Министерство образования и науки Российской Федерации

федеральное государственное бюджетное образовательное

учреждение высшего образования

«Алтайский государственный технический университет

им. И.И. Ползунова»

Факультет \_\_\_информационных технологий\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кафедра \_\_\_\_прикладной математики\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Специальность (направление, профиль) \_\_ПИ\_\_\_\_\_\_\_\_\_\_\_\_\_

Курсовой проект

защищен с оценкой\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_В.С. Троицкий\_

(подпись руководителя проекта) (инициалы, фамилия)

“\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_ 2020 г.

Курсовой проект

\_\_\_\_\_\_Разработка программного обеспечения для учета данных \_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_о лекарствах\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(тема курсового проекта)

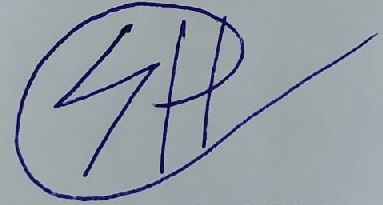
Пояснительная записка

по дисциплине \_\_Программирование-3\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_КП 09.03.04.06.000 ПЗ\_\_\_\_\_\_\_

(обозначение документа)

Студент группы ПИ92 Шульпов\_В.М.\_\_\_\_\_\_\_\_\_25.05.2020



(фамилия, имя, отчество) (подпись) (дата)

Руководитель

проекта доцент, к.т.н.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_В.С. Троицкий

(должность, ученое звание) (подпись) (инициалы, фамилия)

БАРНАУЛ 2020Министерство науки и высшего образования Российской Федерации

ФГБОУ ВО «Алтайский государственный технический университет

имени И.И. Ползунова»

Факультет информационных технологий

Кафедра «Прикладная математика»

З А Д А Н И Е

на курсовой проект по дисциплине «Программирование-3»

студенту группы ПИ-92 Шульпову Виктору Максимовичу

Тема курсового проекта: «Разработка игры с использованием технологии Ray Casting».

Календарный план работы:

|  |  |  |  |
| --- | --- | --- | --- |
| № этапа | Содержание этапа | Дата начала этапа | Количество дней на этап |
| 1 | Получение задания | 01.09.2020 | 1 |
| 2 | Постановка задачи. | 08.09.2020 | 7 |
| 3 | Проектирование программы | 15.09.2020 | 7 |
| 4 | Реализация программы | \*\*.\*\*.2020 | \* |
| 5 | Тестировние программы | \*\*.\*\*.2020 | \* |
| 6 | Оформление пояснительной записки | \*\*.\*\*.2020 | \* |
| 7 | Защита курсового проекта | \*\*.\*\*.2020 | \* |

Руководитель проекта \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Троицкий В.С., доцент

подпись

Дата выдачи задания «1» сентября 2020 г.

число месяц год

Задание принял к исполнению \_\_\_\_\_\_ Шульпов В.М.

подпись

**Содержание**

[Введение 4](#_Toc41345394)

[1 Обзор предметной области и постановка задачи 5](#_Toc41345395)

[1.1 Обзор предметной области 5](#_Toc41345396)

[1.2 Постановка задачи 5](#_Toc41345397)

[1.2.1 Техническое задание 5](#_Toc41345398)

[1.2.2 Требования к функциональности 6](#_Toc41345399)

[2 Проектирование 7](#_Toc41345400)

[2.1 Укрупненный алгоритм решения 7](#_Toc41345401)

[2.2 Структура данных 8](#_Toc41345402)

[2.3 Структура файлов 9](#_Toc41345403)

[3 Реализация 10](#_Toc41345404)

[3.1 Выбор средств реализации 10](#_Toc41345405)

[3.2 Структура программы 10](#_Toc41345406)

[3.3 Состав программы 11](#_Toc41345407)

[3.4 Описание реализации ПО 13](#_Toc41345408)

[Заключение 17](#_Toc41345409)

[Список использованных источников 18](#_Toc41345410)

[Приложения 19](#_Toc41345411)

[Приложение А Исходный код программы 19](#_Toc41345412)

[Приложение Б Результаты работы программы 49](#_Toc41345413)

# Введение

Для проектов самой разной направленности всегда была актуальна проблема хранения информации, ее структуризации и возможности эффективного использования. Сегодня эта задача может быть значительно упрощена за счет работы с электронными таблицами и базами данных.

Работа с пользовательскими данными - это распространенный вид деятельности в сфере информационных технологий, предполагающий большой объем работы и большую ответственность. Поэтому каждому разработчику прикладного программного обеспечения необходимо получить опыт в реализации подобных проектов.

**Цель данной работы:** изучение принципов разработки консольных приложений для организации работы с большими наборами структурированных данных.

Изучение технологии Ray Casting, разработка программного обеспечения с использованием методалогии ООП на языках С# и Python.

**Главная задача:** написать программу, работающую в консольном режиме и позволяющую организовать сбор, хранение и обработку структурированных данных определенной направленности.

Написать ПО, позволяющее пользователю играть, а именно: передвигаться по комнате, в которой есть стены и собирать монетки или стрелять в противников, появляющихся в случайной точке комнаты. (будущая разработка обоих режимов)

# 1 Обзор предметной области и постановка задачи

## 1.1 Обзор предметной области

Запись – строка, содержащая информацию об одной игрушке.

*Таблица* содержит множество строк – записей.

Функция *редактирование* служит для изменения отдельных записей таблицы.

*Удаление* служит для исключения записей из таблицы

*Запрос* служит для поиска определенных записей по условиям, которые задал пользователь.

*Добавление* служит для внесения новых записей в таблицу

*БД – база данных* – совокупность всех записей таблицы

Для работы с базами данных существуют специализированные языки – SQL («язык структурированных запросов»), Python с его библиотеками для работы с данными и другие. Но можно реализовать данную задачу и на языке высокого уровня, «Си», чему и посвящена эта работа.

## 1.2 Постановка задачи

### 1.2.1 Техническое задание

Организовать работу с базой данных, состоящих из строк. Каждая строка содержит:

* Название лекарства
* Название лекарства;
* Страна-производитель;
* Дата производства;
* Номера аптек;
* Цена лекарства.

Например, Perekis\_vodoroda Russia 2020.01.25 1 2 12 0 0 40

Реализовать обработку файла с данными по лекарствам (добавление, удаление, редактирование записей, выполнение запросов по поиску), используя среду разработки Visual Studio.

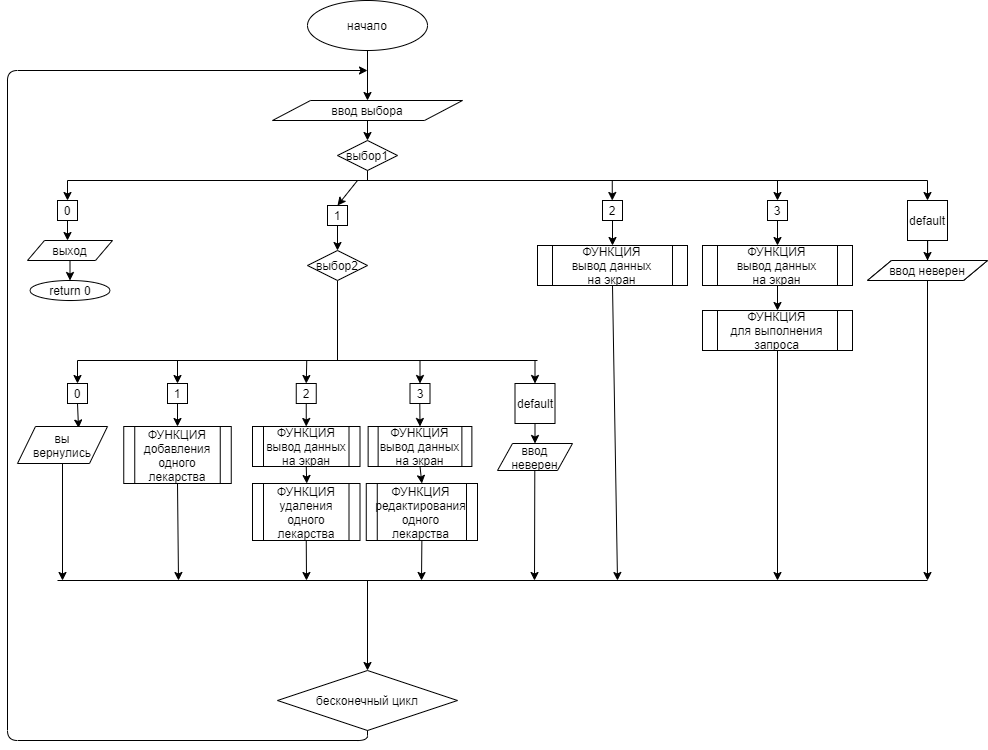
### 1.2.2 Требования к функциональности

Структура диалога программы:

1. добавить/изменить/удалить
   1. ввести новое лекарство
   2. удалить лекарство
   3. редактировать существующее лекарство (множественный выбор)
      1. название
      2. страна-производитель
      3. дата
      4. наличие в опеределнной аптеке
      5. цена
   4. вернуться назад
2. вывести всю таблицу
3. сделать запрос (множественный выбор)
   1. по названию
   2. стране-производителю
   3. по дате
   4. по наличию в опеределнной аптеке
   5. по цене
4. выйти

# 2 Проектирование

## 2.1 Укрупненный алгоритм решения



## 2.2 Структура данных

Запись содержит следующие поля:

* + Id лекарства (уникальный номер) – целое неотрицательное число (int);
  + Название лекарства – строка, состоящая из символов (массив char);
  + Страна-производитель – строка, состоящая из символов (массив char);
  + Дата производства – строка, состоящая строго из 8 цифр и 2 точек (в формате 2020.01.25) (char\*);
  + Номера аптек, в которых лекарство имеется в наличии (максимум 5) – наименование города – строка, состоящая из целых чисел до 99, разделенных пробелами (номер 0 обозначает, что аптека не указана) (массив int);
  + Цена лекарства – целое положительное число (int).

