

**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. 22Б16

Шувалов Ф.В.

Преподаватель

Киямов Ж.У.

Санкт-Петербург
2023 г.

Оглавление

1. Элементы оглавления не найдены.

1. Цель

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

2. Задачи:

- Разработайте асинхронный сервер, используя библиотеку `asyncio`, прослушивающий определенный порт для входящих подключений. Напишите асинхронный клиентский скрипт, способный подключаться к серверу и отправлять сообщения. Обеспечьте возможность одновременного подключения нескольких клиентов с использованием `asyncio`-задач.
- Реализуйте поддержку чат-комнат, где пользователи могут выбирать, в какую комнату они хотят войти. Добавьте функционал рассылки сообщений от одного клиента всем участникам комнаты. Используйте `asyncio.Queue` для обработки асинхронных событий, таких как новые сообщения от клиентов.
- Обработайте исключения для предотвращения остановки сервера из-за неправильного поведения клиента. Проведите тестирование сервера, включая единичное тестирование асинхронных функций. По необходимости добавьте дополнительные функции, такие как 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

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
Create_and_enter_room	Создание и вход в комнату
Enter_existing_room	Вход в уже созданную комнату
send_messege	Отправляет сообщение
recive_messege	Принимает сообщение
handle_client	Связывает интерфейс и сервер
error	Детекция ошибок
Send_Silent_message	Отправка анонимного сообщения
Send_direct_message	Отправка прямого сообщения
Send_Help_message	Отправка сообщения с командами
Send_log_message	Отправка логов

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.

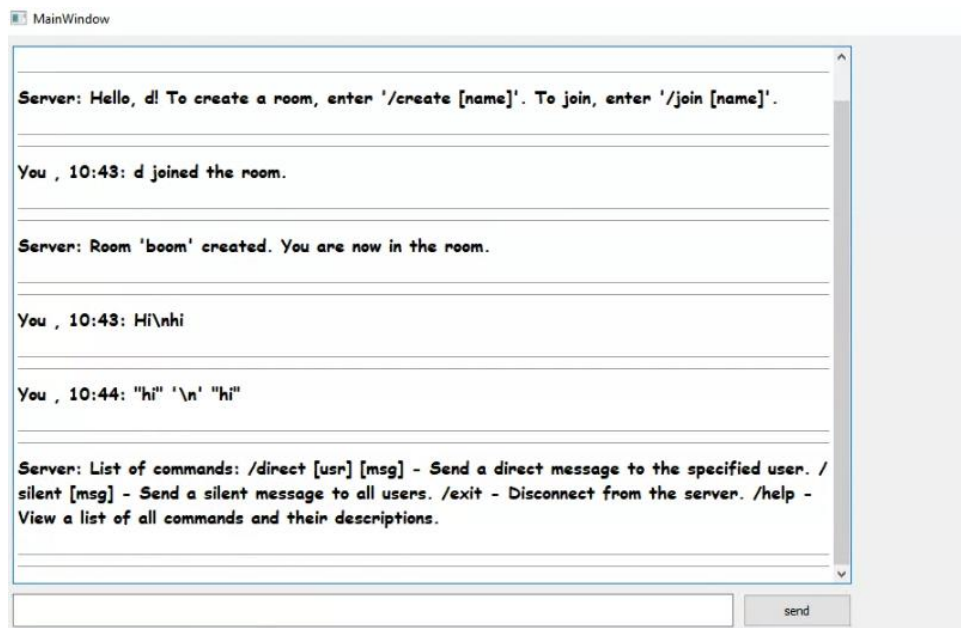
Код можно найти по ссылке <https://github.com/zer0rbt/FP>

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 gui-клиент чат-сервера. В нем есть команды /create, /enter, /start и /exit. После входа в комнату появляются команды /help, /direct, /silent. На экране ниже отображаются 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/>

