Харківський національний університет імені В. Н. Каразіна

Факультет комп'ютерних наук

ЗВІТ

З ЛАБОРАТОРНОЇ РОБОТИ №5

# ДИСЦИПЛІНА: «КРОС-ПЛАТФОРМНЕ ПРОГРАМУВАННЯ»

Виконав: студентка групи КС22

Мазуренко Анжеліки

Перевірив: Споров Олександр

Євгенович

Харків

2024

**Лабораторні роботи №5 Java Net Programming**

**(UDP Sockets)**

**Git Repository:** [**https://github.com/miorezu/LB5\_JavaNet**](https://github.com/miorezu/LB5_JavaNet)

**Завдання №1**

Напишіть простий розподілений клієнт / серверний додаток за допомогою TCP

сокетів. У цьому додатку сервер приймає завдання від клієнтів, виконує ці завдання, визначає час їх виконання і потім повертає всю цю інформацію клієнту. При цьому саме клієнти створюють свої власні завдання і відправляють їх на сервер для виконання (клас завдання повинен реалізовувати інтерфейс, визначений відповідно до договору із сервером).

Визначення класу завдання відправляється клієнтом на сервер і, щойно class - файл стає доступним, сервер може виконувати отримане завдання. Аналогічно, сервер створює об’єкт класу результату і відправляє його разом із визначенням класу клієнта. При цьому клас результату реалізує інтерфейс, відомий клієнту.

В якості клієнтського завдання можна взяти завдання обчислення факторіала достатньо великого числа.

У нас є дві програми LB5\_Server та LB5\_Client. У першій у нас є 2 класи та 2 інтерфейси: ResultImpl, Server, Result, Executable. У другій програмі 2 таких самих інтерфейси та 2 класи – Client, Factorial. Factorial – завдання клієнта, яке відправляється на сервер.

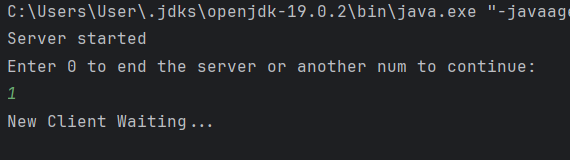


Рисунок 1 - запускаємо сервер

Запускаємо сервер і бачимо,що він очікує кліента. Запустимо клієнта

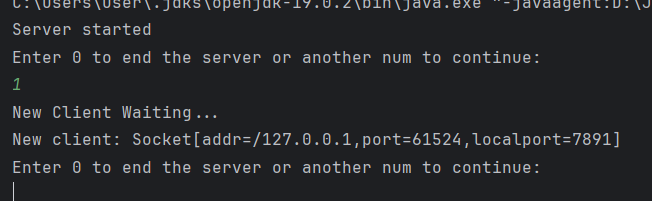


Рисунок 2 – на сервері видно, що клієнт під’єднався

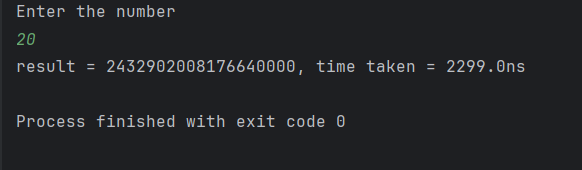
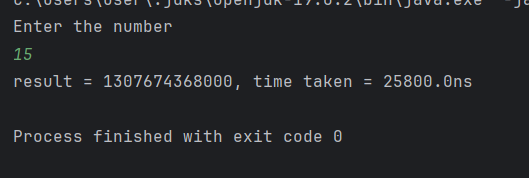


Рисунок 3-4 - клієнтом відправяємо на сервер завдання і отримуємо результат

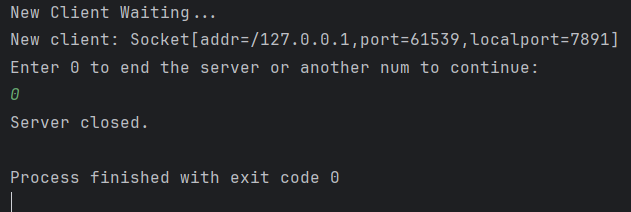


Рисунок 5 - коли є запит закрити чи продовжити роботу сервера, натиснувши 0 завершимо роботу

**Завдання №2**

Розглянемо взаємодію двох комп'ютерів у мережі. Перш ніж розпочати роботу,

комп'ютери повинні обмінятися IP-адресами. Це може стати досить непростим завданням.