Для работы с этими данными использовалась структура. Она выбрана с целью объединения данных в одну группу (для лучшей читабельности кода).

/\*med - структура для добавления записи, редактирования записи

new\_med - для записи данных при редактировании\*/

struct {

int id; //ключ

char name[MAX\_LENGTH]; //название

char country[MAX\_LENGTH]; //страна-производитель

char date[MAX\_LENGTH]; //дата производства

int pharmacy\_number[MAX\_PHARMACIES]; //номера аптек

int price; //максимальная цена

} med, new\_med;

Имеется важная переменная, которая передается почти во все функции:

int count = 0; //текущее количество во лекарств

С помощью неё в процессе изменения таблицы (добавления/удаления записей) отслеживается количество записей на текущий момент, то есть эта переменная может измениться не раз за один запуск программы. Она хранится в первой строке файле, об этом говорится подробнее в следующем подзраделе.

## 2.3 Структура файлов

Для работы программы необходим исполняемый файл программы.

Записи базы данных хранятся построчно в файле medicines.txt. При его отсутсвии он создается автоматически. В первой строчке всегда хранится количество записей. Для редактирования и удаления используется вспомогательный файл rewrite.txt, который после удаления/редактирования переименовывается в medicines.txt (после удаления основого medicines.txt).

Структура хранения

1 строка – 5 символов для количества записей в таблице и символ ‘\n’

2 строка и все последующие записаны в одном формате:

5 символов – id, 25 символов – название, 25 символов – страна, 25 символов – дата, 25 (5\*5) символов – 5 номеров аптек, 10 символов (последний ‘\n’) – цена.

Пример:

0 Perekis\_vodoroda Russia 2020.01.25 1 2 12 0 0 40

# 3 Реализация

## 3.1 Выбор средств реализации

Данное ПО реализовано на языке С с использованием среды разработки программного Microsoft Visual Studio 2019. Язык С - компилируемый статически типизированный язык программирования общего назначения.

Для проверки введённых данных используются условные операторы.

Используются файлы TXT формата. Расширение TXT традиционно используется для простых текстовых файлов.

Корректировка списка (добавление, удаление и корректировка отдельных полей) организована так, чтобы корректировать можно было файл любого размера, то есть без переноса всей информации из файла в оперативную память. Для решения этой проблемы используется вспомогательный файл. Например, для удаления из файла «А» определенной записи следует читать из файла «А» по одной записи и проверять: если эту запись нужно оставить, то переписываем ее во вспомогательный файл «В», если запись не нужна (то есть ее надо удалить), то не переписываем ее в файл «В». В итоге в файле «В» получим то, что необходимо иметь в «А» после требуемого удаления. Затем либо удаляем файл «А» и затем переименовываем файл «В» в файл «А», либо переписываем все из файла «В» во вновь созданный файл «А» (то есть файлы «А» и «В» становятся одинаковыми), а затем удаляем уже не нужный файл «В». Аналогично поступаем и для корректировки отдельных полей определенной записи. Такая организация работы с файлом, когда все записи в файле читаются от начала файла последовательно друг за другом, называется «последовательный доступ к файлу». Это самый простой и безопасный способ работы с файлом, но длительный по времени.

## 3.2 Структура программы

Программа имеет один исходный файл программы – CourseWork.cpp

В программе реализованы следующие модули и взаимодействие между ними, изображенные на рисунке 1.

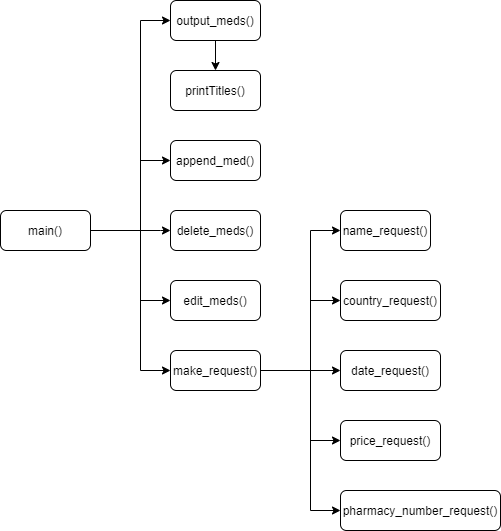


Рисунок 1 – Модульная структура

## 3.3 Состав программы

int main() – ОСНОВНАЯ ФУНКЦИЯ (в ней реализован «диалог программ»)

int append\_med(int \*\_count) - ФУНКЦИЯ ДОБАВЛЕНИЯ 1 ЛЕКАРСТВА

int edit\_meds(int \*\_count); - ФУНКЦИЯ РЕДАКТИРОВАНИЯ 1 ЛЕКАРСТВА

int delete\_meds(int\* \_count); - ФУНКЦИЯ УДАЛЕНИЯ 1 ЛЕКАРСТВА

int make\_request(int\* \_count); - ФУНКЦИЯ ВЫБОРА И ВЫПОЛНЕНИЯ ЗАПРОСОВ

int output\_meds(int\* \_count); - ФУНКЦИЯ ВЫВОДА ВСЕХ ЛЕКАРСТВ

void printTitles(); - ФУНКЦИЯ ПЕЧАТИ ЗАГОЛОВОЧНОЙ ЧАСТИ ТАБЛИЦЫ

bool name\_request(char med\_name[], char name[]); - ФУНКЦИЯ ЗАПРОСА ПО НАЗВАНИЮ ЛЕКАРСТВА

bool country\_request(char med\_country[], char country[]); - ФУНКЦИЯ ЗАПРОСА ПО СТРАНЕ ЛЕКАРСТВА

bool date\_request(char med\_date[], char date[]); - ФУНКЦИЯ ЗАПРОСА ПО ДАТЕ ЛЕКАРСТВА

bool price\_request(int med\_price, int price, char choice\_price); - ФУНКЦИЯ ЗАПРОСА ПО ЦЕНЕ ЛЕКАРСТВА

bool pharmacy\_number\_request(int med\_pharmacy\_number[], int pharmacy\_number); - ФУНКЦИЯ ЗАПРОСА ПО НАЛИЧИЮ ЛЕКАРСТВА В АПТЕКЕ

## 3.4 Описание реализации ПО

При запуске программы пользователю предоставляется высвечивается название программы «УПРАВЛЕНИЕ ДАННЫМИ ЛЕКАРСТВ АПТЕКИ» и выбор дальнейших действий:

1 - добавить/изменить/удалить

2 - вывести всю таблицу

3 - сделать запрос

0 – выйти

На этом шаге программа открывает файл medicines.txt и читает число, обозначающее количество записей в базе данных, в переменную count. Если же файл не существует – создает его и записывает 0 как количество записей в первую строку.

Если пользователь вводит «1», ему предоставляется возможность ввести новое лекарство, удалить лекарство, редактировать существующее или вернуться назад:

1 - добавить/изменить/удалить

1 - ввести новое лекарство

2 - удалить лекарство

3 - редактировать существующее лекарство

0 - вернуться назад

Рассмотрим каждый случай отдельно.

При вводе «1» вызывается функция *добавлени*я лекарства (append\_med(&count)). Параметр *count* будет записан как id новой записи. Пользователь должен будет ввести такие данные, как: название, страна, дата, количество аптек, затем их номера, цену.

При вводе «2» вызывается функция вывода всей таблицы для того, чтобы пользователь мог выбрать запись и понять, какое id ей принадлежит. Далее вызывается функция *удаления* лекарства (delete\_meds(&count)). Параметр *count* будет использован для выбора записи, которую нужно «уничтожить». Пользователю будет дана возможность отказаться от удаления перед тем, как программа запросит id записи.

При вводе «3» вызывается функция вывода всей таблицы для того, чтобы пользователь мог выбрать запись и понять, какое id ей принадлежит. Далее вызывается функция *редактирования* лекарства (edit\_meds(&count)). Параметр *count* будет использован для выбора записи, которую нужно «изменить». Пользователю будет дана возможность отказаться от редаткирования перед тем, как программа запросит id записи. После этого программа спросит, какие поля нужно отредактировать:

Какое поле вы хотите изменить?

