

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА  
ШЕВЧЕНКА**

**ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА КІБЕРНЕТИКИ**

**ЗВІТ**

**Лабораторній роботі №3**

**Студентки групи ІПС-23**

**Матвійчук Анастасії Миколаївни**

**Київ 2023**

**Тема:** Паралельна мультипоточна реалізація алгоритмів.

**Було реалізовано:** У рамках даної лабораторної роботи було реалізовано послідовну та паралельну версії алгоритмів для виконання операцій з матрицями (зокрема пошуку оберненої матриці до заданої), що підтримують використання комплексних чисел - `GaussJordanInverse`, `ParallelGaussJordanInverse`, `LUInverse`, `ParallelLUInverse`, `Strassen`, `ParallelStrassen`. Програмна реалізація була здійснена з використанням засобів мультипоточності та з дотриманням принципів об'єктно-орієнтованого програмування.

Для досягнення гнучкості та розширюваності коду, були використані **патерни проєктування**, а саме:

- Фабричний метод – створення класу `MatrixInverseFactory`, що є загальним інтерфейсом для керування різними версіями алгоритму пошуку оберненої матриці до заданої: `GaussJordanInverse`, `ParallelGaussJordanInverse`, `LUInverse`, `ParallelLUInverse`;
- Декоратор – створення класу `TimeMatrixInverseFactory`, що дає змогу динамічно додавати об'єктам класу `MatrixInverseFactory` нову функціональність для вимірювання часу різних версій алгоритму.

Були реалізовані **юніт-тести** для всіх версій алгоритмів, що дозволило перевірити правильність їх функціонування на різних вхідних даних. Також було проведено **вимірювання часу виконання алгоритмів** для різних розмірів даних та різної кількості потоків. Порівняння результатів паралельної та послідовної версій дозволило оцінити ефективність використання мультипоточності. Результати вимірювання часу показали переваги використання паралельної мультипоточної версії алгоритмів у порівнянні з послідовною версією.

**Документація** коду була створена з використанням `Doxygen`, що дозволяє зрозуміло та систематично описати функціонал, структуру та використовувані класи в коді. Це сприяє зрозумілості, облегшує розробку, підтримку та подальше розширення реалізованої системи.

**Висновок:** Виконання цієї лабораторної роботи дозволило покращити розуміння паралельного мультипоточного програмування, практично застосувати принципи об'єктно-орієнтованого програмування та патерни проєктування, а також оцінити ефективність паралельної реалізації алгоритмів у порівнянні з послідовною версією.