

Харківський національний університет імені В. Н. Каразіна  
Факультет комп'ютерних наук  
Кафедра штучного інтелекту та програмного забезпечення

ЗВІТ  
З ЛАБОРАТОРНОЇ РОБОТИ №5

дисципліна: «Крос-платформне програмування»

Виконав: студент групи КС21  
Барбінов Віктор Андрійович

Перевірив: доцент кафедри ШІтаПЗ  
Споров Олександр Євгенович

Харків  
2024

## Завдання №2 (у папці src репозиторію)

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.

## Результати виконання завдання №2 (консольна версія) :

```
Enter server IP address:  
localhost  
Enter server port:  
12345  
Enter a number to calculate factorial:  
27  
Result: 10888869450418352160768000000, Time Taken: 74800,00 ns  
  
Process finished with exit code 0
```

Рисунок 1 – Клієнтська частина

```
Server is waiting for connections...  
Task executed and result sent.  
  
Process finished with exit code 0
```

Рисунок 2 – Серверна частина

## Результати виконання версії №2 (версія з графічним інтерфейсом):

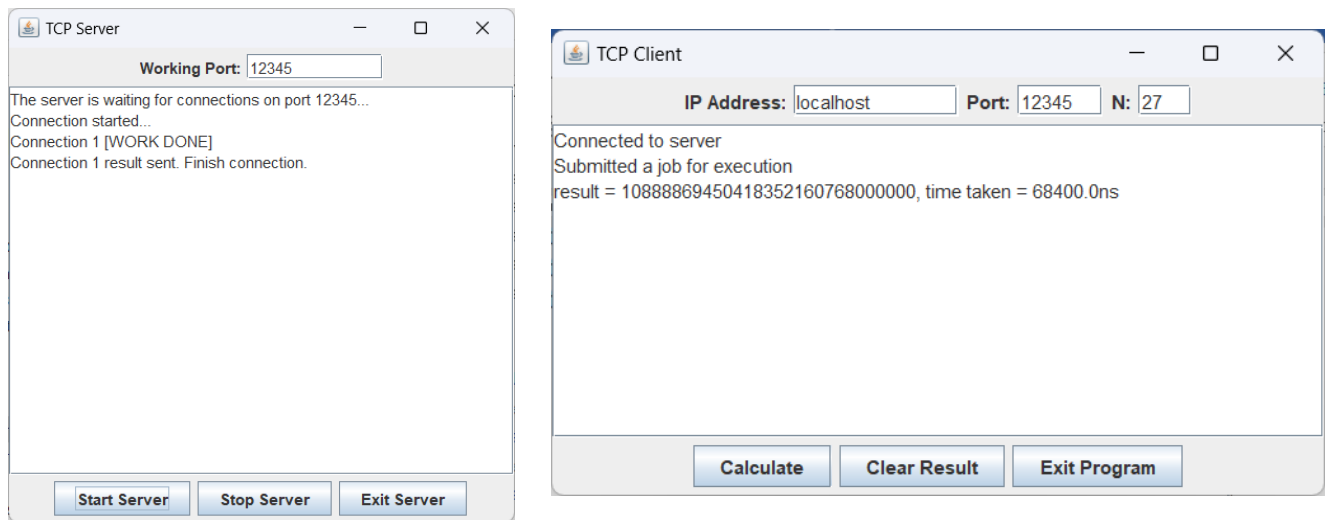


Рисунок 3 – Результат виконання програми

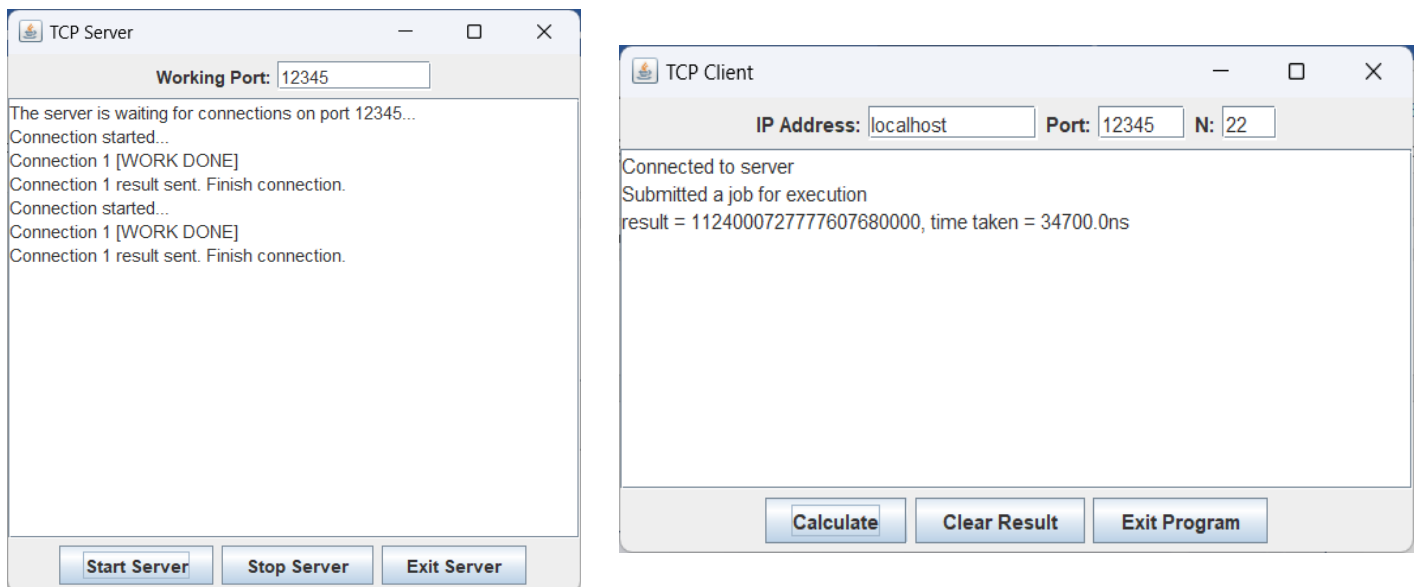


Рисунок 4 – Результат виконання програми

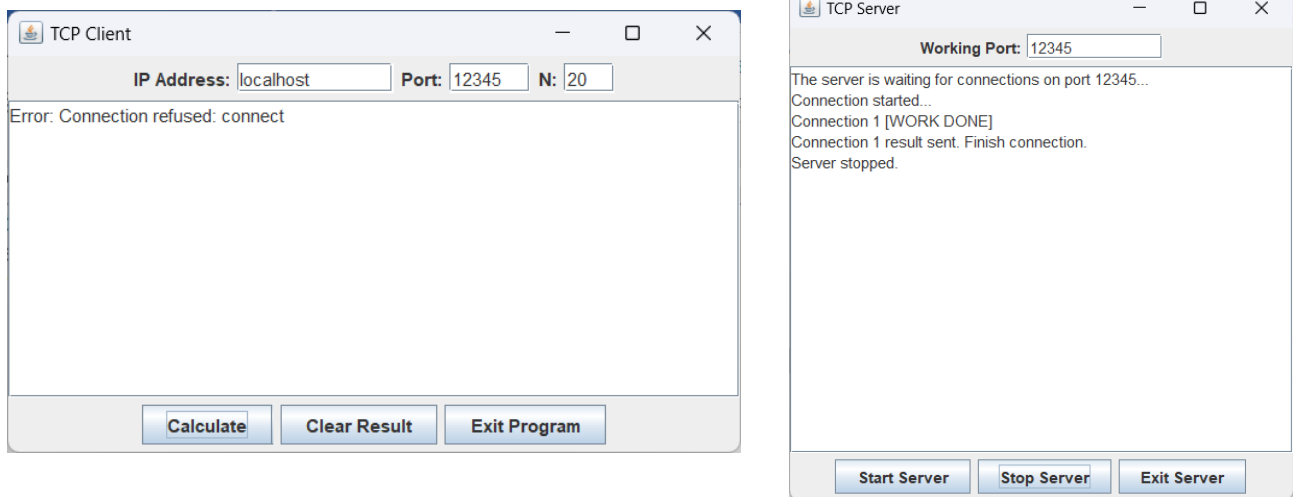