1)название

2) страну-производителя

3) дату

4)номера аптек

5)цену

(ШАБЛОН: 1 0 0 1 0 - для редактирования названия {1 столбец} и номеров аптек {4 столбец})

Выше представлен выбор, где «1» – это истина, «0» – ложь, то есть пользователь должен ввести пять цифр (0 или 1), где «1» будет обозначать то, что поле будет предложено для редактирования.

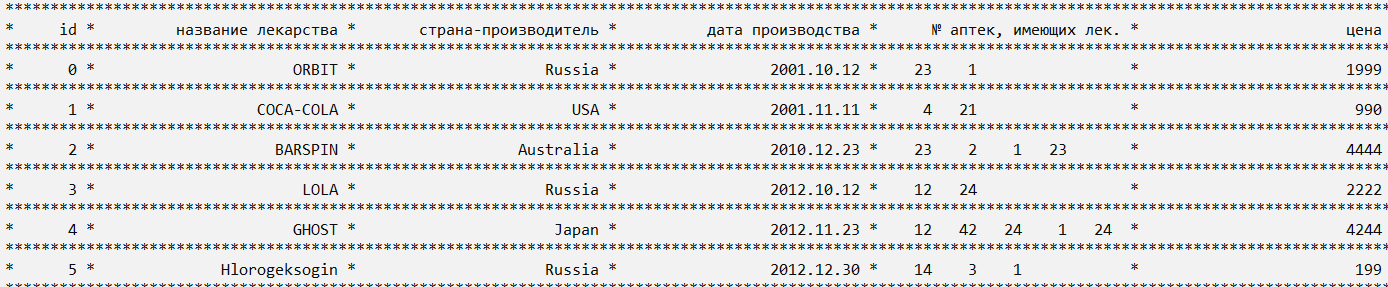
При вводе «0» программа возвращается на шаг, то есть возвращается в начальное положение:

1 - добавить/изменить/удалить

2 - вывести всю таблицу

3 - сделать запрос

0 – выйти

При вводе «2» будет вызвана функция *вывода всех данных* из файла (output\_meds(&count)), где параметр *count* – это количество записей:

При вводе «3» будет вызвана функция выбора и выполнения запросов, которая будет вызывать такие функции, как:

bool name\_request(char med\_name[], char name[]); - ФУНКЦИЯ ЗАПРОСА ПО НАЗВАНИЮ ЛЕКАРСТВА

bool country\_request(char med\_country[], char country[]); - ФУНКЦИЯ ЗАПРОСА ПО СТРАНЕ ЛЕКАРСТВА

bool date\_request(char med\_date[], char date[]); - ФУНКЦИЯ ЗАПРОСА ПО ДАТЕ ЛЕКАРСТВА

bool price\_request(int med\_price, int price, char choice\_price); - ФУНКЦИЯ ЗАПРОСА ПО ЦЕНЕ ЛЕКАРСТВА

bool pharmacy\_number\_request(int med\_pharmacy\_number[], int pharmacy\_number); - ФУНКЦИЯ ЗАПРОСА ПО НАЛИЧИЮ ЛЕКАРСТВА В АПТЕКЕ

Будет предложен выбор полей для выполнения запроса (аналогично редактированию записи):

По каким столбцам вы хотите сделать запрос?

1) по названию

2) стране-производителю

3) по дате

4) по наличию в опеределнной аптеке

5) по цене

(ШАБЛОН: 1 0 0 1 0 - запрос по названию {1 стобец} и цене {4 столбец})

Затем по выбранным полям нужно будет ввести данные:

Для названия нужно будет ввести строку (выберутся лекарства с названием, включающим в себя последовательность символов, введеную пользователем)

Для страны-производителя – тоже строку (лекарства будут выбираться по точному совпадению)

Для даты – ввести строку по формату «ГГГГ.ММ.ДД» (будут выбраны лекарства, даты которых младше введенной)

Для наличия в определенной аптеке – номер аптеки (число). Будет производиться выбор хотя бы одному совпадению в записи.

Для цены – число, а также выбор «не дешевле», «не дороже» или «равно»

1 - добавить/изменить/удалить

2 - вывести всю таблицу

3 - сделать запрос

0 – выйти

И наконец при вводе «0» программа завершит своё выполнение.

В функциях добавления, редактирования и удаления будет перезаписыватсья файл, то есть информация в нём будет изменяться сразу же после выполнения функции, а не после завершения программы.

# Заключение

Был создан программный продукт, позволяющий эффективно работать с базой данных Аптеки после изучения краткой по объему документации. Реализованы защиты на ввод некорректных данных, эффективная обработка запроса в реальном режиме.

В процессе выполнения задания были подробно изучены методы работы с файлами на языке С.

Преимущества приложения:

* Интуитивно понятный интерфейс;
* Быстрая и эффективная обработка запроса;
* Возможность оперативного редактирования/ добавления/ удаления записей;
* Предоставление красивого вывода данных в консоли

Недочёты приложения:

* Работа только с одной таблицей;
* Отсутствие возможности выбора файла для обработки;
* Отсутсвие выбора пользователя
* В некоторых местах нарушил своё же правило записи названий переменных/функций (было бы лучше, если использовался для функций CamelCase (с [англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) — «ВерблюжийРегистр»), а для перменных слова разделялись бы только нижним подчеркиванием.

Возможны дальнейшие усовершенствования программного продукта с течением времени:

* Работа с несколькими таблицами одновременно;
* Сортировка данных по определенным полям.
* Запись в файл лекарств для каждой аптеки при необходимости.
* Не консольный интерфейс

# Список использованных источников

Программирование C и C++ [Электронный ресурс]: сайт. – Челябинск, 2020. – URL: <https://prog-cpp.ru/c-files/> (дата обращения: 20.05.2020).

Язык Си [Электронный ресурс]: сайт. – Санкт-Петербург, 2018. – URL: <https://learnc.info/c/text_files.html> (дата обращения: 20.05.2020).

Егорова Е.В. Программирование на языке Си. Учебное пособие / Алт. госуд. технич. Ун-т им. И.И. Ползунова. – Барнаул: 2012. – 184 с.

# Приложения

## Приложение А Исходный код программы

/\*Программа предназначена для работы с данными по лекарствам.

Пользователь может добавлять и удалять данные,

также может редактировать и делать запросы по любым полям.

Разработана запись в файл, благодаря которой файлы будут хранится на диске.\*/

//в программе можно вводить не более 5 аптек (для удобства вывода таблицы данных)

#define \_CRT\_SECURE\_NO\_WARNINGS

#include "stdio.h"

#include <conio.h>

#include <stdlib.h>

#include <string.h>

#include <locale.h>

#include <Windows.h>

#define clear(stream) rewind((stream))

#define MAX\_MEDICAMENTS 100 //максимальное количество лекарств

#define MAX\_LENGTH 50 //максимальная длина текста

#define MAX\_PHARMACIES 5 //максимальное кол-во аптек, имеющих в наличии лекраство

#define NUM\_KEYS\_TABLE 6 //кол-во ключей таблицы (заголовков столбцов)

#define NUM\_SORT\_FIELDS 5 //кол-во сортируемых столбцов

#define NUM\_EDIT\_FIELDS 5 //кол-во редактируемых столбцов

#define LENGTH\_DATE 10 //длина даты (2000.09.14)

////////////////////////////////////////////////////////////////////////////////////////////////////

//ГЛОБАЛЬНЫЕ ПЕРЕМЕННЫЕ

char stars[] = "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

char titles[NUM\_KEYS\_TABLE][MAX\_LENGTH] = {

"id",

"название лекарства",

"страна-производитель",

"дата производства",

"№ аптек, имеющих лек.",

"цена" };

char fileName[] = "medicines.txt"; //основной файл

char fileName2[] = "rewrite.txt"; //для переписывания при редактировании, удалении

/\*med - структура для добавления записи, редактирования записи

new\_med - для записи данных при редактировании\*/

struct {

int id; //ключ

char name[MAX\_LENGTH]; //название

char country[MAX\_LENGTH]; //страна-производитель

char date[MAX\_LENGTH]; //дата производства

int pharmacy\_number[MAX\_PHARMACIES]; //номера аптек

int price; //максимальная цена

}med, new\_med;

