Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

**Комп’ютерний практикум №5**

з дисципліни «Технології паралельних обчислень»

Тема: «Застосування високорівневих засобів паралельного програмування для побудови алгоритмів імітації та дослідження їх ефективності»

|  |  |
| --- | --- |
| **Виконав:**  студент групи ІТ-04  Коновальчук Андрій Володимирович  Дата здачі \_\_\_\_\_\_\_\_  Захищено з балом \_\_\_\_\_\_\_ | **Перевірила**:  асистент кафедри ІПІ  Дифучина Олександра Юріївна |

Київ 2023

**Хід роботи:**

***Завдання 1:*** З використанням пулу потоків побудувати алгоритм імітації багатоканальної системи масового обслуговування з обмеженою чергою, відтворюючи функціонування кожного каналу обслуговування в окремій підзадачі. Результатом виконання алгоритму є розраховані значення середньої довжини черги та ймовірності відмови.

*Опис пункту та фрагменту коду:*

Для імітації багатоканальної СМО необхідно створити обмежену чергу задач та ініціалізувати пул потоків котрі оброблятимуть ці задачі. Підрахувати випадки відмов та значення черги на кожній ітерації.

В коді використовуємо клас LinkedBlockingQueue та ThreadPoolExecutor. Останній приймає значення кількості потоків, очікування (нульове) та блокуючу чергу. Є дві затримки: одна симулює рух черги, інша симулює обробку завдання. Обробкою завдання займається клас DelayedTask, котрий розширює клас Runnable.