Рисунок 5 – Спроба порахувати факторіал при зупинці сервера

Завдання №1 (файли UDPClient, UPDServer, ActiveUsers, User у репозиторії)

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 перевіряють роботу сервера: відправляють запит, який реєструє клієнта (комп'ютер) на сервері, і отримують відповідь сервера — список вже зареєстрованих комп'ютерів.

**Результати виконання завдання №1 :**

```
Server start...
Request from: 127.0.0.1 port: 57125
Request from: 127.0.0.1 port: 57126
Request from: 127.0.0.1 port: 57127
Request from: 127.0.0.1 port: 57128
```

Рисунок 6 – Серверна частина

```
Client with port 57125 sending request
      |
      V
Registered User: Host name: 127.0.0.1      Host address: 127.0.0.1 Port: 57125
-----
Client with port 57126 sending request
      |
      V
Registered User: Host name: 127.0.0.1      Host address: 127.0.0.1 Port: 57126
-----
Client with port 57127 sending request
      |
      V
Registered User: Host name: 127.0.0.1      Host address: 127.0.0.1 Port: 57127
-----
Client with port 57128 sending request
      |
      V
Registered User: Host name: 127.0.0.1      Host address: 127.0.0.1 Port: 57128
-----

Process finished with exit code 0
```

Рисунок 7 – Клієнтська частина