Київський національний університет

імені Т.Шевченка

**Звіт**

до лабораторної роботи №1

на тему:

«*Взаємодія розподілених процесів через механізм сокетів****»***

***Студента другого курсу***

***Групи К-26***

***Факультету комп’ютерних наук***

***та кібернетики***

***Черненка Євгенія Олександровича***

***Київ-2022***

**Мета**

В даній лабораторній роботі необхідно освоїти механізм (технологію) сокетів стеку протоколів TCP/IP, зокрема його реалізацію в MS Windows. Індивідуальний варіант роботи полягає в розробці двох програм (клієнта та сервера, які запускаються на різних станціях мережі), розробці протоколу обміну даними між ними та демонстрації роботи програм.

Індивідуальний варіант – 18 «Гра "вгадування 4-значного числа"»: «Користувач на клієнті вгадує 4-значне ціле, яке зберігається на сервері. З клієнта передаються 4 цифри, на які сервер дає відповідь із двох цифр: кількість правильних цифр та кількість цифр на своїх місцях. Клієнт в ході гри може її завершити, почати нову гру, завершити сеанс. Клієнт може здатися і тоді сервер розкриває загадане число.. Користувач на клієнті може вибирати спосіб задання спроб: в діалозі вводити самому чи автоматична генерація випадкових 4-х значних чисел в заданій кількості.»

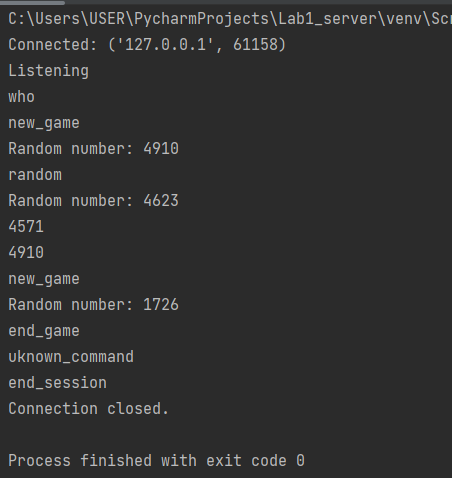
**Основні принципи виконання роботи**

Нижче подані основні принципи написання, не вдаючись до детального розбору коду, який зроблений в розділі «Код програми» у вигляді коментарів до поданого програми:

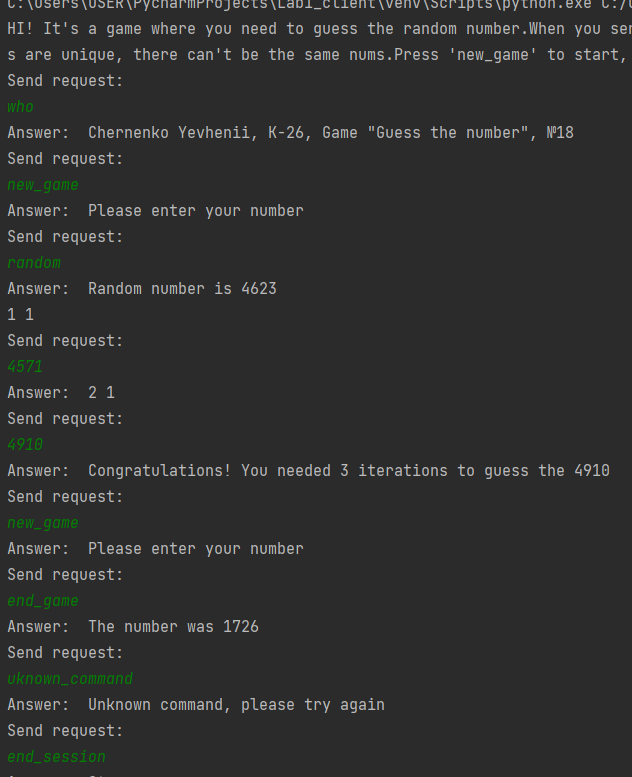
* Програма представлена у вигляді консольного додатку. Спочатку запускається сервер, а потім запускається клієнт. З серверу до клієнту передається привітальне повідомлення та правила гри.
* Програма написана мовою програмування Python з використанням модулів logging та socket. Для сервера додатково був використаний модуль random для генерації випадкового числа.
* Клієнт та сервер обмінюються даними закодованими в UTF-8. Розмір повідомлень (даних) до 255 байт.
* Реалізовано команди: who (клієнт отримує інформацію про студента, який розробив програму), new\_game (розпочати нову гру), end\_game (достроково закінчити гру), end\_session (закінчення ігрової сесії), random (генерація числа на сервері, передання його як введеного числа).
* Програма-клієнт та програма-сервер реалізовані для прослуховування сокету у локальній мережі на адресі “127.0.0.1” та на порті: 1025+18 = 1043.
* Реалізовано реакцію на невідомі команди, при введенні такої команди клієнт отримає повідомлення "Unknown command, please try again".
* Реалізовано запис подій у системний для клієнта та серверу. Формат запису: Час – Команда .
* Отримання повідомлень реалізовано у нескінченому циклі: отримуємо повідомлення,
* Функція-обробник перевіряє правильність та повертає повідомлення та дані, щодо кількості ітерацій в грі. У разі відгадування числа виводиться кількість ітерацій.
* Прослуховування сокету завершується командою end\_session, тому сервер повинен отримати команду, інакше сокет буде занятий, і сервер не зможе розпочати роботу, коли його запустять.