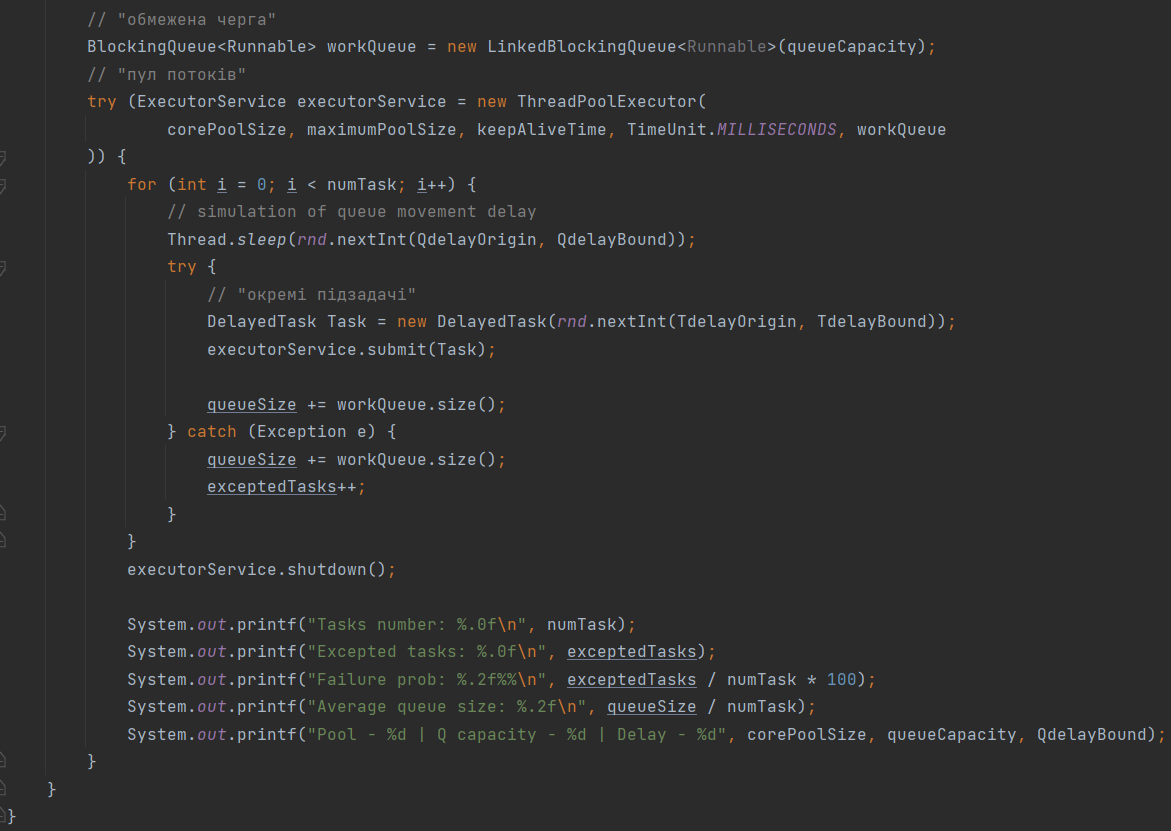


Рисунок 1 – Клас Main

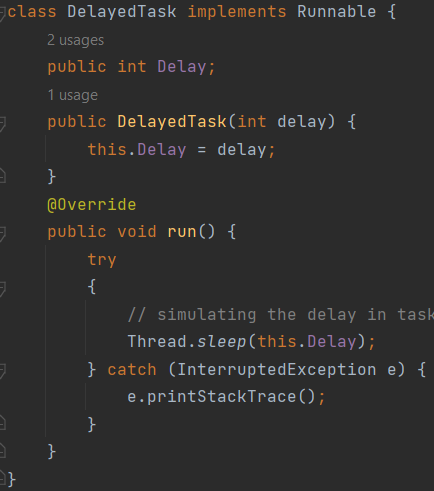


Рисунок 2 – Клас DelayedTask

*Спостереження, результати, висновки:*

В залежності від зміни параметрів змінюється значення середньої довжини черги та ймовірності відмови. Змінюються параметри кількості потоків у пулі, фіксована величина черги, величина затримки.

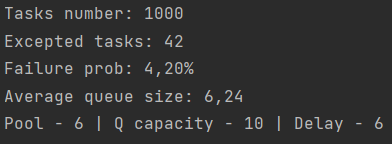
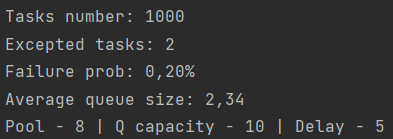
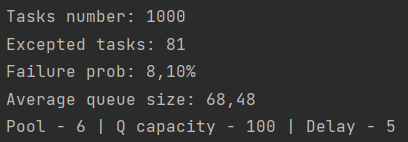
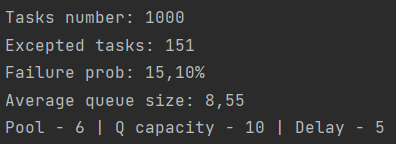


Рисунок 3 – Результати роботи зі зміною параметрів

***Завдання 2:*** З використанням багатопоточної технології організувати паралельне виконання прогонів імітаційної моделі СМО для отримання статистично значимої оцінки середньої довжини черги та ймовірності відмови.

*Опис пункту та фрагменту коду:*

Імплементуємо код з попереднього пункту в клас Model. Створимо клас з результатами роботи моделі Result. Запустимо n потоків зі списком об’єктів класу результату, виведемо результати та обчислимо статистичне значення параметрів.

