**Министерство науки и высшего образования Российской Федерации**

**федеральное государственное автономное образовательное учреждение высшего образования**

**“НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО”**

**Реактивная Java**

Отчёт

**Лабораторная работа №1**

Выполнили:

Голоскок Дмитрий Сергеевич P4119

Симовин Кирилл Константинович P4116

Преподаватель:

Гаврилов Антон Валерьевич

Санкт-Петербург

2024

**Задание**

1. Написать для согласованной предметной области как минимум 3 базовых класса и генераторы объектов. Генератор должен уметь создавать указанное количество различных объектов сответствующего класса со случайными (но при этом валидными) характеристиками. Класс, представляющий собой массовый объект должен обязательно содержать поля следующих типов:

- Один из примитивов (`int`, `long`, `double`).

- `String`.

- Дата/время (`LocalDate`, `LocalTime`, ...).

- `Enum`.

- `Record`.

- Массив или коллекция.

Остальные поля - произвольные, какие нужны для предметной области.

Два оставшихся класса должны представлять собой дополнительные атрибуты и характеристики массового класса.

2. С помощью генератора создать коллекцию объектов.

3. Написать код, реализующий расчет согласованных агрегированных статических данных тремя способами:

- Итерационным циклом по коллекции.

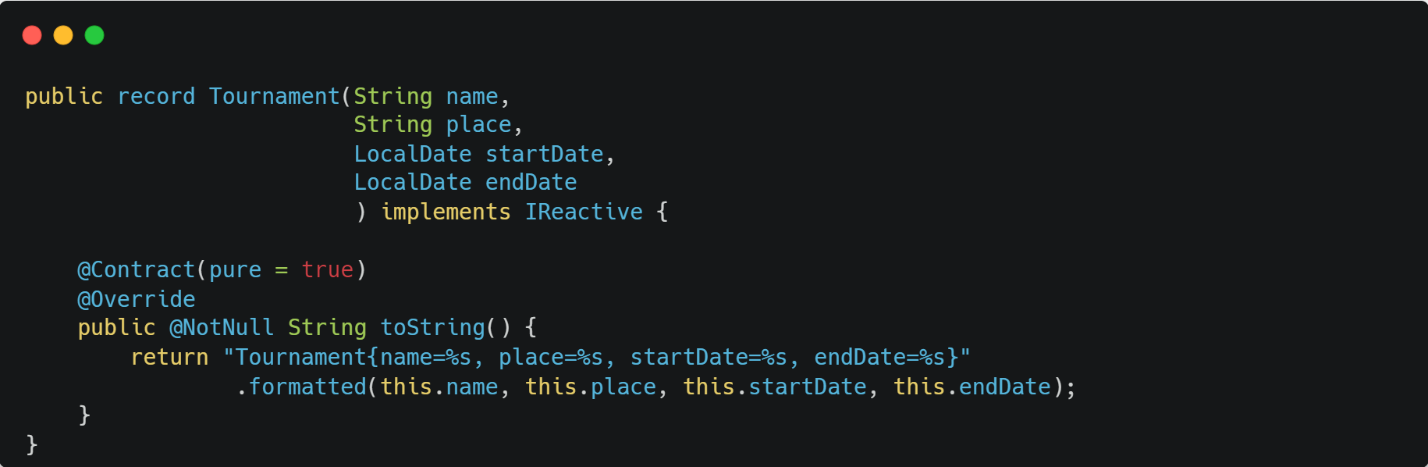
- Конвейером с помощью Stream API на базе коллекторов из стандартной библиотеки.

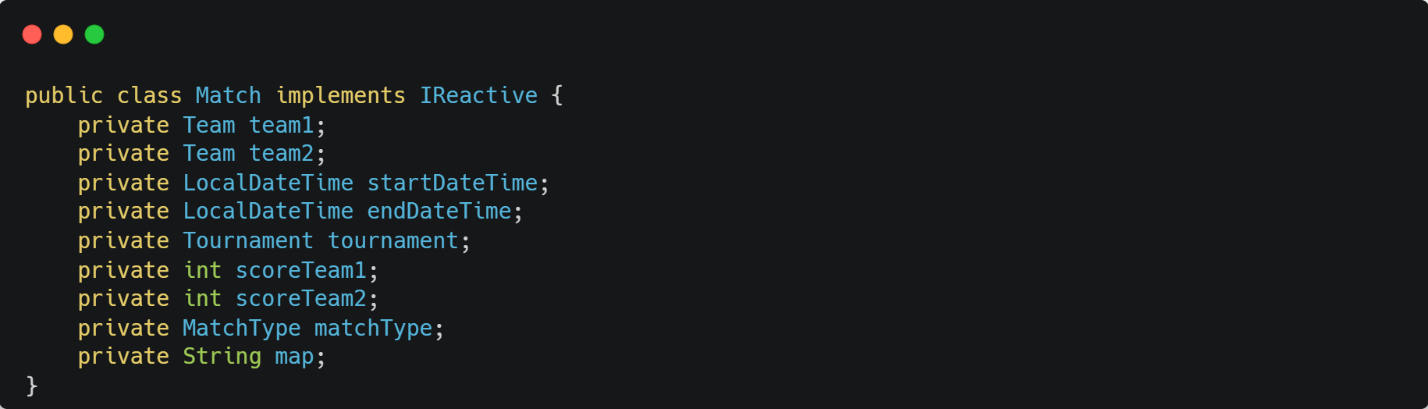
- Конвейером с помощью собственного коллектора.

4. Для каждого варианта измерить время выполнения, зафиксировав моменты начала и окончания расчета для количества

элементов в коллекции - 5000, 50000 и 250000. Время измерять с помощью методов класса `System` или `Instant`.

**Классы**





**Собственный коллектор**



Сравнительная таблица по методам обработки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 100 | 5000 | 50000 | 250000 |
| Цикл | 0.013 | 0.005 | 0.038 | 0.078 |
| Стандартный коллектор | 0.01 | 0.009 | 0.023 | 0.063 |
| Собственный коллектор | 0.006 | 0.006 | 0.023 | 0.054 |

Ссылка на Github c кодом

<https://github.com/BestAlice/ReactiveJava/tree/main>

**Выводы**

Как мы видим из сравнительной таблицы, расчёт с использованием итеративным циклом работает быстрее прочих при среднем объёме данных ~5000. В остальных случаях оптимальным оказался написанный нами коллектор.