FILE \*file;//основной файл

FILE \*file2;//вспомогательный файл

////////////////////////////////////////////////////////////////////////////////////////////////////

int append\_med(int \*\_count); // ФУНКЦИЯ ДОБАВЛЕНИЯ 1 ЛЕКАРСТВА

int edit\_meds(int \*\_count); // ФУНКЦИЯ РЕДАКТИРОВАНИЯ 1 ЛЕКАРСТВА

int delete\_meds(int\* \_count); // ФУНКЦИЯ УДАЛЕНИЯ 1 ЛЕКАРСТВА

int make\_request(int\* \_count); // ФУНКЦИЯ ВЫБОРА И ВЫПОЛНЕНИЯ ЗАПРОСОВ

int output\_meds(int\* \_count); // ФУНКЦИЯ ВЫВОДА ВСЕХ ЛЕКАРСТВ

void printTitles(); // ФУНКЦИЯ ПЕЧАТИ ЗАГОЛОВОЧНОЙ ЧАСТИ ТАБЛИЦЫ

bool name\_request(char med\_name[], char name[]); // ФУНКЦИЯ ЗАПРОСА ПО НАЗВАНИЮ ЛЕКАРСТВА

bool country\_request(char med\_country[], char country[]); // ФУНКЦИЯ ЗАПРОСА ПО СТРАНЕ ЛЕКАРСТВА

bool date\_request(char med\_date[], char date[]); // ФУНКЦИЯ ЗАПРОСА ПО ДАТЕ ЛЕКАРСТВА

bool price\_request(int med\_price, int price, char choice\_price); // ФУНКЦИЯ ЗАПРОСА ПО ЦЕНЕ ЛЕКАРСТВА

bool pharmacy\_number\_request(int med\_pharmacy\_number[], int pharmacy\_number); // ФУНКЦИЯ ЗАПРОСА ПО НАЛИЧИЮ ЛЕКАРСТВА В АПТЕКЕ

////////////////////////////////////////////////////////////////////////////////////////////////////

int main(){

system("color F0");

setlocale(LC\_ALL, "Russian");

int choice; //вариант выбора в меню(0, 1, 2...)

int сhoice\_correction; //подвыбор в "вводе и корректировке"

int f\_exit = 0; //флаг выхода

int count = 0;//текущее кол-во лекарств

int error; //номер ошибки

file = fopen(fileName, "a");

fclose(file);

file = fopen(fileName, "r+");

if (file == NULL) return -1; /\* Файл не открылся \*/

rewind(file);

fscanf(file, "%5d", &count);

if (count == NULL) fprintf(file, "%5d\n", count); //записываем 0 (кол-во записей) в начало

else {

rewind(file);

fscanf(file, "%5d", &count);

}

fclose(file);

printf("УПРАВЛЕНИЕ ДАННЫМИ ЛЕКАРСТВ АПТЕКИ\n");

do {

printf("\n1 - добавить/изменить/удалить\n2 - вывести всю таблицу\n3 - сделать запрос\n0 - выйти\nВаш выбор:\t");

clear(stdin);

scanf\_s("%d", &choice);

switch (choice)

{

case 0:

puts("\nВЫ ВЫШЛИ\n");

return 0;

case 1:

printf("\t1 - ввести новое лекарство\n\t2 - удалить лекарство\n\t3 - редактировать существующее лекарство\n\t0 - вернуться назад\nВаш выбор:\t");

clear(stdin);

scanf\_s("%d", &сhoice\_correction);

switch (сhoice\_correction)

{

case 0:

puts("\nВЫ ВЕРНУЛИСЬ");

break;

case 1:

if (append\_med(&count) == -1) printf("\nФайл не открылся\n");

break;

case 2:

error = output\_meds(&count);

if (error == 1) printf("\nВыводить нечего, файл пуст\n");

if (error == -1) printf("\nФайл не открылся\n");

error = delete\_meds(&count);

if (error == -1) printf("\nФайл не открылся\n");

if (error == 1) printf("\nУдалять нечего, файл пуст\n");

if (error == 2) printf("\nВы отменили удаление лекарства\n");

break;

case 3:

error = output\_meds(&count);

if (error == 1) printf("\nВыводить нечего, файл пуст\n");

if(error == -1) printf("\nФайл не открылся\n");

error = edit\_meds(&count);

if (error == -1) printf("\nФайл не открылся\n");

if (error == 1) printf("\nРедактировать нечего, файл пуст\n");

if (error == 2) printf("\nВы отменили редактирование лекарства\n");

break;

default:

puts("\nВВОД НЕВЕРЕН\n");

break;

}

break;

case 2:

error = output\_meds(&count);

if (error == 1) printf("\nВыводить нечего, файл пуст\n");

if(error==-1) printf("\nФайл не открылся\n");

break;

case 3:

error = output\_meds(&count);

if (error == 1) printf("\nВыводить нечего, файл пуст\n");

if (error == -1) printf("\nФайл не открылся\n");

error = make\_request(&count);

if (error == -1) printf("\nФайл не открылся\n");

if (error == 1) printf("\nВыборка невозможна, файл пуст\n");

break;

default:

puts("\nВВОД НЕВЕРЕН\n");

break;

}

} while (1);

}

int append\_med(int \*\_count){ // ФУНКЦИЯ ДОБАВЛЕНИЯ ЛЕКАРСТВА

file = fopen(fileName, "r+");

if (file == NULL) return -1; /\* Файл не открылся \*/

fseek(file, 0L, SEEK\_END);

bool date\_flag = true; //флаг проверки даты

int pharm = 0; //кол-во аптек

fprintf(file, "%5d", \*\_count);

puts("\n\t\tВВОД НОВОГО ЛЕКАРСТВА\n");

printf("\t\tВведите название (ШАБЛОН: Perekis\_vodoroda)\n\t\t\tНазвание:\t");

clear(stdin);

scanf\_s("%s", &med.name, MAX\_LENGTH);

fprintf(file, "%25.25s", med.name);

clear(stdin);

printf("\t\tВведите страну-производителя (ШАБЛОН: Arabskie\_Emirati)\n\t\t\tСтрана:\t");

clear(stdin);

scanf\_s("%s", &med.country, MAX\_LENGTH);

fprintf(file, "%25.25s", med.country);

printf("\t\tВведите дату производства лекарства (ШАБЛОН [год.месяц.число] с нулями: 2012.09.02)\n\t\t\tДата:\t");

do {

date\_flag = true;

clear(stdin);

scanf\_s("%s", &med.date, MAX\_LENGTH);

for (int k = 0; k < LENGTH\_DATE; k++)

{

if (

med.date[k] != '.' && (med.date[k] < '0' || med.date[k] > '9') //не точка и не цифра

|| strlen(med.date) != LENGTH\_DATE //длина не соотвествует

|| med.date[4] != '.' //5 символ не точка

|| med.date[7] != '.' //7 символ не точка

|| med.date[5] > '1' //6 символ больше 1

|| med.date[8] > '3' //9 символ больше 1

)

{

printf("\t\tВВОД НЕВЕРЕН. ВВОДИТЕ, ПОЖАЛУЙСТА, ПО ШАБЛОНУ\n\t\t\tДата:\t");

date\_flag = false;

break;

}

}

} while (date\_flag == false);

fprintf(file, "%25.25s", med.date);

printf("\t\tСколько аптек имеют в наличии это лекраство? (максимум %d)\t", MAX\_PHARMACIES);

do {

clear(stdin);

scanf\_s("%d", &pharm);

if (pharm > MAX\_PHARMACIES || pharm < 1) printf("\t\tНекорректный ввод, введите заново:\t");

else break;

} while (1);

printf("\t\t(ШАБЛОН: 12)\n");

for (int i = 0; i < MAX\_PHARMACIES; i++) {

if (i < pharm) {

do {

printf("\t\t\tНомер аптеки (%d) (до №99):\t", i + 1);

clear(stdin);

scanf\_s("%d", &med.pharmacy\_number[i]);

} while (med.pharmacy\_number[i] <= 0 || med.pharmacy\_number[i]>99);

fprintf(file, "%5d", med.pharmacy\_number[i]);

}

else fprintf(file, "%5d", 0);

}

do {

printf("\t\tВведите цену лекарства (ШАБЛОН: 200)\n\t\t\tЦена:\t");

clear(stdin);

scanf\_s("%d", &med.price);

} while (med.price <= 0);

fprintf(file, "%10d\n", med.price);

(\*\_count)++;

rewind(file);

fprintf(file, "%5d", \*\_count);

fseek(file, 0L, SEEK\_END);

fclose(file);

return 0;

};

int edit\_meds(int\* \_count) {// ФУНКЦИЯ РЕДАКТИРОВАНИЯ 1 ЛЕКАРСТВА

file = fopen(fileName, "r+");

if (file == NULL) return -1; // Файл не открылся

if (\*\_count == 0) {

return 1;

}

puts("\n\t\tРЕДАКТИРОВАНИЕ СУЩЕСТВУЮЩЕГО ЛЕКАРСТВА");

int id; // вводимый id

int choice\_edit[NUM\_EDIT\_FIELDS]; //массив с выбором редактируемых полей (1 или 0)

int flag\_input = 0; // флаг корректного ввода выбора вариантов редактирования

bool date\_flag = true; //флаг для проверки ввода даты

int pharm; // введенное кол-во аптек

printf(" Вы точно хотите отредактировать лекарство?\n Нажмите ENTER, если да, иначе - другую клавишу\n");

if (\_getch() != 13) return 2;

printf("Введите id лекарства: ");

do {

clear(stdin);

scanf\_s("%d", &id);

if (id >= \*\_count || id < 0) printf("Такого id в таблице нет, введите другое: ");

else break;