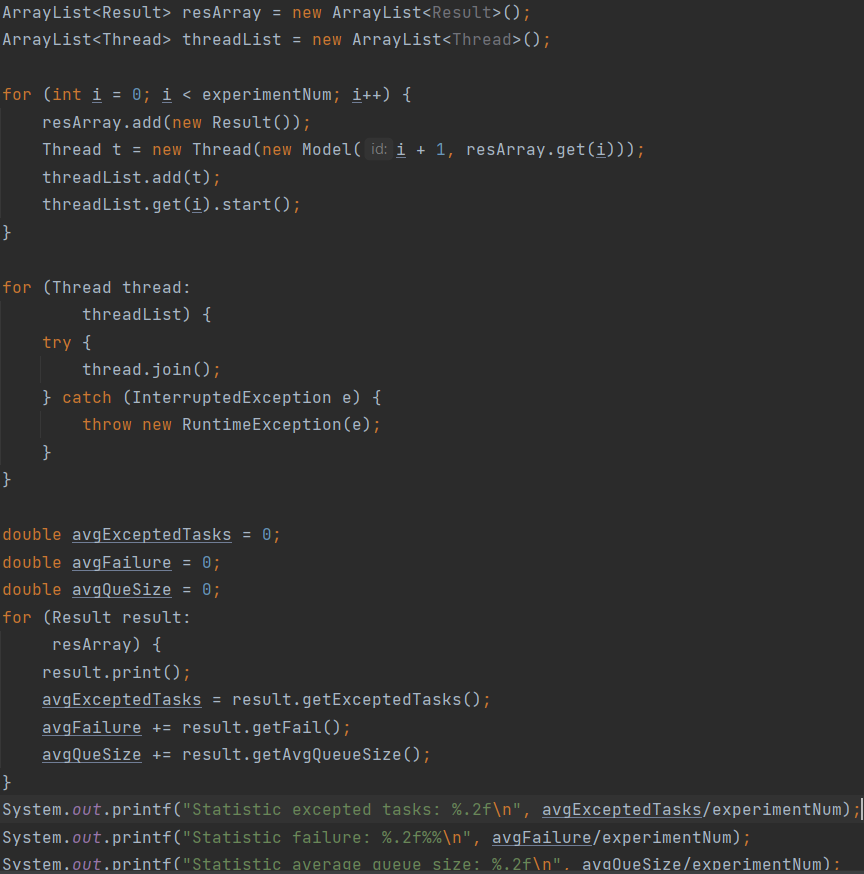


Рисунок 4 – Клас Main, паралельний прогон моделі

*Спостереження, результати, висновки:*

Виводимо список результатів прогону моделі та знаходимо статистично значиму оцінку середньої довжини черги та ймовірності відмови. Бачимо відносно однакову роботу моделей. Результати виведені по-порядку за допомогою використання списку з результатами, і тому що один результат зі списку відноситься до однієї моделі.

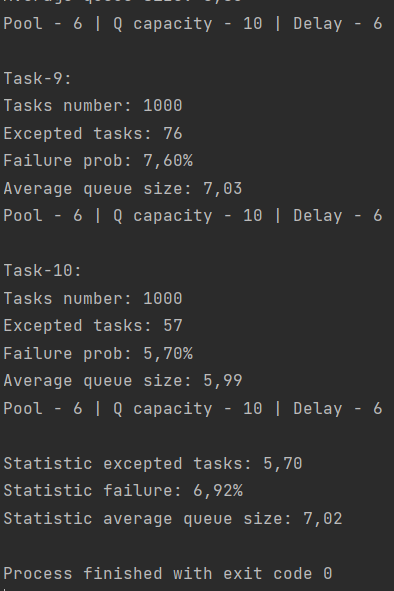
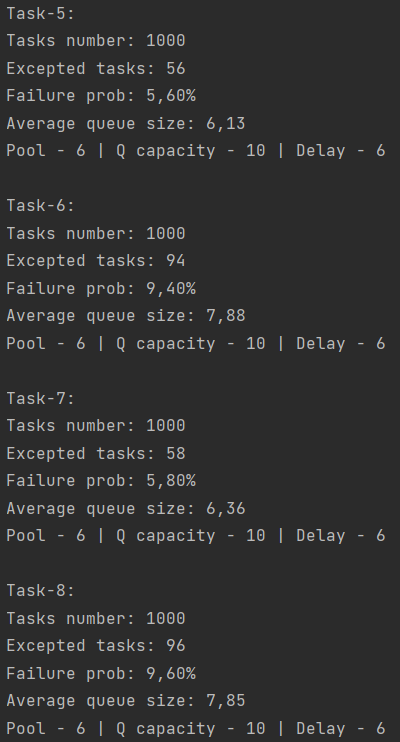
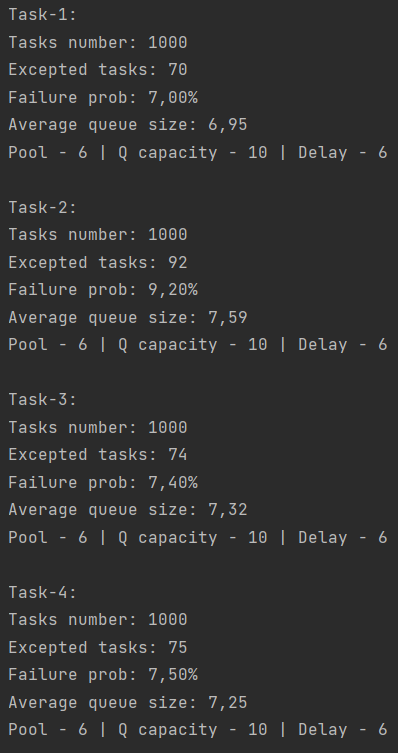


Рисунок 5 – Результати експерименту із паралельним прогоном моделі

***Завдання 3:*** Виводити результати імітаційного моделювання (стан моделі та чисельні значення вихідних змінних) в окремому потоці для динамічного відтворення імітації системи.

