САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ПРОЦЕССОВ УПРАВЛЕНИЯ

ОТЧЕТ

по лабораторной работе №3

по дисциплине «Функциональное программирование»

на тему «разработка асинхронного чат-сервера»

Студент гр. 22Б15 _____ Суворов Н.В.

Преподаватель _____ Киямов Ж.У.

Санкт-Петербург
2023 г.

Оглавление

1. Цель

Создать асинхронный чат-сервер, способный обслуживать множество клиентов одновременно и обеспечивать в реальном времени обмен сообщениями.

2. Задача:

- Разработать асинхронный сервер с использованием библиотеки asyncio, прослушивающий определенный порт для входящих подключений.
- Написать асинхронный клиентский скрипт, способный подключаться к серверу и отправлять сообщения.
- Обеспечить возможность одновременного подключения нескольких клиентов с помощью asyncio-задач.
- Реализовать поддержку чат-комнат, где пользователи могут выбирать, в какую комнату они хотят войти.
- Реализовать рассылку сообщений от одного клиента всем участникам комнаты.
- Использовать asyncio.Queue для обработки асинхронных событий, таких как новые сообщения от клиентов.
- Обработать исключения для предотвращения остановки сервера из-за неправильного поведения клиента.
- Провести тестирование сервера, включая единичное тестирование асинхронных функций.
- Добавить дополнительные функции по необходимости, например, возможность загрузки файлов или отправки личных сообщений.

3. Теория:

- В ходе выполнения задачи была использована библиотека asyncio для асинхронного программирования, позволяющего обрабатывать множество клиентов одновременно.
- Сервер прослушивает определенный порт и создает отдельную задачу для каждого подключившегося клиента, обрабатывая их сообщения асинхронно.
- Для обмена сообщениями между клиентами и сервером была использована асинхронная очередь asyncio.Queue.

4. Входные данные и решение

Github
https://github.com/AlexShinalov/functional_programming/tree/main/laba%203

Таблица 4.1 Классы

Классы	Назначение
designer	Графический интерфейс
client	Клиентская часть
server	Серверная часть

Таблица 4.2 Функции

Имя	Описание
__init__	Инициализация переменных, настройка параметров интерфейса, запуск корректировки интерфейса
start server	Запускает сервер
exit	Выход из комнаты
enter room	Вход в комнаты
create room	Создает комнату
send_messege	Отправляет сообщение
recive_messege	Принимает сообщение
handle_client	Связывает интерфейс и сервер
error	Детекция ошибок

5. Рекомендации программиста

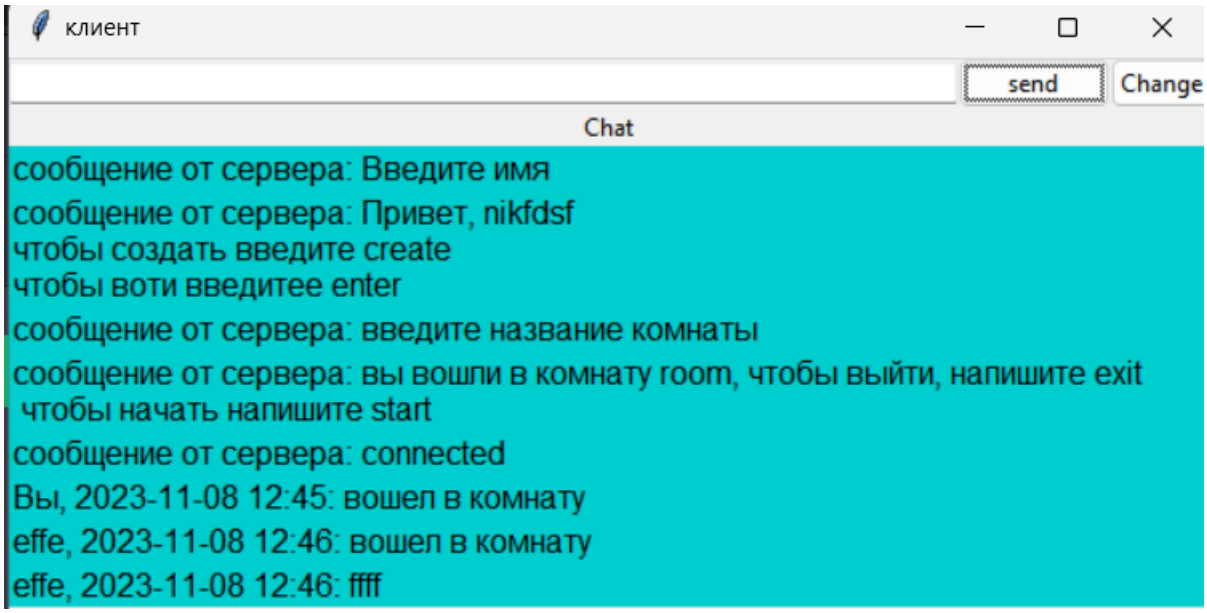
Для запуска программы необходима 64-битная операционная система Windows и Python версии не ниже 3.1. Для корректной работы программы рекомендуется использовать IDE PyCharm версии 2023.21 и pip install версии 23.1.0.

6. Рекомендации пользователю

Запустите программу, создайте комнату или выберите ту, к которой хотите присоединиться. При наличии ошибок проверьте консоль, там будут выведен все возможные проблемы

7. Пример

На картинке мы можем видеть giu-клиент чат-сервера. В нем есть команды create, enter, start и exit. Кнопки send и change theme. В экране ниже приходят сообщения от пользователей.



8. Вывод

В результате выполнения задачи был разработан асинхронный чат-сервер, способный обслуживать множество клиентов одновременно. Реализованы функции поддержки чат-комнат, рассылки сообщений и обработки исключений для надежной работы сервера.

9. Литература

https://docs.python.org/3/library/asyncio.html

https://www.python.org/