} while (1);

printf("\t\tКакое поле вы хотите изменить?\n\t\t1)название\n\t\t2) страну-производителя\n\t\t3) дату\n\t\t4)номера аптек\n\t\t5)цену\n\t\t(ШАБЛОН: 1 0 0 1 0 - для редактирования названия {1 столбец} и номеров аптек {4 столбец})\n\t\t\tВаш выбор редактируемых полей:\t");

do {

clear(stdin);

for (int i = 0; i < NUM\_SORT\_FIELDS; i++)

scanf\_s("%d", &choice\_edit[i]);

for (int i = 0; i < NUM\_SORT\_FIELDS; i++)

{

flag\_input = 1;

if (choice\_edit[i] != 0 && choice\_edit[i] != 1) {

printf("\n\tВВОД НЕВЕРЕН\n\tВВОДИТЕ, ПОЖАЛУЙСТА, ПО ШАБЛОНУ\n\tВаш выбор редактируемых полей:\t");

flag\_input = 0;

break;

}

}

} while (flag\_input == 0);

/\*записыываем новые выбранные данные в переменные\*/

for (int i = 0; i < NUM\_EDIT\_FIELDS; i++) {

if (choice\_edit[i] == 1) {

switch (i) {

case 0:

printf("\t\tВведите название (ШАБЛОН: Perekis\_vodoroda)\n\t\t\tНазвание:\t");

clear(stdin);

scanf\_s("%s", &new\_med.name, MAX\_LENGTH);

//fscanf(file, "%25.25s", &med.name);

//printf("\n%s\_\_\n", med.name);

//fprintf(file, "%25.25s", med.name);

break;

case 1:

printf("\t\tВведите страну-производителя (ШАБЛОН: Arabskie\_Emirati)\n\t\t\tСтрана:\t");

clear(stdin);

scanf\_s("%s", &new\_med.country, MAX\_LENGTH);

//fprintf(file, "%25.25s", med.country);

break;

case 2:

printf("\t\tВведите дату производства лекарства (ШАБЛОН [год.месяц.число] с нулями: 2012.09.02)\n\t\t\tДата:\t");

do {

date\_flag = true;

clear(stdin);

scanf\_s("%s", &new\_med.date, MAX\_LENGTH);

for (int k = 0; k < LENGTH\_DATE; k++)

{

if (

new\_med.date[k] != '.' && (new\_med.date[k] < '0' || new\_med.date[k] > '9') //не точка и не цифра

|| strlen(new\_med.date) != LENGTH\_DATE //длина не соотвествует

|| new\_med.date[4] != '.' //5 символ не точка

|| new\_med.date[7] != '.' //7 символ не точка

|| new\_med.date[5] > '1' //6 символ больше 1

|| new\_med.date[8] > '3' //9 символ больше 1

)

{

printf("\t\tВВОД НЕВЕРЕН. ВВОДИТЕ, ПОЖАЛУЙСТА, ПО ШАБЛОНУ\n\t\t\tДата:\t");

date\_flag = false;

break;

}

}

} while (date\_flag == false);

//fprintf(file, "%25.25s", med.date);

break;

case 3:

printf("\t\tСколько аптек имеют в наличии это лекраство?\t");

do {

clear(stdin);

scanf\_s("%d", &pharm);

if (pharm > MAX\_PHARMACIES || pharm < 1) printf("\t\tВведите меньше %d аптек:\t", MAX\_PHARMACIES + 1);

} while (pharm > MAX\_PHARMACIES || pharm < 1);

printf("\t\t(ШАБЛОН: 12)\n");

for (int i = 0; i < MAX\_PHARMACIES; i++) {

if (i < pharm) {

do {

printf("\t\t\tНомер аптеки (%d):\t", i + 1);

clear(stdin);

scanf\_s("%d", &new\_med.pharmacy\_number[i]);

} while (new\_med.pharmacy\_number[i] <= 0);

//fprintf(file, "%3d ", med.pharmacy\_number[i]);

}

else {

new\_med.pharmacy\_number[i] = 0;

//fprintf(file, "%3d ", 0);

}

}

//med.pharmacy\_number[pharm] = 0;

break;

case 4:

do {

printf("\t\tВведите цену лекарства (ШАБЛОН: 200)\n\t\t\tЦена:\t");

clear(stdin);

scanf\_s("%d", &new\_med.price);

} while (new\_med.price <= 0);

//fprintf(file, "%10d\n", med.price);

break;

}

}

}

file2 = fopen(fileName2, "w");

if (file2 == NULL) return -1; // Файл не открылся

fseek(file, 0L, SEEK\_SET);

fseek(file2, 0L, SEEK\_SET);

fscanf(file, "%5d ", \_count);

fprintf(file2, "%5d\n", \*\_count);

for (int i = 0; i < \*\_count; i++) {

fscanf(file, "%5d", &med.id);

fscanf(file, "%25s", &med.name);

fscanf(file, "%25s", &med.country);

fscanf(file, "%25s", &med.date);

for (int j = 0; j < MAX\_PHARMACIES; j++)

fscanf(file, "%5d", &med.pharmacy\_number[j]);

fscanf(file, "%10d\n", &med.price);

if(med.id==id){

fprintf(file2, "%5d", med.id);

if (choice\_edit[0] == 1) fprintf(file2, "%25.25s", new\_med.name);

else fprintf(file2, "%25.25s", med.name);

if (choice\_edit[1] == 1) fprintf(file2, "%25.25s", new\_med.country);

else fprintf(file2, "%25.25s", med.country);

if (choice\_edit[2] == 1) fprintf(file2, "%25.25s", new\_med.date);

else fprintf(file2, "%25.25s", med.date);

for (int i = 0; i < MAX\_PHARMACIES; i++)

if (choice\_edit[3] == 1) fprintf(file2, "%5d", new\_med.pharmacy\_number[i]);

else fprintf(file2, "%5d", med.pharmacy\_number[i]);

if (choice\_edit[4] == 1) fprintf(file2, "%10d\n", new\_med.price);

else fprintf(file2, "%10d\n", med.price);

}

else{

fprintf(file2, "%5d", med.id);

fprintf(file2, "%25.25s", med.name);

fprintf(file2, "%25.25s", med.country);

fprintf(file2, "%25.25s", med.date);

for (int i = 0; i < MAX\_PHARMACIES; i++)

fprintf(file2, "%5d", med.pharmacy\_number[i]);

fprintf(file2, "%10d\n", med.price);

}

}

fclose(file);

fclose(file2);

if( -1 == remove(fileName)) printf("ОШИБКА УДАЛЕНИЯ ФАЙЛА");

if (-1 == rename(fileName2, fileName)) printf("ОШИБКА ПЕРЕИМЕНОВАНИЯ ФАЙЛА");

return 0;

}

int delete\_meds(int\* \_count) {// ФУНКЦИЯ УДАЛЕНИЯ 1 ЛЕКАРСТВА

file = fopen(fileName, "r+");

if (file == NULL) return -1; // Файл не открылся

if (\*\_count == 0) {

return 1;

}

int id\_increment=0; //id инкремент

puts("\n\t\tУДАЛЕНИЕ ЛЕКАРСТВА\n");

int id;

printf(" Вы точно хотите удалить лекарство?\n Нажмите ENTER, если да, иначе - другую клавишу\n");

if (\_getch() != 13) return 2;

printf("Введите id лекарства: ");

do {

clear(stdin);

scanf\_s("%d", &id);

if (id >= \*\_count || id < 0) printf("Такого id в таблице нет, введите другое: ");

else break;

} while (1);

file2 = fopen(fileName2, "w");

if (file2 == NULL) return -1; // Файл не открылся

fseek(file, 0L, SEEK\_SET);

fseek(file2, 0L, SEEK\_SET);

fscanf(file, "%5d ", \_count);

fprintf(file2, "%5d\n", (\*\_count)-1);

for (int i = 0; i < \*\_count; i++) {

fscanf(file, "%5d", &med.id);

fscanf(file, "%25s", &med.name);

fscanf(file, "%25s", &med.country);

fscanf(file, "%25s", &med.date);

for (int j = 0; j < MAX\_PHARMACIES; j++)

fscanf(file, "%5d", &med.pharmacy\_number[j]);

fscanf(file, "%10d\n", &med.price);

if (med.id != id) {

fprintf(file2, "%5d", id\_increment++);

fprintf(file2, "%25.25s", med.name);

fprintf(file2, "%25.25s", med.country);

fprintf(file2, "%25.25s", med.date);

for (int i = 0; i < MAX\_PHARMACIES; i++)

fprintf(file2, "%5d", med.pharmacy\_number[i]);

fprintf(file2, "%10d\n", med.price);

}

}

(\*\_count)--;

fclose(file);

fclose(file2);

remove(fileName);

rename(fileName2, fileName);

return 0;