*Опис пункту та фрагменту коду:*

Цей пункт також є доповненням першого і передбачає створення логера. Логер має працювати в окремому потоці і має виводити таку корисну інформацію: номер ітерації, статус (стан) моделі, простір черги (заповнено і вільно), кількість відмов.

Через те що за умовою задачі ми маємо працювати в окремому потоці, і пул потоків уже створений, тому ми використовуємо newCachedThreadPool(), котрий створює пул потоків, який за потреби створює нові потоки, але повторно використовує попередньо створені потоки, коли вони доступні.



Рисунок 6 – Клас Logger

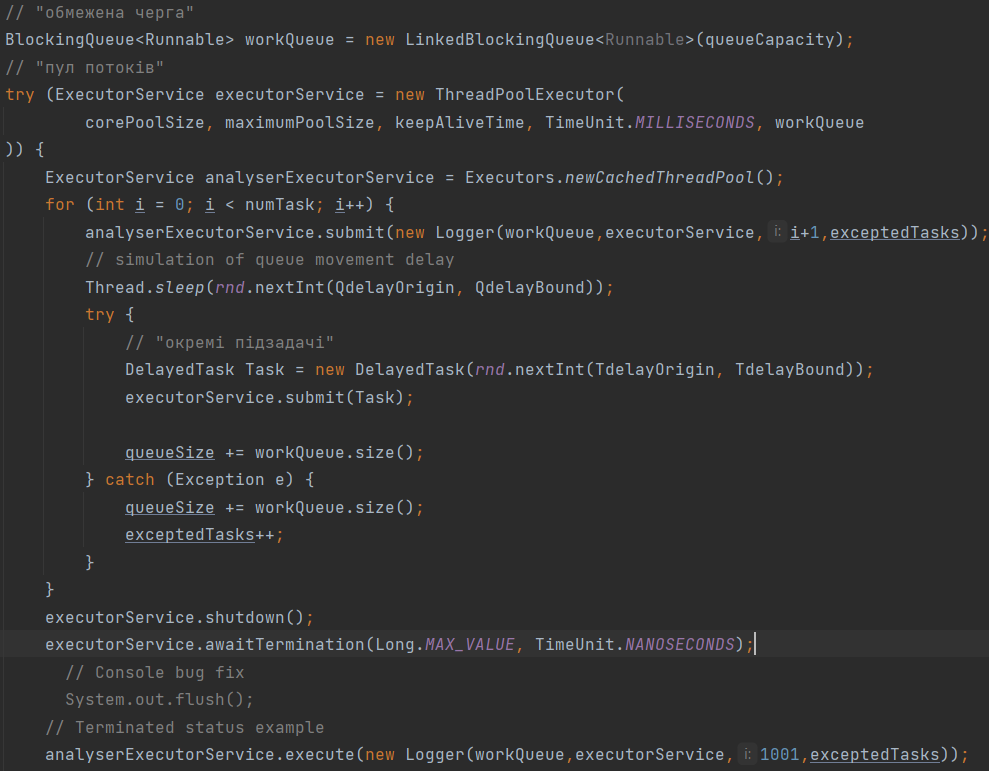


Рисунок 7 – Клас Main

*Спостереження, результати, висновки:*

Бачимо послідовний вивід інформації про роботу імітаційного моделювання на кожній ітерації. 1001 ітерація зроблена для прикладу статусу “terminated” після виконання методу awaitTermination().

