

# Практическая работа №3

## Теоретическое введение

### Полезные ссылки

- 1) Официальная документация <https://reactivex.io/documentation>

## Задание

### Задание 1

Реализовать следующую систему:

Датчик температуры. Каждую секунду публикует значение температуры (случайное значение от 15 до 30). Датчик CO<sub>2</sub>. Каждую секунду публикует значение содержания CO<sub>2</sub> в воздухе. (Случайное значение от 30 до 100). Сигнализация. Получает значения от датчиков. Если один из показателей превышает норму, выводит предупреждение об этом. Если норму превышают оба показателя выводит сообщение «ALARM!!!».

Норма показателей: Температура — 25. CO<sub>2</sub> — 70.

Обязательно использование классов Observer и Observable из библиотеки RxJava.

### Задание 2

2.1.1 Преобразовать поток из 1000 случайных чисел от 0 до 1000 в поток, содержащий квадраты данных чисел.

2.1.2 Преобразовать поток из 1000 случайных чисел от 0 до 1000 в поток, содержащий только числа больше 500.

2.1.3 Преобразовать поток из случайного количества (от 0 до 1000) случайных чисел в поток, содержащий количество чисел.

2.2.1. Даны два потока по 1000 элементов: первый содержит случайную букву, второй — случайную цифру. Сформировать поток, каждый элемент которого объединяет элементы из обоих потоков. Например, при входных потоках (A, B, C) и (1, 2, 3) выходной поток — (A1, B2, C3).

2.2.2. Даны два потока по 1000 элементов. Каждый содержит случайные цифры. Сформировать поток, обрабатывающий оба потока последовательно. Например, при входных потоках (1, 2, 3) и (4, 5, 6) выходной поток — (1, 2, 3, 4, 5, 6).

2.2.3. Даны два потока по 1000 элементов. Каждый содержит случайные цифры. Сформировать поток, обрабатывающий оба потока параллельно. Например, при входных потоках (1, 2, 3) и (4, 5, 6) выходной поток — (1, 4, 2, 5, 3, 6).

2.3.1. Дан поток из 10 случайных чисел. Сформировать поток, содержащий все числа, кроме первых трех.

2.3.2. Дан поток из 10 случайных чисел. Сформировать поток, содержащий только первые 5 чисел.

2.3.3. Дан поток из случайного количества случайных чисел. Сформировать поток, содержащий только последнее число.

Каждый студент выполняет задания 2.1.X, 2.2.X, 2.3.X, где X = номер по журналу%3+1.

Должно быть реализовано при помощи инструментов RxJava.

### **Задание 3**

Реализовать класс UserFriend. Поля — int userId, friendId. Заполнить массив объектов UserFriend случайными данными.

Реализовать функцию: Observable<UserFriend> getFriends(int userId), возвращающую поток объектов UserFriend, по заданному userId. (Для формирования потока из массива возможно использование функции Observable.fromArray(T[] arr)).

Дан массив из случайных userId. Сформировать поток из этого массива. Преобразовать данный поток в поток объектов UserFriend. Обязательно получение UserFriend через функцию getFriends.

### **Задание 4**

Реализовать следующую систему.

Файл. Имеет следующие характеристики:

0. Тип файла (например XML, JSON, XLS)

1. Размер файла — целочисленное значение от 10 до 100.

Генератор файлов — генерирует файлы с задержкой от 100 до 1000 мс.

Очередь — получает файлы из генератора. Вместимость очереди — 5 файлов.

Обработчик файлов — получает файл из очереди. Каждый обработчик имеет параметр — тип файла, который он может обработать. Время обработки файла: «Размер файла\*7мс». Система должна быть реализована при помощи инструментов RxJava.