preorder a nie inorder jak mówi nazwa, funkcję inorder realizuje natomiast funkcja `flattenTree`
- Funkcje pomocnicze są udokumentowane (wydaje mi się, że niepotrzebnie)
- Moduł Main nie robi nic poza importem wszystkiego i wypisaniem "1" na wyjście
- Brak testów QuickCheck dla modułu Tree, jednak obecne testy HUnit są bardzo rozbudowane

## Podsumowanie

Kod, mimo kilku uwag, napisany poprawnie i czytelnie, funkcje są udokumentowane, testy obecne. Projekt podzielony na moduły, użyte narzędzie Stack oraz system kontroli wersji.