

# Recenzja projektu

## “Data-structures”

### Zespół projektowy:

Agnieszka Pulnar

Jakub Posłuszny

<https://github.com/pulnara/data-structures>

### Zespół recenzujący:

Tomasz Michalski

Szymon Lichoń

## 1. Program

Program spełnia swoje założenia jeśli chodzi o poprawną implementację struktur danych w języku Haskell.

## 2. Kod

### MyList.hs

Nie zauważyliśmy żadnych błędów w konstrukcji funkcji - są dość krótkie i dobrej kolejności obsługują kolejne przypadki wartości argumentów. Jako plus należy zaliczyć komentarz przy jednej z funkcji, dotyczący tego, jak można inaczej ją zrealizować.

### BinTree.hs

Dziwnym rozwiązaniem wydaje nam się zastosowana konwencja nazewnictwa modułów - wg nas lepszym pomysłem byłaby zmiana nazwy modułu BinTree na np. BinTree.<nazwa>, co odpowiadałoby modułowi gdzie zostały wydzielone poboczne funkcje (BinTree.Internal). Samo rozdzielenie obu funkcji do dwóch modułów też wydaje się niejasnym rozwiązaniem - nie rozumiemy do końca kryteriów podziału. Technicznie funkcje napisane są jednak poprawnie i obejmują przypadki dla różnych wartości argumentów.

### AVLTree.hs

Zauważyliśmy, że w funkcji rotate wielokrotnie wywoływane są te same funkcje z tymi samymi argumentami - dobrym pomysłem byłoby wydzielenie ich do bloku where i przypisanie do zmiennych. Uważamy że zwiększyłoby to czytelność kodu i jego szybkość. Podoba nam się utrzymanie dobrego formatowania kodu, jak i użycie guards to definiowania kolejnych przypadków.

### 3. Testy

#### MyList.hs

Testy jednostkowe obejmują większość utworzonych funkcji. W niektórych (np. `test_concat` dla funkcji `listConcat`) brakuje testowania większości ujętych przypadków, dla których funkcja ta może zostać wywołana (np. konkatencja kiedy jedna z list jest pusta itp.). Testy Quick Check są wyczerpujące.

#### BinTree.hs

Testy poszczególnych funkcji obejmują wiele przypadków, co jest zdecydowanie plusem. Właściwości testowane testami parametrycznymi mają wystarczający zakres do ogólnego sprawdzenia poprawności funkcji. Nie stwierdziliśmy żadnych zauważalnych braków w obu rodzajach testów.

#### AVLTree.hs

Testy jednostkowe pokrywają wiele przypadków testowych, przez co każda z funkcji jest testowana wystarczająco dobrze, żeby móc stwierdzić jej poprawność podczas implementacji. Ciekawym, choć niekoniecznym dodatkiem, byłaby wizualizacja lub opis tego, jak powinny zachowywać się drzewa AVL w poszczególnych przypadkach. Nie jest to jednak brak - coś takiego stanowiłoby jedynie mały dodatek ułatwiający szybką identyfikację przypadku testowego dla rozległych drzew. Brakuje jednak testów parametrycznych.

### 4. Dokumentacja

Dokumentacja pokrywa wszystkie funkcje zawarte w programie `Data-structures`. Nieco myląca jest jednak struktura dokumentacji gdzie `BinTree.Internal` jest odosobnione z nie do końca jasnych dla użytkownika powodów. Opis funkcji zawartych w każdym module jest natomiast przejrzysty i pozwala na zrozumieniu idei ich działania. Niektóre opisy jednak są zbyt ogólne lub stanowią parafrazę nazwy funkcji. Jednakże w ogólnym rozrachunku dokumentacja ta jest przydatnym narzędziem pozwalającym użytkownikowi na szybkie wdrożenie się w działanie programu jak i zrozumienie technik implementacji.

### 5. Podsumowanie

Projekt `Data-structures` poprawnie implementuje struktury danych oraz pozwala na zaobserwowanie wykonywanych na nich operacji co razem z wglądem w kod źródłowy stanowi wysoką wartość edukacyjną. Biorąc pod uwagę nasz początkujący stopień znajomości języka Haskell oraz paradygmatu funkcyjnego nie można mu niczego zarzucić.