

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ В. Н. КАРАЗІНА
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ

ЛАБОРАТОРНА РОБОТА № 5
З дисципліни «Крос-платформне програмування»
з теми «Java Net Programming (UDP Sockets)»

Виконав:

студент 2 курсу, групи КС24

Бондаренко Олег

Перевірив:

Споров Олександр Євгенович

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 реалізувати інтерфейс, визначений відповідно до договору з сервером). Визначення класу завдання відправляється клієнтом на сервер і, як тільки class - файл стає доступний, сервер може виконувати отримане завдання. Аналогічно, сервер створює об'єкт класу результату і відправляє його разом з визначенням класу клієнту. При цьому клас результату реалізує інтерфейс, відомий клієнту. В якості клієнтського завдання можна взяти завдання обчислення факторіала досить великого числа.

Завдання №2

Розглянемо взаємодію двох комп'ютерів в мережі. Перед тим, як почати роботу, комп'ютери повинні обмінятися IP-адресами. Це може стати досить непростим завданням. Створимо спеціальний UDP сервер, який допоможе комп'ютерів обмінятися «Координатами»: IP-адресами і номерами портів. Потім створимо UDP клієнтів, які перевіряють роботу сервера: відправляють запит, який реєструє відправив комп'ютер на сервері і отримують відповідь сервера - список вже зареєстрованих комп'ютерів.

Виконання роботи

Завдання №1

Інтерфейси Result (реалізують класи, які дають результат) та Executable (класи, які виконують завдання(факторіал)).

Пакет server відповідає серверній частині: класи Connection, ResultImpl, ServerFrame, Start.

Пакет client відповідає клієнтській частині: класи ClientFrame, Factorial.

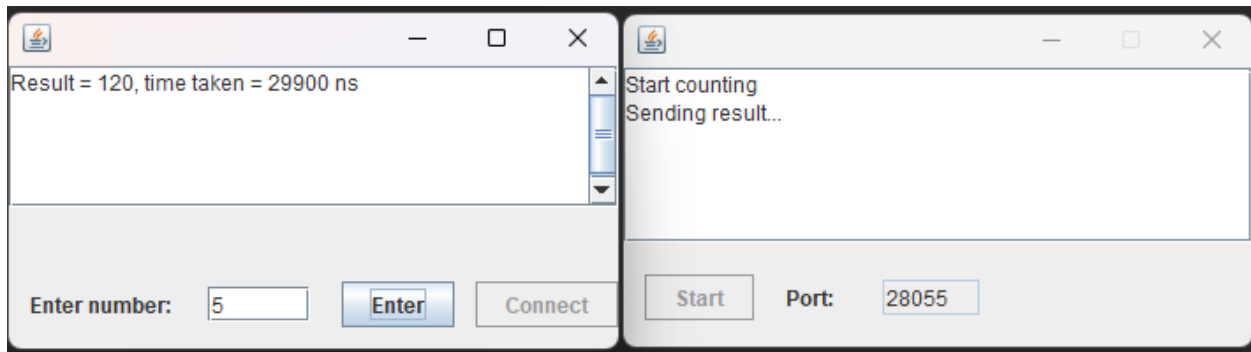


Рисунок 1. Завдання 1

Завдання №2

Клас User(клас призначений для зберігання інформації Інтернет адреса зареєстрованого комп'ютера. Реалізує інтерфейс Serializable), клас ActiveUsers(зберігає інформацію про всіх зареєстрованих на даному сервері комп'ютерах), клас UPDServer(представляє сервер UPD), UPDClient(представляє клієнти UPD).

```
Do you want to shutdown the server after receiving data? 1.yes/ 2.no ->2
Sending a request to the server... Request sent

Getting a list of users.
User list received

Online users: 1
    Address: DESKTOP, port: 64344

Process finished with exit code 0
```

Рисунок 2. Клас UPDClient

```
The server is running..

Waiting for a new connection... User is connected
User registration... User 192.168.1.107 with port: 64344 registered
Sending an answer...The answer sent

Waiting for a new connection...
```

Рисунок 3. Клас UPDServer