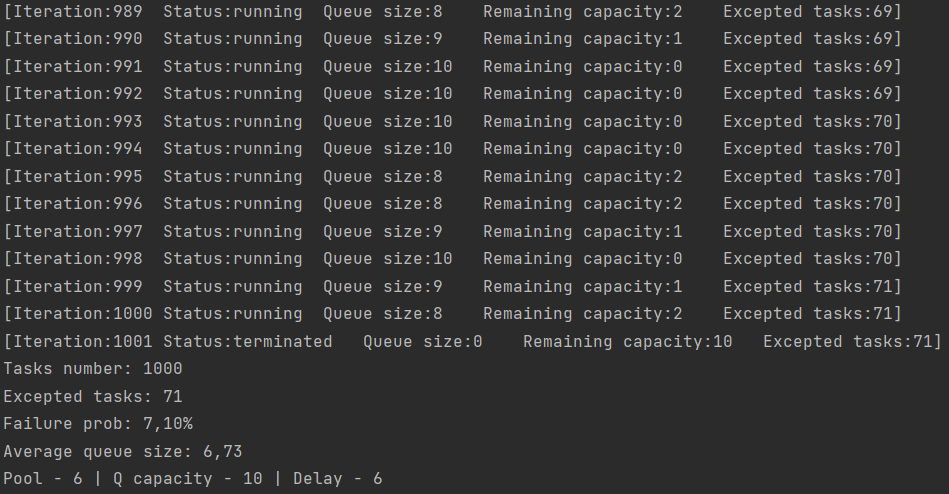


Рисунок 8 – Результати роботи логера

***Завдання 4:*** Побудувати теоретичні оцінки показників ефективності для одного з алгоритмів практичних завдань 2-5.

*Опис пункту та фрагменту коду:*

Для вирішення цієї задачі було використано завдання з першого пункту. Необхідно визначити такі показники ефективності: прискорення (Sp(n)), ефективність (Ep(n)), вартість (Cp(n)). Обчислюємо їх за формулами:

, ,

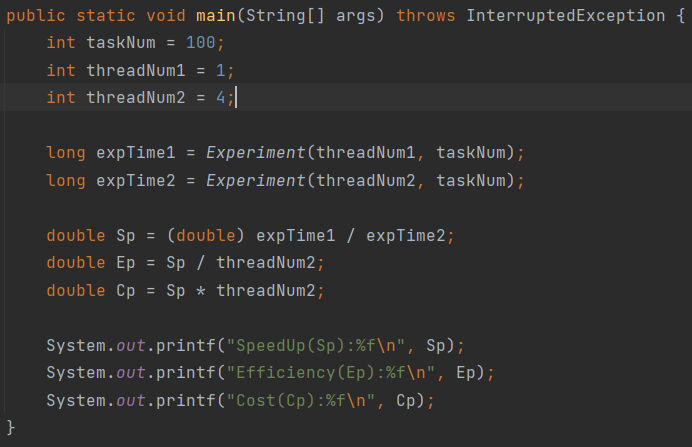


Рисунок 9 – Метод Main

*Спостереження, результати, висновки:*

Наша задача є достатньо простою, не містить багато обчислень та має імітацію затримки, а також швидкість ядра процесора дозволяє обчислювати такі завдання з високою швидкістю. Тому результати експерименту показують низькі параметри ефективності. Результати експерименту з різними параметрами:

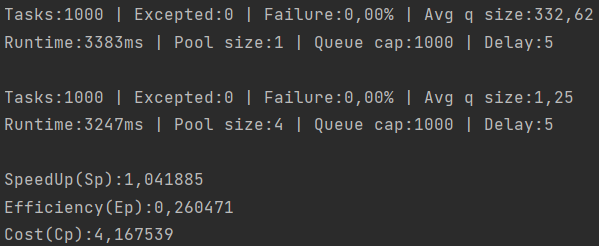


Рисунок 10 – Експеримент 1000 завдань 4 потоки

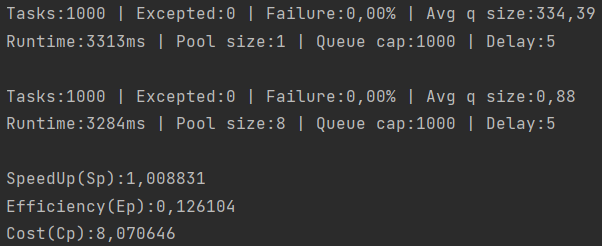


Рисунок 11 – Експеримент 1000 завдань 8 потоки

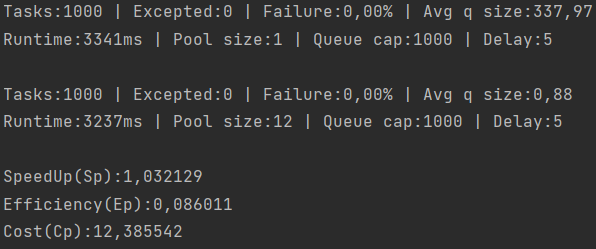


Рисунок 12 – Експеримент 1000 завдань 12 потоки

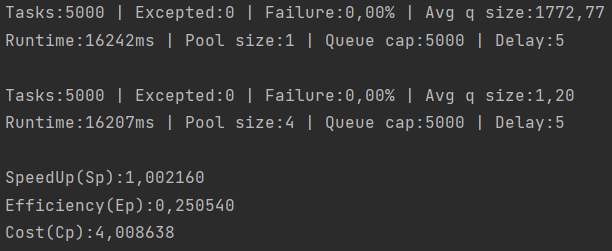


Рисунок 13 – Експеримент 5000 завдань 4 потокb

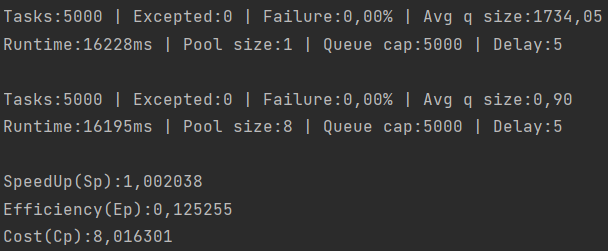


Рисунок 14 – Експеримент 5000 завдань 8 потоків

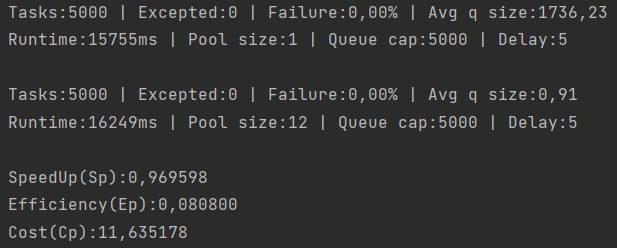


Рисунок 15 – Експеримент 5000 завдань 12 потоків

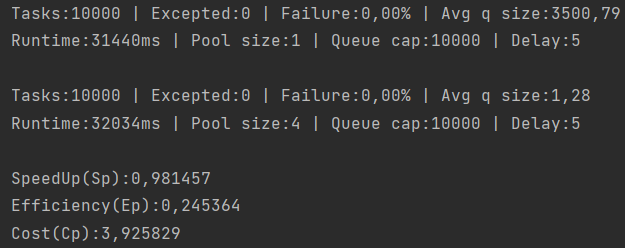


Рисунок 16 – Експеримент 10000 завдань 4 потоки

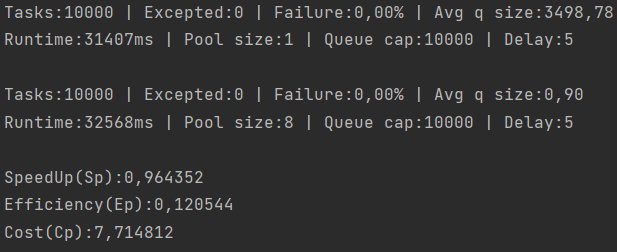


Рисунок 17 – Експеримент 10000 завдань 8 потоків

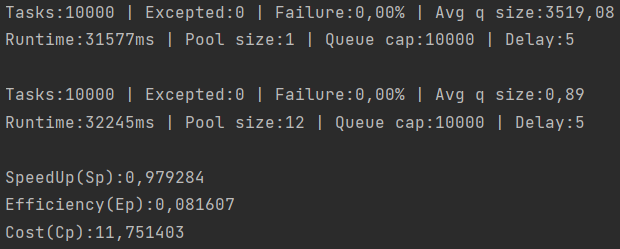


Рисунок 18 – Експеримент 10000 завдань 12 потоків

Висновок: алгоритм 5 лабораторної не вигідно розпарелелювати.

**Висновок:**

В ході виконання практикуму було побудовано алгоритм імітації багатоканальної СМО з обмеженою чергою. Було проведено експерименти, зібрано статистику та оцінювання функціювання цієї моделі з використанням багатопоточної технології.

У звіті наявні скріншоти з прикладами результатів роботи коду, а також опис роботи коду та пояснення результатів.

**Лістинг коду у вигляді GitHub-репозиторію:**

<https://github.com/m4cy43/parallel_programming/tree/master/lab5>