};

void printTitles() { // ФУНКЦИЯ ПЕЧАТИ ЗАГОЛОВОЧНОЙ ЧАСТИ ТАБЛИЦЫ

printf("\n %s\n", stars);

for (int i = 0; i != NUM\_KEYS\_TABLE; i++)

{

if (i == 0) {

printf(" \* ");

printf(" %5.5s \* ", titles[i]);

}

else printf(" %25.25s \* ", titles[i]);

}

printf("\n %s\n", stars);

}

int output\_meds(int\* \_count){ // ФУНКЦИЯ ВЫВОДА ВСЕХ ЛЕКАРСТВ

if (\*\_count == 0) return 1;

else {

file = fopen(fileName, "r+");

if (file == NULL) return -1; /\* Файл не открылся \*/

fseek(file, 0L, SEEK\_SET);

fscanf(file, "%5d", \_count);

printf("Данные по лекарствам (%d): \n", \*\_count);

printTitles();

for (int i = 0; i < \*\_count; i++) {

printf(" \* ");

fscanf(file, "%5d", &med.id);

printf(" %5d \* ", med.id);

fscanf(file, "%25s", &med.name);

printf(" %25.25s \* ", med.name);

fscanf(file, "%25s", &med.country);

printf(" %25.25s \* ", med.country);

fscanf(file, "%25s", &med.date);

printf(" %25.25s \* ", med.date);

for (int j = 0; j < MAX\_PHARMACIES; j++) {

fscanf(file, "%5d", &med.pharmacy\_number[j]);

if (med.pharmacy\_number[j] != 0)

printf("%5d", med.pharmacy\_number[j]); //рассчитано для 5 аптек

else

printf(" ");

}

printf(" \* ");

fscanf(file, "%10d\n", &med.price);

printf(" %25d \* ", med.price);

printf("\n %s\n", stars);

}

fclose(file);

}

return 0;

}

int make\_request(int\* \_count) {// ФУНКЦИЯ ВЫБОРА И ВЫПОЛНЕНИЯ ЗАПРОСА

if (\*\_count == 0) return 1;

file = fopen(fileName, "r");

if (file == NULL) return -1; /\* Файл не открылся \*/

fseek(file, 0L, SEEK\_SET);

fscanf(file, "%5d ", \_count);

int flag\_title = 0; //флаг (была ли выведена заголовочная часть таблицы)

int choice\_request[NUM\_SORT\_FIELDS]; //массив со значениями 1 или 0 для выбора запроса

printf("\tПо каким столбцам вы хотите сделать запрос?\n\t1) по названию\n\t2) стране-производителю\n\t3) по дате\n\t4) по наличию в опеределнной аптеке\n\t5) по цене\n\t(ШАБЛОН: 1 0 0 1 0 - запрос по названию {1 стобец} и цене {4 столбец})\n\tВаш запрос:\t");

int flag\_input = 0;

do {

clear(stdin);

for (int i = 0; i < NUM\_SORT\_FIELDS; i++)

scanf\_s("%d", &choice\_request[i]);

for (int i = 0; i < NUM\_SORT\_FIELDS; i++)

{

flag\_input = 1;

if (choice\_request[i] != 0 && choice\_request[i] != 1) {

printf("\n\tВВОД НЕВЕРЕН\n\tВВОДИТЕ, ПОЖАЛУЙСТА, ПО ШАБЛОНУ\n\tВаш запрос:\t");

flag\_input = 0;

break;

}

}

} while (flag\_input == 0);

bool date\_flag = true; //флаг для ввода валидной даты

bool price\_flag = true; //флаг для ввода валидной цены

bool request\_flag = true; //флаг запроса (изменяется в ходе проверки по каждому столбцу)

bool is\_request\_flag = false; //флаг, показывающий, что пользователь выбрал хотя бы один запрос (случай 0 0 0 0 0 - false)

bool title\_flag = false; //флаг = true, если заголовки таблицы выведены хотя бы один раз

for (int i = 0; i < \*\_count; i++) {

request\_flag = true;

fscanf(file, "%5d", &med.id);

fscanf(file, "%25s", &med.name);

fscanf(file, "%25s", &med.country);

fscanf(file, "%25s", &med.date);

for (int j = 0; j < MAX\_PHARMACIES; j++)

fscanf(file, "%5d", &med.pharmacy\_number[j]);

fscanf(file, "%10d\n", &med.price);

for (int j = 0; j < NUM\_SORT\_FIELDS; j++)

{

if (choice\_request[j] == 1) {

switch (j) {

case 0:

char name[MAX\_LENGTH];

if (i == 0) {

printf("Введите название лекарства\n(ШАБЛОН: Perekis\_vodoroda)\nНазвание:\t");

clear(stdin);

scanf\_s("%s", &name, MAX\_LENGTH);

}

is\_request\_flag = request\_flag = request\_flag && name\_request(med.name, name);

break;

case 1:

char country[MAX\_LENGTH];

if (i == 0) {

printf("Введите страну-производителя лекарства\n(ШАБЛОН: Arabskie\_Emirati)\nСтрана:\t");

clear(stdin);

scanf\_s("%s", &country, MAX\_LENGTH);

}

is\_request\_flag = request\_flag = request\_flag && country\_request(med.country, country);

break;

case 2:

char date[MAX\_LENGTH];

if (i == 0) {

printf("Введите дату производства лекарства\n(ШАБЛОН [год.месяц.число] с нулями: 2012.09.02)\nДата:\t");

do {

date\_flag = true;

clear(stdin);

scanf\_s("%s", &date, MAX\_LENGTH);

for (int k = 0; k < LENGTH\_DATE; k++)

{

if (date[k] != '.' && (date[k] < '0' || date[k] > '9') || strlen(date) != LENGTH\_DATE || date[4] != '.' || date[7] != '.')

{

printf("ВВОД НЕВЕРЕН\nВВОДИТЕ, ПОЖАЛУЙСТА, ПО ШАБЛОНУ\nДата:\t");

date\_flag = false;

break;

}

}

} while (date\_flag == false);

}

is\_request\_flag = request\_flag = request\_flag && date\_request(med.date, date);

break;

case 3:

int pharmacy\_number;

if (i == 0) {

do {

printf("Введите номер апетки\n(ШАБЛОН: 12)\nНомер аптеки:\t");

clear(stdin);

scanf\_s("%d", &pharmacy\_number);

} while (pharmacy\_number <= 0);

}

is\_request\_flag = request\_flag = request\_flag && pharmacy\_number\_request(med.pharmacy\_number, pharmacy\_number);

break;

case 4:

int price;

char choice\_price;

if (i == 0) {

do {

price\_flag = true;

printf("Введите цену лекарства (ШАБЛОН: 200)\nЦена:\t");

clear(stdin);

scanf\_s("%d", &price);

printf("знак > (не дешевле), < (не дороже), = (равно) (ШАБЛОН: <)\nЗнак:\t");

clear(stdin);

scanf\_s("%c", &choice\_price, MAX\_LENGTH);

if (choice\_price != '<' && choice\_price != '>' && choice\_price != '=' || price < 0) //в случае, когда пользователь вводит текст, price == 0

{

printf("ВВОД НЕВЕРЕН\nВВОДИТЕ, ПОЖАЛУЙСТА, ПО ШАБЛОНУ\n");

price\_flag = false;

break;

}

} while (price\_flag == false);

}

is\_request\_flag = request\_flag = request\_flag && price\_request(med.price, price, choice\_price);

break;

}

}

}

if (is\_request\_flag == true) {

if (title\_flag == false) {

title\_flag = true;

printTitles();

}

if (request\_flag == true)

{

printf(" \* ");

printf(" %5d \* ", med.id);

printf(" %25.25s \* ", med.name);

printf(" %25.25s \* ", med.country);

printf(" %25.25s \* ", med.date);

for (int j = 0; j < MAX\_PHARMACIES; j++) {

if (med.pharmacy\_number[j] != 0)

printf("%5d", med.pharmacy\_number[j]); //расситано для 5 аптек

else

printf(" ");

}

printf(" \* ");

printf(" %25d \* ", med.price);

printf("\n %s\n", stars);

}

}

}

if (title\_flag == false) printf("По вашему запросу ничего не найдено\n");

fclose(file);

}

bool name\_request(char med\_name[], char name[]) {// ФУНКЦИЯ ЗАПРОСА ПО НАЗВАНИЮ ЛЕКАРСТВА

if (strstr(med\_name, name) != NULL)

return true;

return false;

}

bool country\_request(char med\_country[], char country[]) {// ФУНКЦИЯ ЗАПРОСА ПО СТРАНЕ ЛЕКАРСТВА

if (strcmp(med\_country, country) == NULL)

return true;

return false;

}

bool date\_request(char med\_date[], char date[]) { // ФУНКЦИЯ ЗАПРОСА ПО ДАТЕ ЛЕКАРСТВА

if (strcmp(med\_date, date) > 0)

return true;

return false;