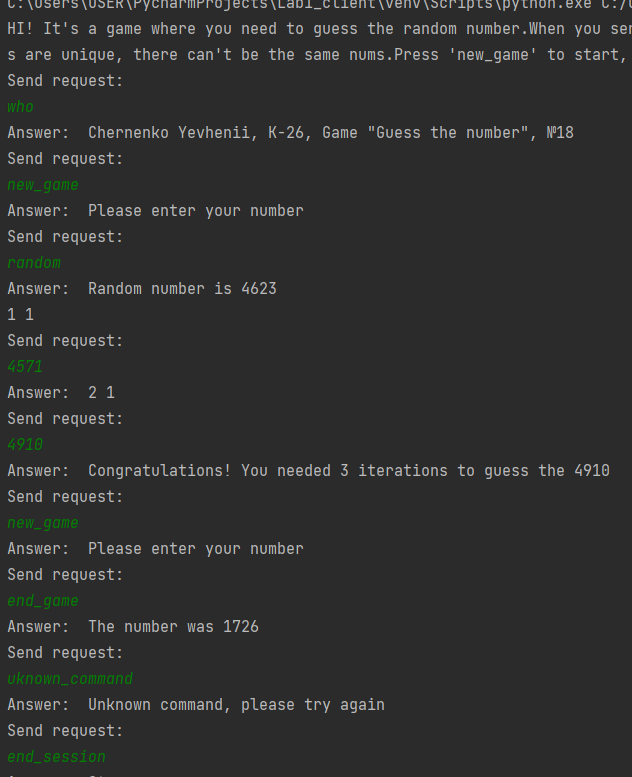
**Лістинг програми**

Сервер:



Клієнт:





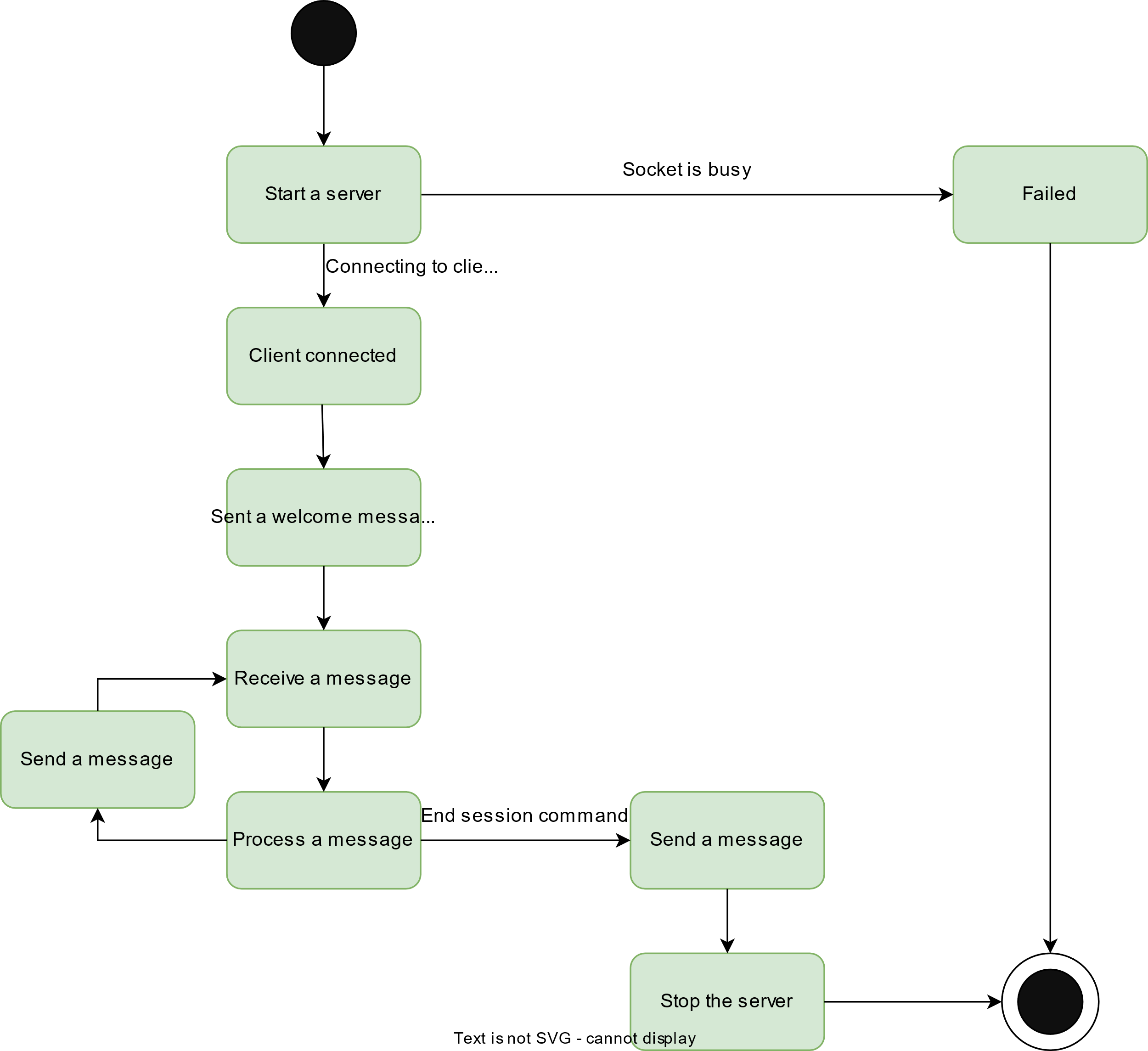


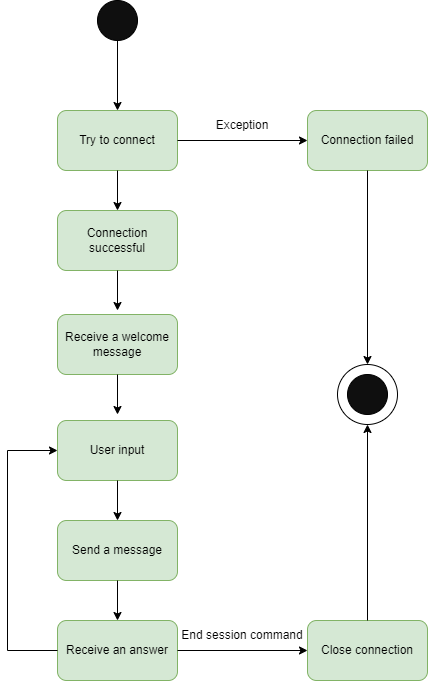
**Висновок**

В ході виконання лабораторної роботи було написано дві програми: програма-сервер та програма-клієнт, завдяки цьому я вивчив та практично освоїв механізм роботи сокетів стеку протоколів TCP/IP.

