# Звіт до лабораторної 1

Студента групи ТТП-32

Остренка Олександра

## 1. Умова лабораторної:

Залишити у першому списку елементи, що входять у другий список по одному разу.

## 2. Опис виконання:

Реалізацію лабораторної можна розбити на етапи:

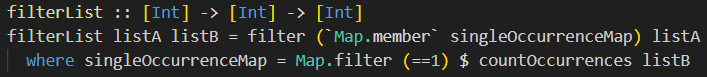
* *Зіставити кожен елемент з другого списку з кількістю його входжень.*



Map (словник) зіставляє деякий унікальний ключ (в нашому випадку – число з другого списку) з якимось значенням (в нашому випадку – кількість входження відповідного числа). Функція countOccurences отримує в якості параметру деякий список цілих чисел, а повертає словник виду ключ-значення, де і ключ і значення типу integer.

Функція foldr працює наступним чином. Вона приймає три аргументи: анонімну функцію (\x acc -> Map.insertWith (+) x 1 acc); значення, яким ініціалізується акумулятор (акумулятор - це деяка сутність, яку повертає дана функція, в даному випадку акумулятор ініціалізується пустим словником Map, відповідно повертати функція foldr буде словник), та список, над яким робиться дійство (сам список на момент ініціалізації не прописується, але, вважаємо, що «х», який використовує анонімна функція, - то є елемент того самого списку). Стандартно, функція foldr бере елемент «х» списку, і використовує його значення щоб порахувати акумулятор (Приклад функції foldr для визначення кількості згадок символу «е» в списку: “count e = foldr (\x acc -> if e==x then acc+1 else acc) 0”) . В даному ж прикладі число зі списку вставляється в словник Map за наступним правилом: якщо це число (майбутній ключ словника) є унікальним, то значення до цього ключа ініціалізується одиницею; якщо таке число (майбутній ключ словника) вже присутнє в словнику, то значення, що відповідає даному ключу буде збільшено на одиницю. Приклад роботи функції для списку [1, 2, 2, 3, 3, 4, 4, 4, 3, 1, 5]:  


* *В першому списку залишити лиш ті елементи, для котрих в другому списку кількість входжень дорівнює одиниці.*



Функція filterList приймає в якості аргументів два цілочисельних списки, а повертає один цілочисельний список.

Функція filter працює наступним чином. Вона приймає два аргументи: предикат, тобто анонімну функцію, яка має повертати виключно булеві значення (`Map.member` singleOccurrenceMap) та список, з якого і будуть відбиратись елементи, задовольняючі предикат. В рядку «filter (`Map.member` singleOccurrenceMap) listA» ми фільтруємо listA таким чином, щоб залишились лиш ті елементи, які є ключами в словнику (Map) singleOccurrenceMap. Тим часом словник singleOccurrenceMap формується наступним чином. Спочатку викликається функція «$ countOccurrences listB», яка формує словник виду число-кількість входжень (описано в першому пункті); далі, «Map.filter (==1)» вибирає лише ті елементи отриманого словника, значення котрих дорівнює одиниці (а це значить, що вони зустрічались лише один раз). По ітогам, ми виберемо з listA лише ті числа, які зустрічаються лиш раз в listB.

## 3. Тестування

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

## 4. Посилання на GitHub.

[Haskell-LAB1-/LAB1(Haskell) at main · SashAmlet/Haskell-LAB1- (github.com)](https://github.com/SashAmlet/Haskell-LAB1-/tree/main/LAB1(Haskell))