МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

Сибирский государственный университет телекоммуникаций и информатики

СибГУТИ

ОТЧЕТ

по дисциплине *Программирование*

по теме:

Клавиатурный тренажер

Студент: *Воинова Т. А.*

*Группы ИКС-432*

Преподаватель: Вейлер В. А.

Новосибирск, 2025

Содержание

Оглавление

[Содержание 2](#_Toc199191329)

[Введение 3](#_Toc199191330)

[Основные структуры данных 4](#_Toc199191331)

[Алгоритмы работы программы 5](#_Toc199191332)

[Технические аспекты 5](#_Toc199191333)

[Работа программы 6](#_Toc199191334)

[Заключение 10](#_Toc199191335)

[Ссылки на ресурсы 10](#_Toc199191336)

Введение

Клавиатурный тренажер - это программное обеспечение, предназначенное для обучения быстрой и безошибочной печати на компьютере. В современном мире, где компьютерные технологии проникли во все сферы жизни, навык слепой печати становится важным как для программистов, так и для офисных работников, студентов и других пользователей. Разработка клавиатурного тренажера на языке C позволяет не только изучить основы программирования, но и понять принципы обработки пользовательского ввода, работы с файлами и организации интерактивного интерфейса.

Цель работы

Разработать консольное приложение — клавиатурный тренажер, который поможет пользователям улучшить скорость и точность набора текста. Программа должна включать:

- Выбор уровня сложности.

- Таймер для измерения скорости печати.

- Подсчет ошибок и вывод статистики после завершения упражнения.

- Возможность сохранения результатов в файл для отслеживания прогресса.

Используемые технологии

- Язык программирования C (стандарт C11/C17).

- Библиотеки для работы с консолью (`conio.h`, `termios.h` в Linux).

- Файловый ввод-вывод (`fopen()`, `fprintf()`, `fscanf()`).

- Функции времени (`time.h`) для реализации таймера.

Ожидаемый результат

Готовая программа, позволяющая пользователю тренировать скорость печати с выводом статистики и сохранением результатов. Проект продемонстрирует умение работать с базовыми конструкциями C, управлять пользовательским вводом и организовывать структуру данных.

Основные структуры данных

typedef enum { EASY, MEDIUM, HARD, ASDF, JKL, CUSTOM, MINECRAFT } Diff;

//структура названий всех сложностей.

typedef struct { //статистика для .json файла

    char date[20];

    double speed;

    double accuracy;

    double elapsed;

    int correct;

    int wrong;

    Diff difficulty;

} Stat;

typedef struct { //статистика игрока

    char nickname[MAX\_NICK\_LEN];

    Stat \*stats;

    size\_t count;

} PlayerStats;

typedef struct { //статистика лидера

    char nickname[MAX\_NICK\_LEN];

    double speed;

    char date[20];

} LeaderEntry;

Алгоритмы работы программы

**Работа с пользователем:**

Выбор режима:

* Тренировка
* Сложность
* Статистика
* Рекорды
* Выход

Выбор сложности:

* EASY
* MEDIUM
* HARD
* ASDF
* JKL:
* Your text

Обработка ввода:

* Сравнение строк
* Засекание времени
* Подсчет процентов
* Сохранение статистики.
* Создание .json файла, со структурой Stat.
* Чтение и сравнение данных из файла.

Технические аспекты

Используемые библиотеки:

#include <stdio.h> //стандартные библиотеки

#include <stdlib.h>

#include <string.h> //работа со строками

#include <time.h> //работа с временем

#include <jansson.h> //работа с .json

Обработка ошибок:

* Проверка ввода

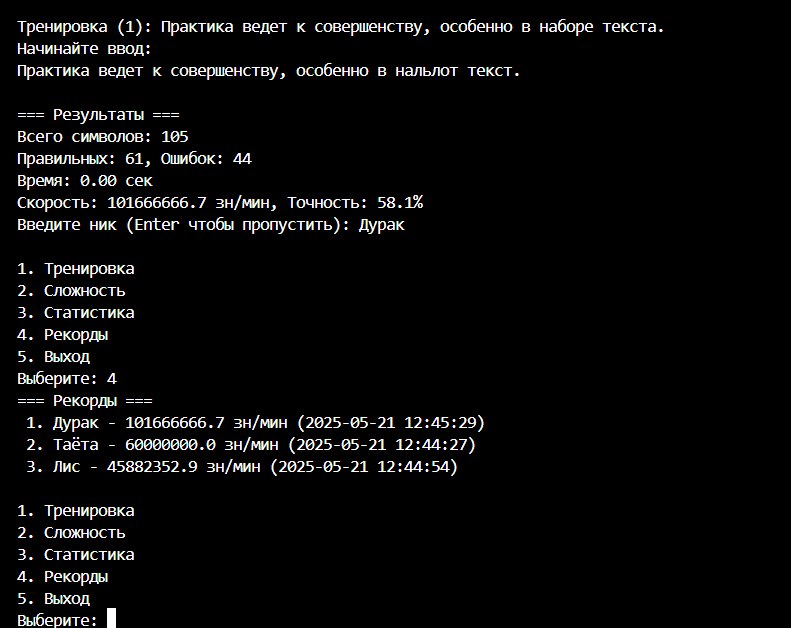
Работа программы

Выбор сложности



Рисунок 1 – Выбор сложности

Режим тренировки

Рисунок 2 – Режим тренировки

Вывод таблицы рекордов

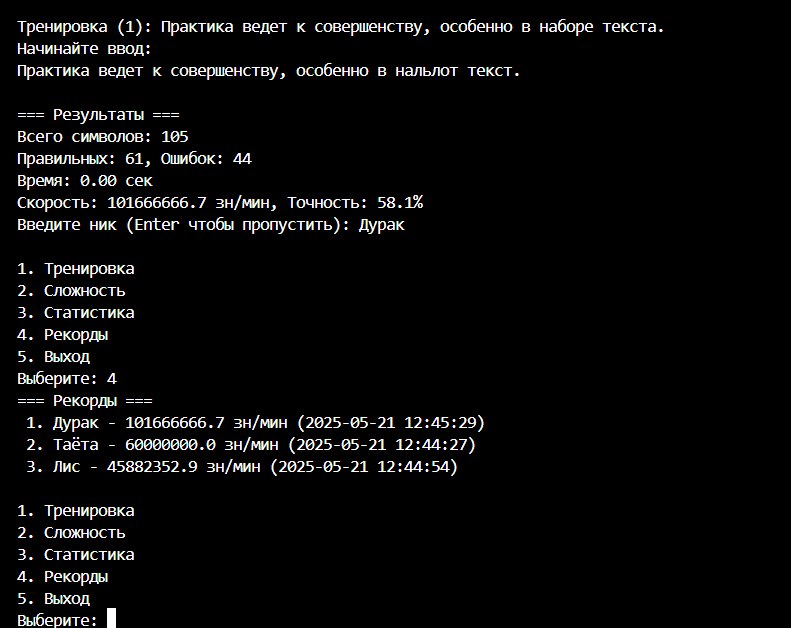


Рисунок 3 – Вывод таблицы рекордов

Вывод всех рекордов

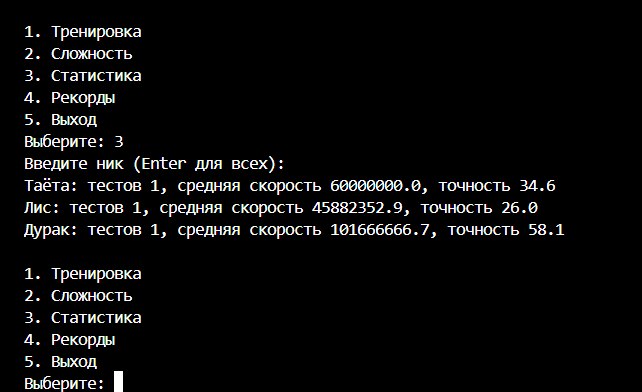
****

Рисунок 4 – Вывод всех рекордов

Выход из программы

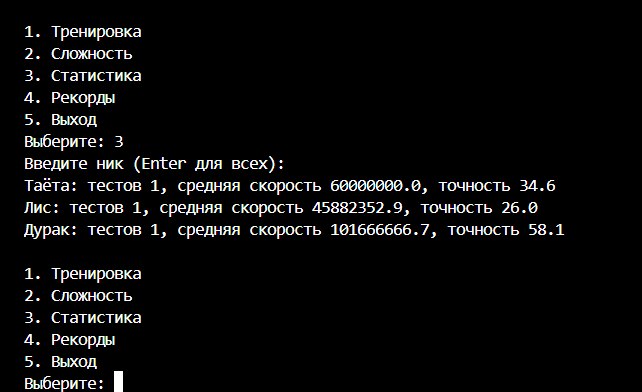
****

Рисунок 5 – Выход