Використання даної технології дозволяє легко з’єднати два пристрої між собою через мережу Інтернет, адже майже на кожному пристрої з великою ймовірністю буде наперед інстальована програма базована на Berkley Sockets API.

Так як для встановлення з’єднання між пристроями потрібно лише те, що б сервер був ввімкнений раніше клієнта, то це робить сокети оптимальним процесом для взаємодії розподілених процесів .

**Додаток 1. Діаграма станів серверу**

**Додаток 2. Діаграма станів клієнту**

**Додаток 3. Код програми серверу**

|  |
| --- |
| **import** socket  **import** random  **import** logging  **def** generateNewNumber():  s = ""  **while** len(s) != 4:  s1 = str(random.randint(0, 10))  **if** s1 **not** **in** s:  s += s1  **print**("Random number: " + s)  **return** s  **class** Server:  **def** \_\_init\_\_(self):  **try**:  HOST = '127.0.0.1'  PORT = 1025 + 18  utf = 'utf-8'  logging.getLogger("Server")  logging.basicConfig(filename="myServer.log",  level=logging.INFO,  format='%(asctime)s - %(levelname)s - %(message)s',  filemode='a',  )  logging.info("Server started")  sock = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)  counter = 0  **try**:  sock.bind((HOST, PORT))  **except** OSError:  **print**("Host is used.")  logging.warning("HOST IS USED. ")  exit()  sock.listen(1)  connection, client\_address = sock.accept()  **print**("Connected:", client\_address)  logging.info("Connected to a client. ")  **for** sentence **in** self.welcome\_message():  connection.send(sentence.encode(utf))  logging.info("Sent a welcome message.")  **print**("Listening")  logging.info('Listening to a client.')  isActiveGame = False  **try**:  **while** True:  data = connection.recv(256).decode(utf)  logging.info(f'Received message: {data}')  **print**(data)  **if** **not** data:  **break**  **if** data == "new\_game":  **if** isActiveGame:  results = "You haven't finished previous game"  **else**:  rand\_num = generateNewNumber()  isActiveGame = True  results = "Please enter your number"  logging.info("Start of a new game ")  **elif** data == "end\_game":  **if** isActiveGame:  isActiveGame = False  results = "The number was " + rand\_num  logging.info("User ended a game ")  **else**:  results = "The game hasn't been started"  **elif** data == "end\_session":  **if** isActiveGame:  results = "The number was " + rand\_num + ". Stop"  **else**:  results = "Stop"  connection.send(results.encode(utf))  logging.info("Sent message: " + results)  logging.info("The end of session")  **break**  **elif** data.\_\_contains\_\_("random") **and** isActiveGame:  rand = generateNewNumber();  answer1, answer2, counter = self.process(rand, rand\_num, counter)  **if** answer1 **is** None **or** answer2 **is** None:  logging.info("Process function broke at some moment.")  **raise** IOError  **if** (answer1, answer2) == (4, 4):  results = "Congratulations! You needed " + str(  counter) + " iterations to guess the " + str(rand\_num)  isActiveGame = False  **else**:  results = "Random number is " + str(rand) + '\n' + str(answer1) + " " + str(answer2)  **elif** data == "who":  results = self.whoami()  **elif** data.isnumeric() **and** len(data) == 4 **and** isActiveGame:  answer1, answer2, counter = self.process(data, rand\_num, counter)  **if** answer1 **is** None **or** answer2 **is** None:  logging.info("Process function broke at some moment.")  **raise** IOError  **if** (answer1, answer2) == (4, 4):  results = "Congratulations! You needed " + str(counter) + " iterations to guess the " + str(rand\_num)  isActiveGame = False  **else**:  results = str(answer1) + " " + str(answer2)  **else**:  results = "Unknown command, please try again"  connection.send(results.encode(utf))  logging.info("Sent message: " + results)  **except**:  **print**("ERROR.")  sock.send("STOP".encode(utf))  logging.error("Caught en error. Closing connection.")  **finally**:  **try**:  sock.close()  **except** Exception **as** e:  **print**(f' !Exception {e}')  logging.warning(f'{e}')  **print**("Connection closed.")  **def** process(self, data: str, rand\_number: str, count: int):  **try**:  counter1 = 0  counter2 = 0  **for** a **in** data:  **if** a **in** rand\_number:  counter1 += rand\_number.count(a)  l1 = list(data)  l0 = list(rand\_number)  **for** i **in** range(0, 4):  **if** l1[i] == l0[i]:  counter2 += 1  count += 1  **return** counter1, counter2, count  **except** Exception **as** e:  **return** f"={e}"  **def** welcome\_message(self):  welcome = [  "HI! It's a game where you need to guess the random number.",  "When you send the random number you will get two numbers like an answer, the 1st num is how much correct",  "numbers there are, and the 2nd one is how much nums are on correct place.",  "Remember that all nums are unique, there can't be the same nums.",  "Press 'new\_game' to start, 'end\_game' to stop game, and 'end\_session' to stop the program. "  ]  **return** welcome  **def** whoami(self):  who = 'Chernenko Yevhenii, K-26, Game "Guess the number", №18'  **return** who  **def** main():  root = Server()  **if** \_\_name\_\_ == "\_\_main\_\_":  main() |

