**ПРОЕКТ**

**PyLocalChat –**

**чат в локальной сети**

**Рожок Владимир Дмитриевич,**

**Ученик 7 класса «В» ГБОУ Школа №192**

**г. Москва**

**2023**

Оглавление

[Введение 2](#_Toc128861402)

[Цель и задачи работы 2](#_Toc128861403)

[Выполнение работы 2](#_Toc128861404)

[Описание работы 3](#_Toc128861405)

[Список используемой литературы 4](#_Toc128861406)

## Введение

Работа посвящена созданию программы для организации многопользовательского чата в локальной сети.

Эта работа актуальна для секретного общения. Она может широко использоваться как скрытый мессенджер для надёжной передачи информации через локальную сеть.

Я выбрал эту тему для своего обучения и т.к. посчитал программу полезной в использовании. Здесь меня привлекла работа с файлами и обмен информацией между устройствами.

## Цель и задачи работы

Целью было создание программы на Python 3, которая обеспечит безопасный и конфиденциальный чат, имеющий систему авторизации с разделением на группы пользователей (такие как админы и обычные пользователи) с разными правами. Например, право на удаление чужих аккаунтов имеют только админы.

Задачи: сделать систему авторизации с сохранением учётных записей и паролей к ним, сервер чата и клиента к нему, встроить клиента в систему авторизации, чтобы всё работало.

## Выполнение работы

Для написания данной программы использовались такие модули как *socket, datetime, time, sys, hashlib*.

Socket: нужен для создания соединений сервера и клиента.

Time: использовался для создания задержки для красивого вывода информации.

Datetime: нужен для создания логов, в которых пишется точное время, дата и событие.

SYS: использовался для перевода информации в байты. Это нужно для отправки информации через сокеты.

Hashlib: требуется для шифровки информации. В программе играет роль шифровщика паролей. Так как хэш необратим

Код писался в *Python* *IDLE*. Срок выполнения: С ноября 2022 по март 2023 гг.

## Описание работы

Написаны: системы входа, QuickL входа, меню, регистрации, удаления аккаунтов, смены пароля, смены типа аккаунта. Также в самой программе есть встроенный клиент, а сам сервер надо запускать вручную. Сейчас я объясню как работает каждая система по подробнее.

Для **регистрации** система запрашивает у пользователя логин и пароль, пароль шифруется методом “hashlib.md5” и записывается вместе с логином и типом аккаунта (при регистрации пользователь) в файл базы аккаунтов.

Как я уже упоминал, систем входа две – обычная и QuickL. После регистрации используется **обычная система входа**:

Система запрашивает логин и пароль у пользователя, а тип аккаунта «Пользователь» система ставит сама. Для того, чтобы зайти как админ, нужно при запуске функции start() для вызова основного меню в скобках написать (‘admin’), тем самым сказав системе, чего именно мы хотим. Но в любом случае система будет проверять каждый из параметров. Для того, чтобы проверить пароль, система шифрует его тем же методом и сравнивает с ранее записанным в файле, предварительно нашедши нужную строку по логину.

После того, как система убедилась, что аккаунт с таким именем и паролем существует, она просит пользователя ввести новый QuickL код, который в будущем поможет нам войти. Затем она записывает тип аккаунта, под которым вошел пользователь, его логин и зашифрованный QuickL код в файл и в переменную аутентификации.

**Система входа QuickL -** Система входа без логина и пароля, использующая код, который пользователь вводил при последнем входе через **обычную** систему входа. Запускается вместо обычной, если в файле аутентификации есть информация и если в переменной аутентификации нет информации, то есть пользователь зашёл, но не вышел, а программа перезапустилась. Если в файле аутентификации ничего нет, то запускается обычная система входа. Система QuickL запрашивает у пользователя код. Она его шифрует и сравнивает с тем, который находится в файле аутентификации. Если коды совпадают, то система записывает в переменную аутентификации логин и тип аккаунта пользователя, которые были записаны в файле.

**Система выхода** очень простая: она стирает всю информацию из переменной и файла аутентификации.

**Система запуска меню** (она же **start**) запускает разные системы входа в зависимости от обстоятельств. Она допускает к меню аккаунта только в том случае, если переменная и файл аутентификации не пустые и данные в них совпадают.

**Системы** **удаления аккаунтов**, **смены пароля**, **смены типа аккаунта** на самом деле очень похожи. Они все перезаписывают файл базы аккаунтов, находя нужную строку по логину и после изменения, либо удаления строки, перезаписывают файл.

**Клиент** работает по типу: подключился, получил информацию, обработал, напечатал, запросил н пользователя сообщение, обработал, отправил, закрыл соединение.

Код программы: <https://github.com/DaregonPL/PyLC>

## [Список используемой литературы](http://sumina-yuliya.ru/dokument_sait/knigi/Dzheyson_Briggs_Python_dlya_detey.pdf)

1. Джейсон Бриггс / [**Python для детей**](http://sumina-yuliya.ru/dokument_sait/knigi/Dzheyson_Briggs_Python_dlya_detey.pdf) // Москва, «Манн, Иванов и Фербер», 2017.

Москва, 2023

Рожок Владимир Дмитриевич