}

bool price\_request(int med\_price, int price, char choice\_price) {// ФУНКЦИЯ ЗАПРОСА ПО ЦЕНЕ ЛЕКАРСТВА

if (choice\_price == '<') {

if (med\_price <= price)

return true;

return false;

}

else if (choice\_price == '>') {

if (med\_price >= price) return true;

return false;

}

else {

if (med\_price == price) return true;

return false;

}

}

bool pharmacy\_number\_request(int med\_pharmacy\_number[], int pharmacy\_number) {// ФУНКЦИЯ ЗАПРОСА ПО НАЛИЧИЮ ЛЕКАРСТВА В АПТЕКЕ

for (int j = 0; j < MAX\_PHARMACIES; j++)

if (med\_pharmacy\_number[j] == pharmacy\_number)

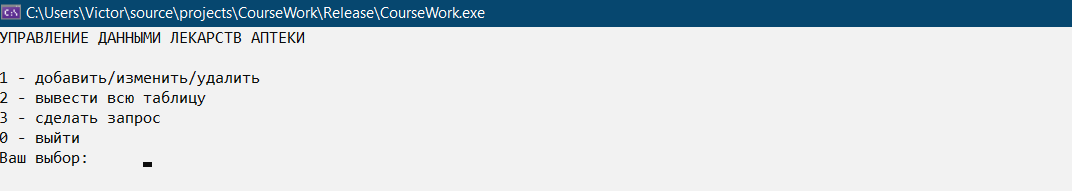
return true;

return false;

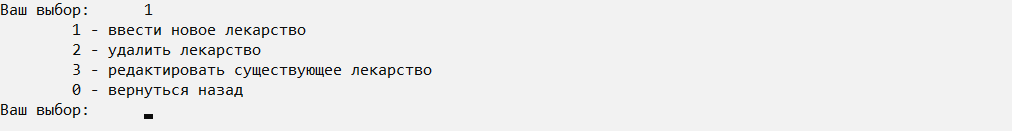
}

## Приложение Б Результаты работы программы

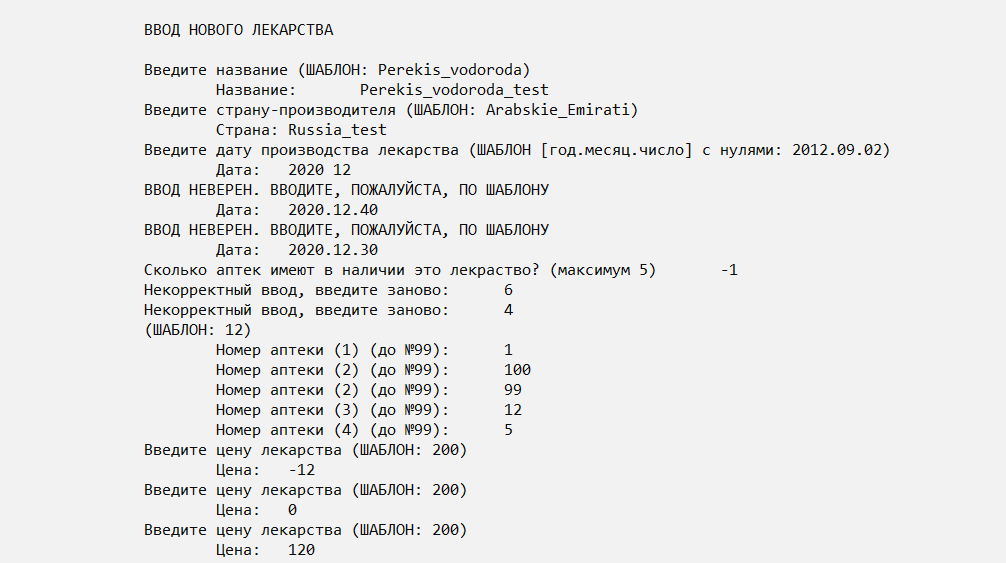
Главное меню:



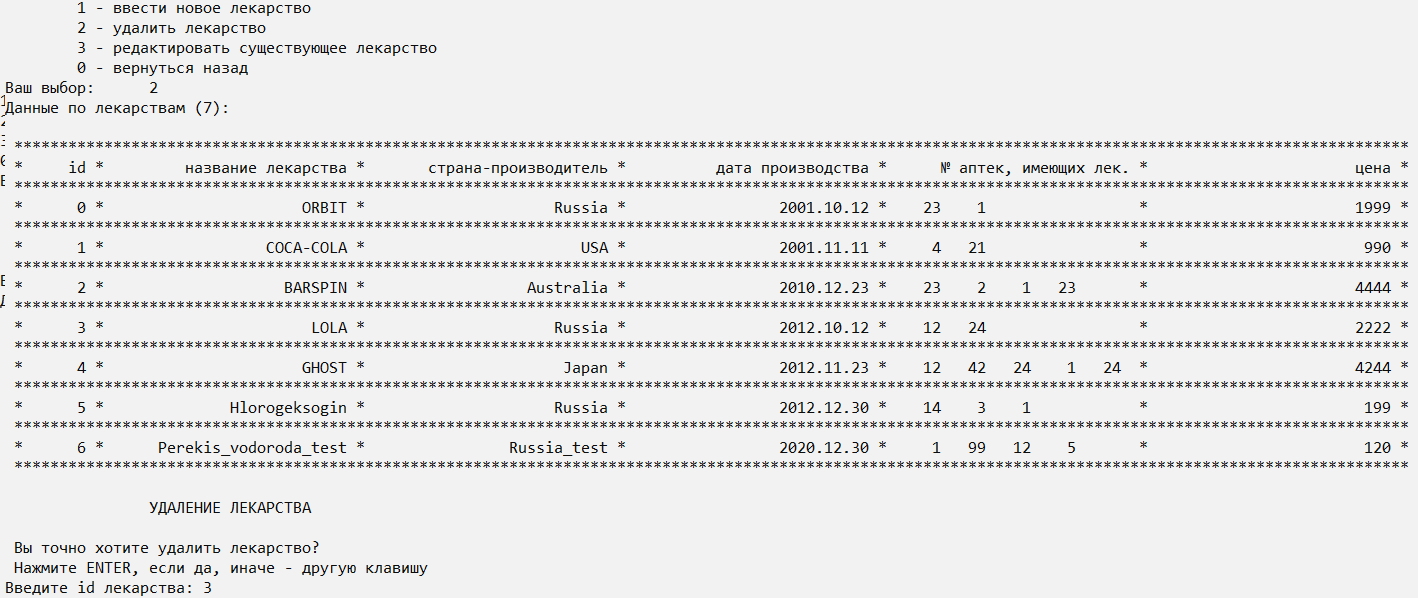
Главное меню/1



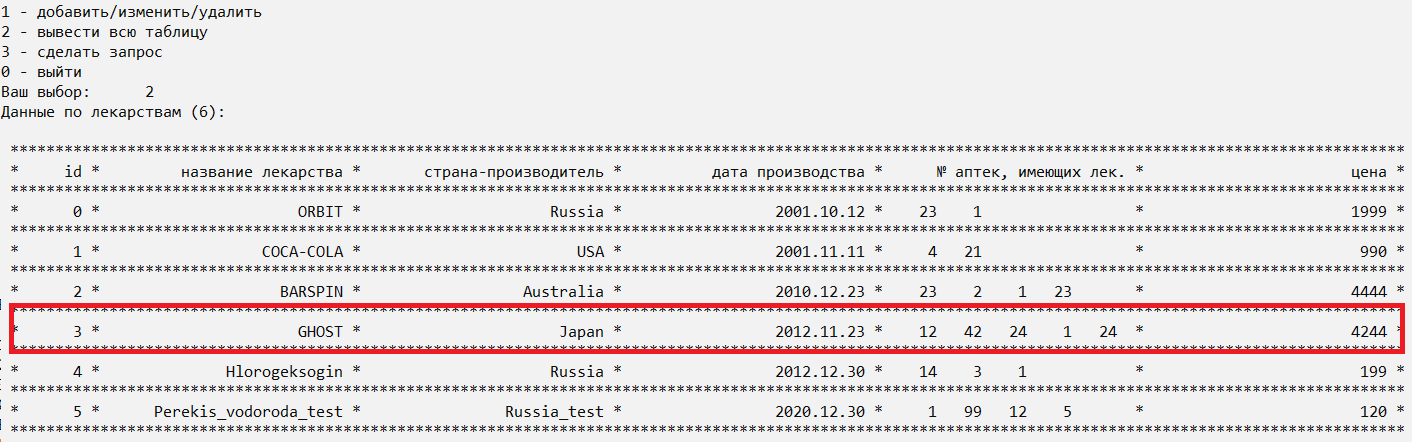
Главное меню/1/1 (также попытки неправильного ввода)



