### 

КОМИТЕТ ПО ОБРАЗОВАНИЮ САНКТ-ПЕТЕРБУРГА

**САНКТ-ПЕТЕРБУРГСКОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**«КОЛЛЕДЖ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ»**

**ОТЧЕТ**

**Основы алгоритмизации и программирования**

**Практическая работа 7. Структуры. Работа с файлами.**

Работу выполнил: Танасов Евгений Эдуардович

Группа:324 Специальность: 09.02.07

Преподаватель: Смирнова Ирина Петровна

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«16» ноября 2024 г.

Преподаватель: Смирнова Ирина Петровна

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_» сентября 2021 г.

Санкт-Петербург 2021

Санкт-Петербург 2015 г.

Оглавление

[**Цель практической работы:** 3](#_Toc181912877)

[**Исходный текст программы:** 3](#_Toc181912878)

[Результат работы программы**:** 4](#_Toc181912879)

# **Цель практической работы:**

Разобраться в построении структуры и в коде объяснить как она работает.

# **Исходный текст программы:**

// Подключаем библиотеки

#include <stdio.h> // Для работы с вводом-выводом

#include <string.h> // Для работы со строками (в данном коде не используется явно)

#include <locale.h> // Для работы с языками

// Определяем структуру TBook для хранения информации о книгах

typedef

struct {

char author[40]; // Поле для автора книги (максимум 39 символов + 1 для '\0')

char title[80]; // Поле для названия книги (максимум 79 символов + 1 для '\0')

int count; // Поле для количества экземпляров книги

} TBook; // Имя структуры

int main() // Точка входа в программу

{

setlocale(LC\_ALL, "rus"); // Подключаем русский язык

const int N = 100; // Максимальное количество книг

TBook Books[N]; // Массив структур для хранения информации о книгах

int i, M; // i — счетчик для циклов, M — текущее количество книг

FILE\* F; // Указатель на файл

char s[80]; // Временная строка для ввода данных

// Открываем файл books.dat в режиме бинарного чтения

F = fopen("books.dat", "rb");

if (F == NULL) { // Проверяем, удалось ли открыть файл

printf("Файл books.dat не найден. Создайте новый список книг.\n");

M = 0; // Если файл отсутствует, начинаем с нуля книг

}

else {

// Считываем данные из файла в массив Books

M = fread(&Books[0], sizeof(TBook), N, F);

fclose(F); // Закрываем файл

}

// Выводим данные о книгах, которые были считаны из файла

printf("Имеющиеся данные о книгах:\n");

for (i = 0; i < M; i++) {

printf("%s, %s, %d\n",

Books[i].author, Books[i].title, Books[i].count);

}

// Проверяем, есть ли место для добавления новой книги

if (M < N) {

printf("\nВведите поля новой структуры:\n");

// Считываем имя автора

printf("author: ");

fgets(Books[M].author, sizeof(Books[M].author), stdin);

Books[M].author[strcspn(Books[M].author, "\n")] = '\0'; // Убираем '\n'

// Считываем название книги

printf(" title: ");

fgets(Books[M].title, sizeof(Books[M].title), stdin);

Books[M].title[strcspn(Books[M].title, "\n")] = '\0'; // Убираем '\n'

// Считываем количество экземпляров (сначала в строку, затем преобразуем)

printf(" count: ");

fgets(s, sizeof(s), stdin);

sscanf(s, "%d", &Books[M].count);

M++; // Увеличиваем количество книг

// Открываем файл для записи обновленного списка книг

F = fopen("books.dat", "wb");

if (F == NULL) { // Проверяем, удалось ли открыть файл

printf("Ошибка: не удалось открыть файл для записи.\n");

return 1; // Завершаем программу с ошибкой

}

// Записываем данные о книгах в файл

fwrite(&Books[0], sizeof(TBook), M, F);

fclose(F); // Закрываем файл

}

else {

printf("Нет места для добавления новых книг.\n");

}

// Ожидание ввода, чтобы пользователь успел увидеть вывод программы

printf("\nНажмите Enter для выхода...");

getchar();

return 0; // Возвращаем успешный код завершения программы

}

# Результат работы программы**:**

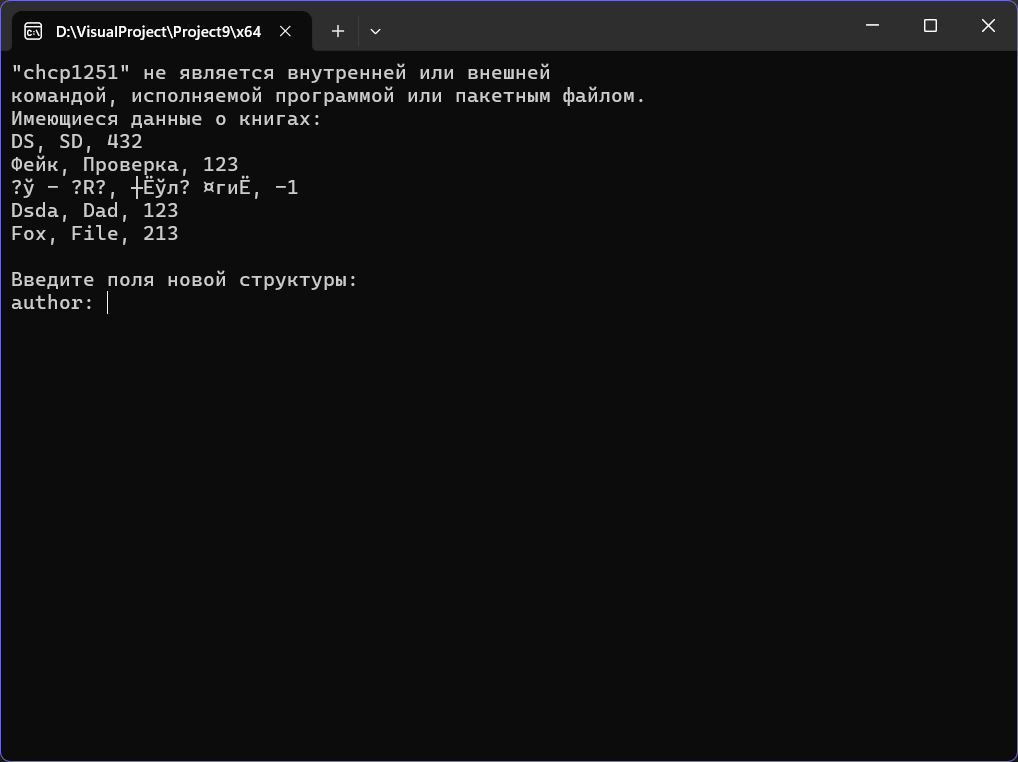


Рисунок 1 – Тест 1