**Задача 1. Stream/Lambda.**

Не использовать циклы (for/while/do-while). Сгенерировать 100 рандомных Long чисел в диапазоне от 0 до 100. Каждое число умножить на PI и отнять 20. Отфильтровать, оставив числа меньшие 100. Отсортировать по возрастанию. Пропустив первые 3 числа произвести следующие операции:

Преобразовать (операция map) лист чисел в Map, где ключем является строка: ("Number: " + value).

Собрать значения в list с помощью collect, где значениями оставить строковые представления ("Number: " + value).

**Задача 2. Reflection API.**

1. Создать свой класс Person с полями: firstName, lastName, age, dateOfBirth
2. Добавить класс User, который наследуется от Person, с полями: login, password, email
3. Добавить гетеры, сетеры. Добавить метод printUserInfo в User. Вывести с помощью рефлексии все поля и методы, сначала через

getMethod("name")

getMethods()

getField("name")

getFields()

а затем через

getDeclaredMethod("name")

getDeclaredMethods()

getDeclaredField("name")

getDeclaredFields()

Посмотреть разницу. Просетать все значения через Reflection. (set метод класса Field). Вывести значения полей через Reflection. (get метод класса Field). Вызвать toString через invoke.

**Задача 3. jUnit тестирование.**

1) создаем 3 классса-теста

2) каждый из которых будет тестировать одну функцию калькулятора

3) протестировать методы 3 наборами данных (используем Parametrize)

4) объеденить 3 класса-теста в Test Suite.

5) добавить CalculatorTestListener, переопределить, метод который вызывается на успешное завершение теста и выводит на экран название нашего метода.