**Міністерство освіти і науки України**

**Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського"**

**Факультет інформатики та обчислювальної техніки**

**Кафедра інформатики та програмної інженерії**

**Звіт**

з лабораторної роботи № 3 з дисципліни «Технології паралельних обчислень»

**«Розробка паралельних програм з використанням механізмів синхронізації: синхронізовані методи, локери, спеціальні типи»**

**Виконав(ла)**

(шифр, прізвище, ім'я, по батькові)

*ІП-02, Зусько Владислав Юрійович*

Київ 2023

**Завдання #1**

Реалізуйте програмний код, даний у лістингу, та протестуйте його при різних значеннях параметрів. Модифікуйте програму, використовуючи методи управління потоками, так, щоб її робота була завжди коректною. Запропонуйте три різних варіанти управління. **30 балів.**

Попередній код мав ряд проблем. До них входили зокрема такі:

* Відсутність перевірки платоспроможності рахунку, клієнт може переказати гроші, яких не має
* Відсутність коректного механізму синхронізації, через що загальна сума змінювалась з плином транзацій

Запропонованими варіантами вирішення проблеми синхронізаці стали наступні техніки: синхронізований метод, синхронізований блок, локер та умова.

**Завдання #2**

Реалізуйте приклад Producer-Consumer application (див. https://docs.oracle.com/javase/tutorial/essential/concurrency/guardmeth.html ). 12 Модифікуйте масив даних цієї програми, які читаються, у масив чисел заданого розміру (100, 1000 або 5000) та протестуйте програму. Зробіть висновок про правильність роботи програми. **20 балів**

Загалом було запропоновано використати абстрактний приклад з реального життя. В нього входять такі сутності як фанат, співак та альбом. Співак з певною періодичністю випускає нові пісні у альбоми, на які фанат реагує. Поки фанат не послухав попередню пісню, співак не може випустити наступну, як тільки послухав – одразу можна випустити наступну.

Після тестування на різній кількості елементів ми впевнюємось, що кількість програма працює правильно, адже кількість зчитувань дорівнює кількості випущених значень. Для забезпечення цілісності даних було використано локер та умову.

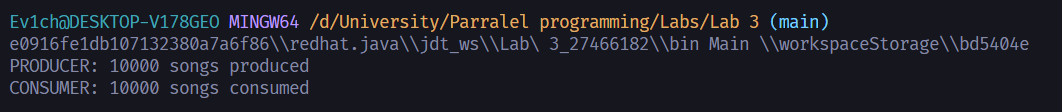


Рисунок – Приклад виконання програми

**Завдання #3**

Реалізуйте роботу електронного журналу групи, в якому зберігаються оцінки з однієї дисципліни трьох груп студентів. Кожного тижня лектор і його 3 асистенти виставляють оцінки з дисципліни за 100-бальною шкалою. **40 балів.**

Головним за логікою у цій задачі є функціонал паралельного виставлення оцінок різним студентам. Важливо забезпечити цей процес так, аби виставлення оцінок одному студенту не блокувало виставлення оцінок іншому студентові. Також, при цьому, важливо забезпечити цілісність даних, аби оцінки не могли перезаписати одна одну. Для забезпечення вказаних можливостей пропонується використати ConcurrentMap та локер. Ця структура даних допоможе правильно обробляти додавання оцінок, а локер забезпечить цілісність у ситуаціях, коли потрібно додати нову групу у журнал, наприклад, тобто нам необхідне блокування всього журналу.

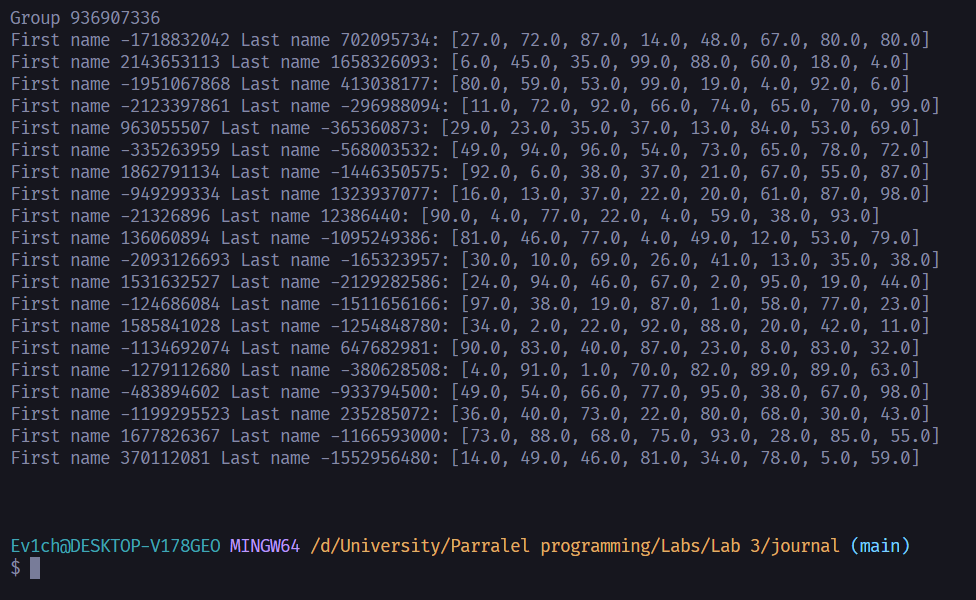


Рисунок – Приклад виконання програми

**Висновок**

Виконуючи цю лабораторну роботи ми змогли застосувати різноманітні методи для забезпечення цілісності даних у програмі, що працює у багатопотоковому режимі. Ми впевнились, що мова містить дуже широкий інструментарій для роботи з мультипотоковістю та дозволяє використати різні методики досягнення паралельної обробки даних. Варто зазначити, що кожен з розглянутих методів буде більш чи менш доречним залежно від контексту та задачі.