**Додаток 4. Код програми клієнта**

|  |
| --- |
| **import** socket  **import** logging  HOST = '127.0.0.1'  PORT = 1025+18  **class** Client:  **def** \_\_init\_\_(self):  **try**:  sock = socket.socket(  socket.AF\_INET,  socket.SOCK\_STREAM,  )  logging.getLogger("Client")  logging.basicConfig(filename="myClient.log",  level=logging.INFO,  format='%(asctime)s - %(levelname)s - %(message)s',  filemode='a',  )  logging.info('Client started.')  sock.connect((HOST, PORT))  logging.info("Connected to a server.")  **while** True:  data = sock.recv(256).decode('utf-8')  **print**(data)  **if** data.\_\_contains\_\_("to stop the program"):  **break**  logging.info('Got welcome message.')  **while** True:  data = input('Send request:\n')  sock.send(data.encode('utf-8'))  logging.info('Sent request: ' + data)  data = sock.recv(256).decode('utf-8')  logging.info('Received message: ' + data)  **print**('Answer: ', data)  **if** 'Stop' **in** data **or** 'STOP' **in** data:  sock.close()  logging.info("End of session")  **break**  **print**('Connection closed.')  logging.info('Connection closed.')  **except** Exception **as** e:  **print**(f'Happened exception: {e}')  logging.info('Caught an error. Closing connection.')  @staticmethod  **def** reconnect():  **while** True:  answer = input("Do you want to try to reconnect? Y/n: ")  **if** answer.lower() **in** 'yn':  **break**  **if** answer.lower() == 'y':  **return** True  **else**:  **return** False  **def** main():  client = Client()  **if** \_\_name\_\_ == "\_\_main\_\_":  main() |

