

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

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.

```
C:\Users\User\.jdk\openjdk-19.0.2\bin\java.exe "-javaagent:D:\J...
Server started
Enter 0 to end the server or another num to continue:
1
New Client Waiting...
```

Рисунок 1 - запускаємо сервер

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

```
C:\Users\User\.jdk\openjdk-19.0.2\bin\java.exe "-javaagent:D:\J...
Server started
Enter 0 to end the server or another num to continue:
1
New Client Waiting...
New client: Socket[addr=/127.0.0.1,port=61524,localport=7891]
Enter 0 to end the server or another num to continue:
|
```

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

```
C:\Users\User\.jdk\openjdk-19.0.2\bin\java.exe "-javaagent:D:\J...
Enter the number
15
result = 1307674368000, time taken = 25800.0ns

Process finished with exit code 0
```

```
Enter the number
20
result = 2432902008176640000, time taken = 2299.0ns

Process finished with exit code 0
```

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

```
New Client Waiting...
New client: Socket[addr=/127.0.0.1,port=61539,localport=7891]
Enter 0 to end the server or another num to continue:
0
Server closed.

Process finished with exit code 0
```

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.

```
C:\Users\User\.jdk\openjdk-19.0.2\bin\java.exe
Server start...
|
```

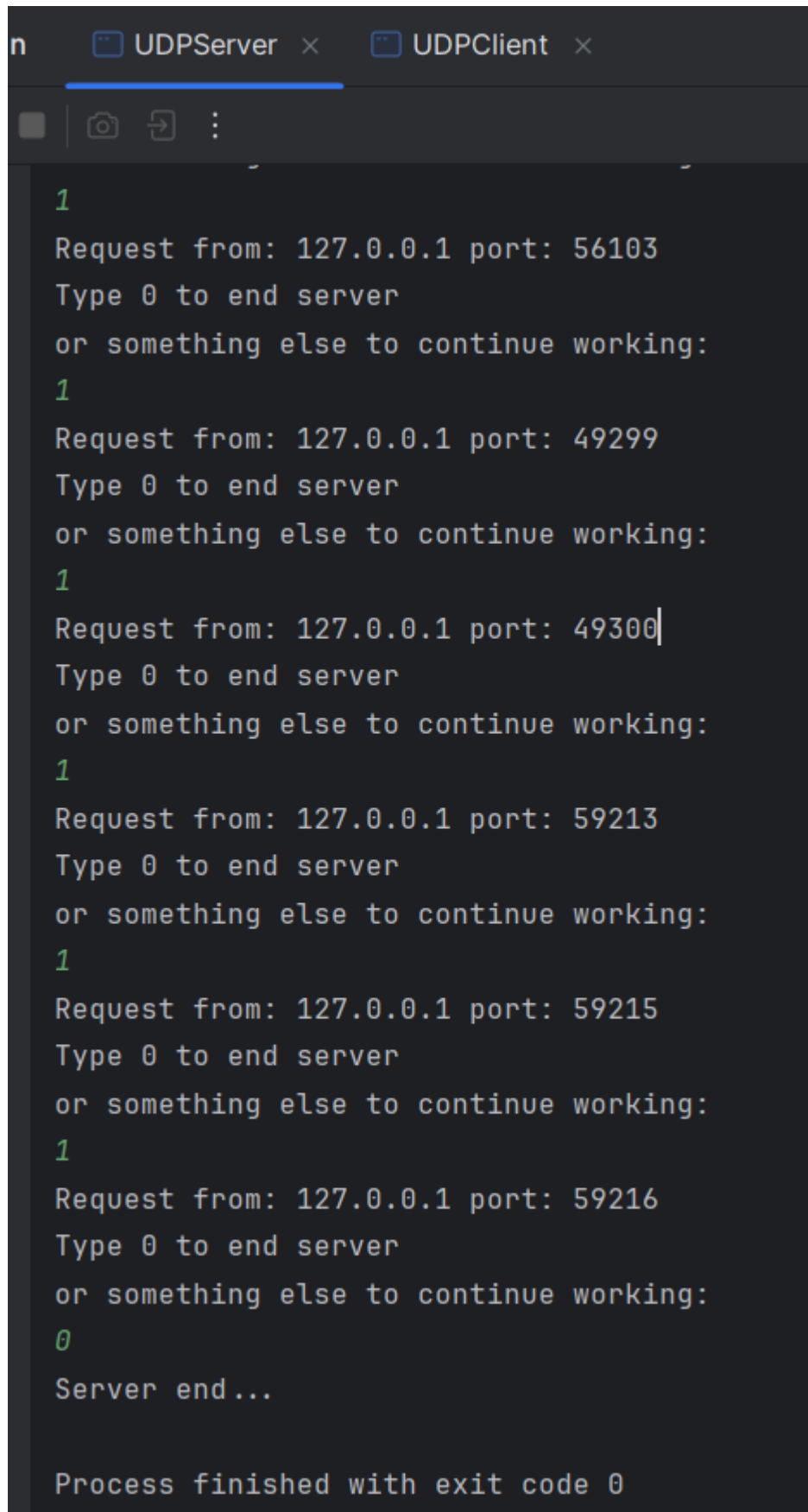
Рисунок 6 – запускаємо сервер

```
UDPServer x UDPCliant x
C:\Users\User\.jdk\openjdk-19.0.2\bin\java.exe
Sending request
Registered users: 1
User{address=/127.0.0.1, port=59144}
```

Рисунок 7 - запускаємо клієнта

```
UDPServer x UDPCliant x
C:\Users\User\.jdk\openjdk-19.0.2\bin\java.exe
Server start...
Request from: 127.0.0.1 port: 59144
Type 0 to end server
or something else to continue working:
|
```

Рисунок 8 - інформація на сервері



```
n  UDPServer x  UDPClient x
1
Request from: 127.0.0.1 port: 56103
Type 0 to end server
or something else to continue working:
1
Request from: 127.0.0.1 port: 49299
Type 0 to end server
or something else to continue working:
1
Request from: 127.0.0.1 port: 49300
Type 0 to end server
or something else to continue working:
1
Request from: 127.0.0.1 port: 59213
Type 0 to end server
or something else to continue working:
1
Request from: 127.0.0.1 port: 59215
Type 0 to end server
or something else to continue working:
1
Request from: 127.0.0.1 port: 59216
Type 0 to end server
or something else to continue working:
0
Server end...

Process finished with exit code 0
```

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

```
C:\Users\User\.jdk\openjdk-19.0.2\bin\java.  
Sending request  
Registered users: 7  
User{address=/127.0.0.1, port=59144}  
User{address=/127.0.0.1, port=56103}  
User{address=/127.0.0.1, port=49299}  
User{address=/127.0.0.1, port=49300}  
User{address=/127.0.0.1, port=59213}  
User{address=/127.0.0.1, port=59215}  
User{address=/127.0.0.1, port=59216}  
  
Process finished with exit code 0
```

Рисунок 10 – результати виконання програми на клієнській частині