Створимо спеціальний UDP сервер, який допоможе комп'ютерам обмінятися «координатами»: IP-адресами та номерами портів. Потім створимо UDP клієнтів, які

перевірять роботу сервера: відправлять запит, який реєструє клієнта (комп’ютер) на сервері, і отримають відповідь сервера — список вже зареєстрованих комп'ютерів.

У програмі в нас наявно 4 класи – ActiveUsers, User, UDPServer та UDPClient.

Перші 2 класи призначені для зберігання інформації , а інших – сервер та клієнт, реалізовані за допомогою UDP протоколів.

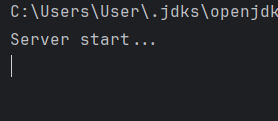


Рисунок 6 – запускаємо сервер

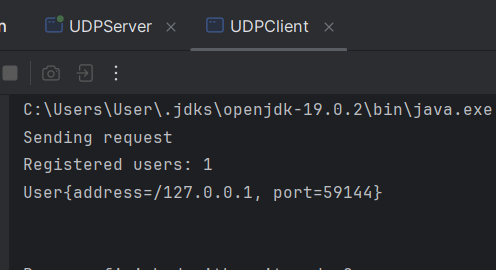


Рисунок 7 - запускаємо клієнта

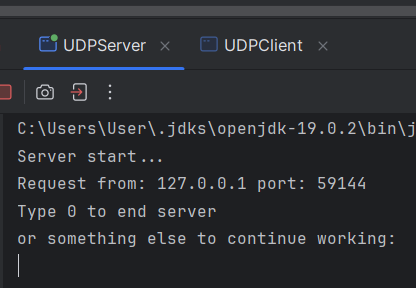


Рисунок 8 - інформація на сервері

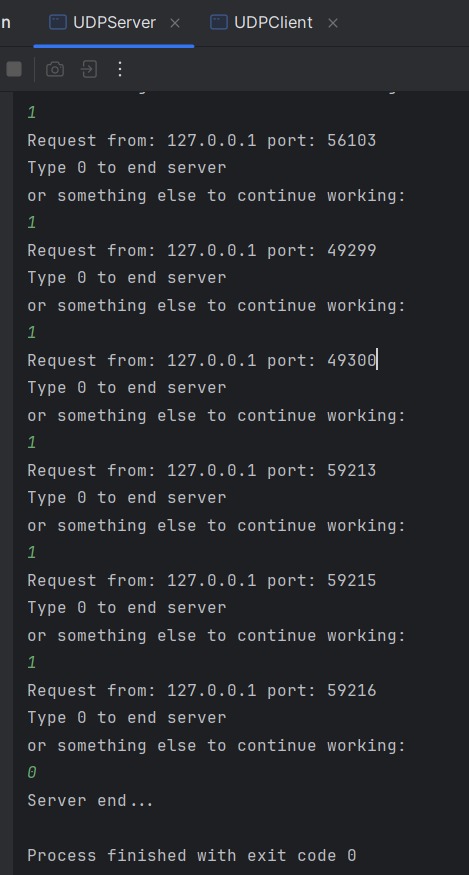


Рисунок 9 – результати виконання програми на серверній частині та закриття серверу

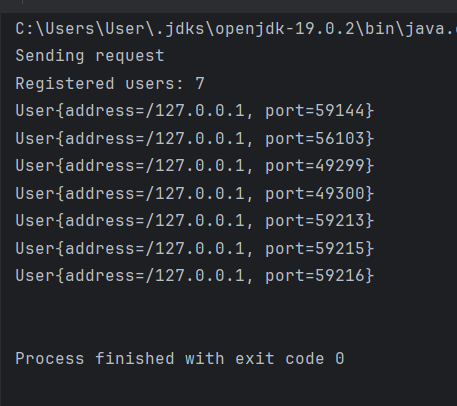


Рисунок 10 – результати виконання програми на клієнській частині