**Додаток 5. Системні журнали серверу та клієнту**

|  |  |
| --- | --- |
| 2022-10-20 11:14:35,563 - INFO - Server started  2022-10-20 11:14:36,954 - INFO - Connected to a client.  2022-10-20 11:14:36,954 - INFO - Sent a welcome message.  2022-10-20 11:14:36,954 - INFO - Listening to a client.  2022-10-20 11:14:42,963 - INFO - Received message: who  2022-10-20 11:14:42,964 - INFO - Sent message: Chernenko Yevhenii, K-26, Game "Guess the number", №18  2022-10-20 11:15:44,685 - INFO - Server started  2022-10-20 11:15:45,408 - INFO - Connected to a client.  2022-10-20 11:15:45,408 - INFO - Sent a welcome message.  2022-10-20 11:15:45,410 - INFO - Listening to a client.  2022-10-20 11:15:57,839 - INFO - Received message: who  2022-10-20 11:15:57,840 - INFO - Sent message: Chernenko Yevhenii, K-26, Game "Guess the number", №18  2022-10-20 11:16:07,435 - INFO - Received message: new\_game  2022-10-20 11:16:07,435 - INFO - Start of a new game  2022-10-20 11:16:07,435 - INFO - Sent message: Please enter your number  2022-10-20 11:16:12,500 - INFO - Received message: random  2022-10-20 11:16:12,501 - INFO - Sent message: Random number is 4623  1 1  2022-10-20 11:16:17,685 - INFO - Received message: 4571  2022-10-20 11:16:17,685 - INFO - Sent message: 2 1  2022-10-20 11:16:21,485 - INFO - Received message: 4910  2022-10-20 11:16:21,486 - INFO - Sent message: Congratulations! You needed 3 iterations to guess the 4910  2022-10-20 11:16:26,707 - INFO - Received message: new\_game  2022-10-20 11:16:26,708 - INFO - Start of a new game  2022-10-20 11:16:26,708 - INFO - Sent message: Please enter your number  2022-10-20 11:16:29,246 - INFO - Received message: end\_game  2022-10-20 11:16:29,246 - INFO - User ended a game  2022-10-20 11:16:29,247 - INFO - Sent message: The number was 1726  2022-10-20 11:16:39,693 - INFO - Received message: uknown\_command  2022-10-20 11:16:39,693 - INFO - Sent message: Unknown command, please try again  2022-10-20 11:16:49,359 - INFO - Received message: end\_session  2022-10-20 11:16:49,359 - INFO - Sent message: Stop  2022-10-20 11:16:49,359 - INFO - The end of session | 2022-10-20 11:14:36,952 - INFO - Client started.  2022-10-20 11:14:36,953 - INFO - Connected to a server.  2022-10-20 11:14:36,954 - INFO - Got welcome message.  2022-10-20 11:14:42,963 - INFO - Sent request: who  2022-10-20 11:14:42,964 - INFO - Received message: ' to start, 'end\_game' to stop game, and 'end\_session' to stop the program.  2022-10-20 11:15:45,406 - INFO - Client started.  2022-10-20 11:15:45,410 - INFO - Connected to a server.  2022-10-20 11:15:45,410 - INFO - Got welcome message.  2022-10-20 11:15:57,839 - INFO - Sent request: who  2022-10-20 11:15:57,840 - INFO - Received message: Chernenko Yevhenii, K-26, Game "Guess the number", №18  2022-10-20 11:16:07,434 - INFO - Sent request: new\_game  2022-10-20 11:16:07,436 - INFO - Received message: Please enter your number  2022-10-20 11:16:12,500 - INFO - Sent request: random  2022-10-20 11:16:12,501 - INFO - Received message: Random number is 4623  1 1  2022-10-20 11:16:17,684 - INFO - Sent request: 4571  2022-10-20 11:16:17,687 - INFO - Received message: 2 1  2022-10-20 11:16:21,485 - INFO - Sent request: 4910  2022-10-20 11:16:21,486 - INFO - Received message: Congratulations! You needed 3 iterations to guess the 4910  2022-10-20 11:16:26,707 - INFO - Sent request: new\_game  2022-10-20 11:16:26,709 - INFO - Received message: Please enter your number  2022-10-20 11:16:29,246 - INFO - Sent request: end\_game  2022-10-20 11:16:29,248 - INFO - Received message: The number was 1726  2022-10-20 11:16:39,693 - INFO - Sent request: uknown\_command  2022-10-20 11:16:39,693 - INFO - Received message: Unknown command, please try again  2022-10-20 11:16:49,359 - INFO - Sent request: end\_session  2022-10-20 11:16:49,359 - INFO - Received message: Stop  2022-10-20 11:16:49,360 - INFO - End of session  2022-10-20 11:16:49,360 - INFO - Connection closed. |