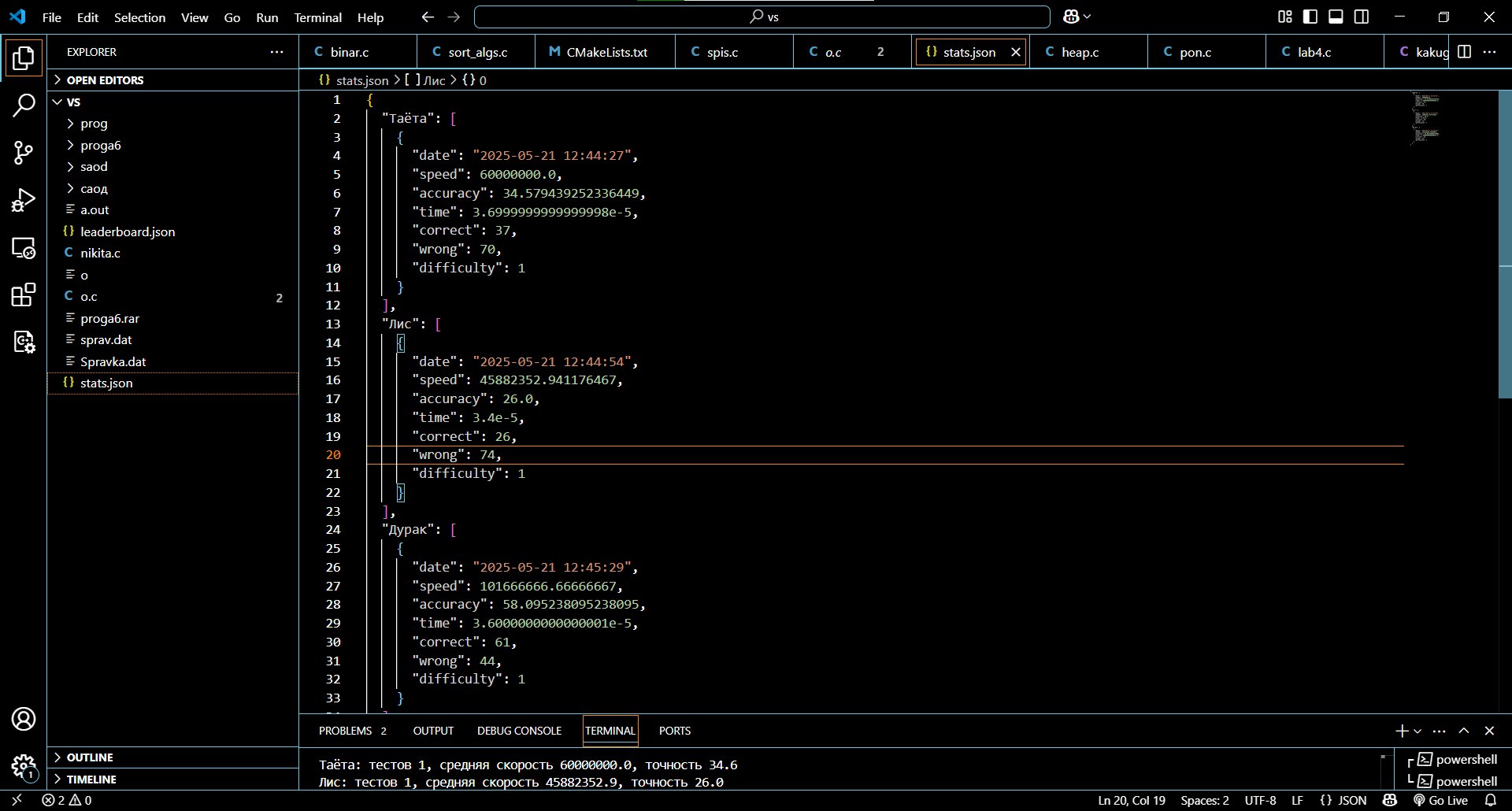
.json файл, запись тренировок

Рисунок 6 – json файл

Заключение

В ходе выполнения лабораторной работы был разработан клавиатурный тренажер на языке C, предназначенный для обучения слепой печати и улучшения скорости набора текста. Основные алгоритмы включают выбор режима, сложности, подсчет и сохранение результатов.

Ссылки на ресурсы

Github репозиторий с реализацией программы.

<https://github.com/TayaVoin/ProgrammingC/tree/main/RGR>