Stream API Task

У Java 8 з'явився Stream API, якій дозволяє працювати з вмістом колекцій, як з потоком значень. Тобто для кожного з елементів колекції виконується множина визначених проміжних операції, а у кінці викликається деяка термінальна операція, що повертає фінальний результат. Даний підхід є популярний при роботі з масивам, колекціями та аналізом даних.

У даному завданні пропонується самостійно реалізувати частину Stream API.

В архіві знаходиться шаблон проекту, який містить інтерфейси та набір класів, які необхідно реалізувати.

Шаблон Maven-проекту: **streams.zip**

В даному проекті вже є шаблони класів і невеликий тест, який повинен проходити після реалізації частини методів.

Завдання:

1. Реалізувати *наступні методи класу AsIntStream по роботі з потоком (stream) цілочисельних даних:*

   static IntStream of(int... values)
   створює початковий потік на основі масиву цілих чисел

   - Double average()
   середнє значення чисел в потоці. Термінальній метод. IllegalArgumentException - if empty

   - Integer max()/min()
   максимальне / мінімальне значення числа в потоці. Термінальній метод. IllegalArgumentException - if empty

   - long count()
   кількість значень (елементів) в потоці. Термінальній метод.

   - Integer sum()
   сума всіх значень в потоці. Термінальній метод. IllegalArgumentException - if empty

   - int[] toArray()
   повертає потік у вигляді масиву. Термінальній метод.

   - void forEach(IntConsumer action)
   для кожного значення з потоку виконує операцію зазначену в реалізації IntConsumer. Даний метод є термінальним і нічого не повертає

- IntStream filter(IntPredicate predicate)

для кожного значення з потоку перевіряє його на предмет чи задовольняє воно умові в реалізації IntPredicate, якщо так - повертає його в результуючий потік, якщо ні - викидає

- IntStream map(IntUnaryOperator mapper)

застосовує до кожного зі значень потоку реалізацію IntUnaryOperator і повертає його в результуючий потік

- IntStream flatMap(IntToIntStreamFunction func)

застосовує до кожного зі значень потоку реалізацію IntToIntStreamFunction, яка на основі кожного зі значення створює новий потік значень, які потім об'єднуються в один результуючий потік

- int reduce(int identity, IntBinaryOperator op)

виконує згортку значень потоку в ціле число, початкове значення задається identity, функція згортки - в реалізації IntBinaryOperator. Термінальній метод.

2. Для покриття коду тестами, рекомендуємо використовувати TDD-підхід. Необхідно написати модульні тести на всі методи класу *AsIntStream*

При реалізації можна використовувати стандартні колекції з Java, але ***не можна використовувати Java Stream API.***

Спробуйте реалізувати функціональність проміжних методів (filter, map, flatMap) таким чином, щоб визначені у них операції не виконувались до виклику одного з термінальних методів. Це буде відповідати підходу “лінивих обчислень”. Таку реалізацію можна зробити використовуючи ***Iterator pattern****. Також такий підхід дозволить не створювати зайвих проміжних масивів/колекцій. Тобто всі операції на потоці повинні бути виконані за один обхід (одну ітерацію по елементах).*

Кінцевий проект необхідно запакувати (zip-архів) та надіслати на email з якого отримали завдання.