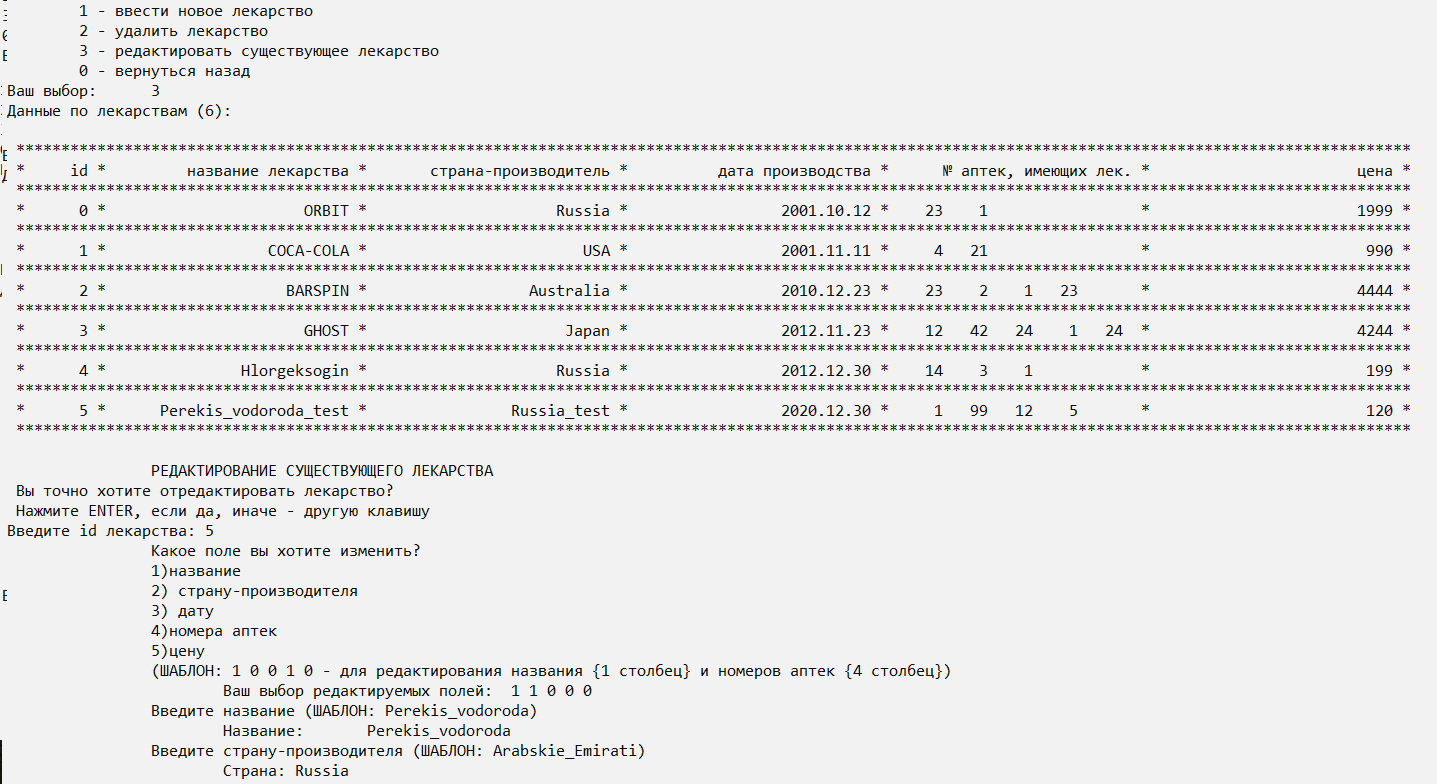
Главное меню/1/2 (пользователю показывается таблица, чтобы он мог выбрать по id)



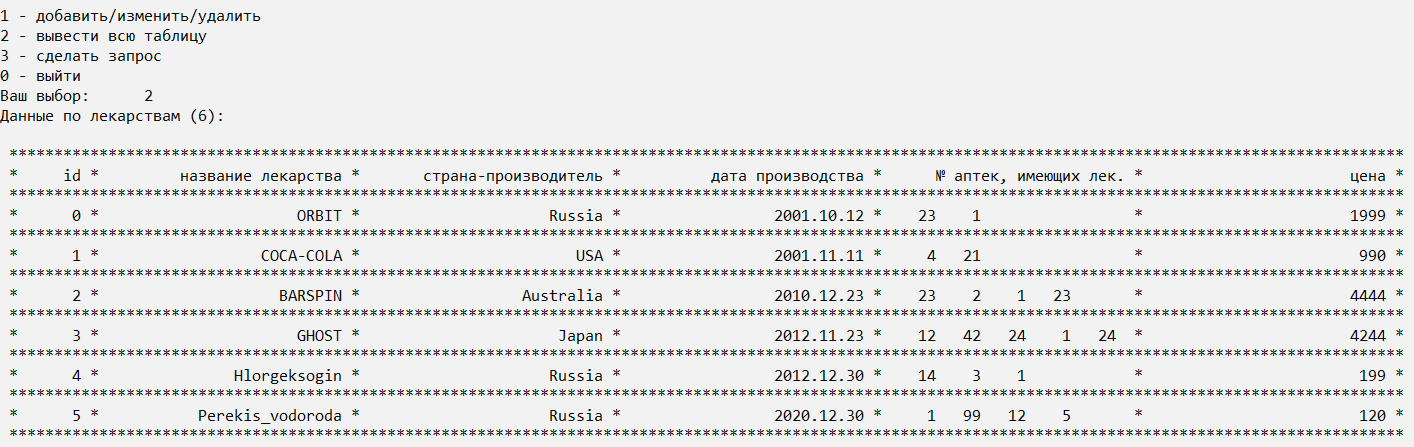
Главное меню/2 (сразу проверим, удалилась ли запись)



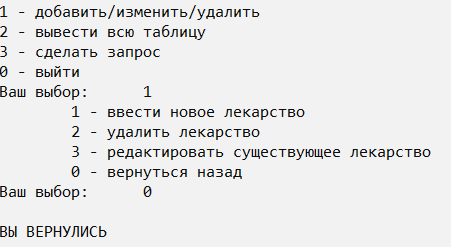
Главное меню/1/3 (пользователю показывается таблица, чтобы он мог выбрать по id)



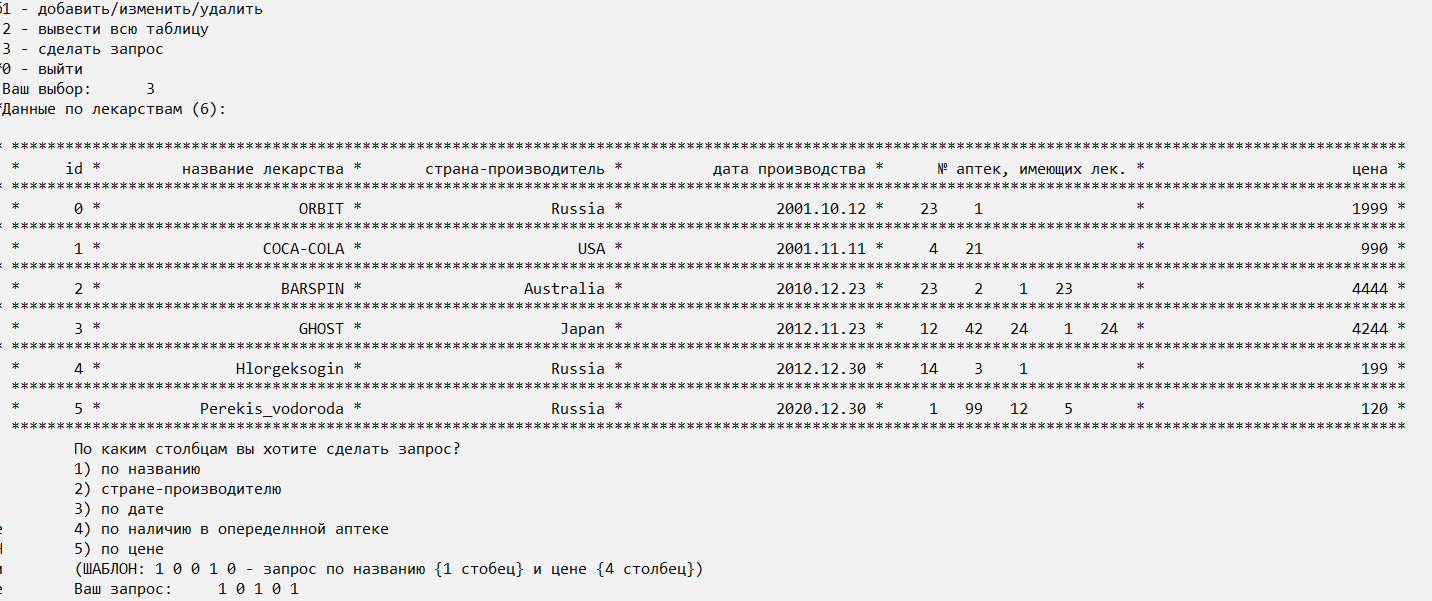
Главное меню/2 (сразу проверим, отредактировалась ли запись)



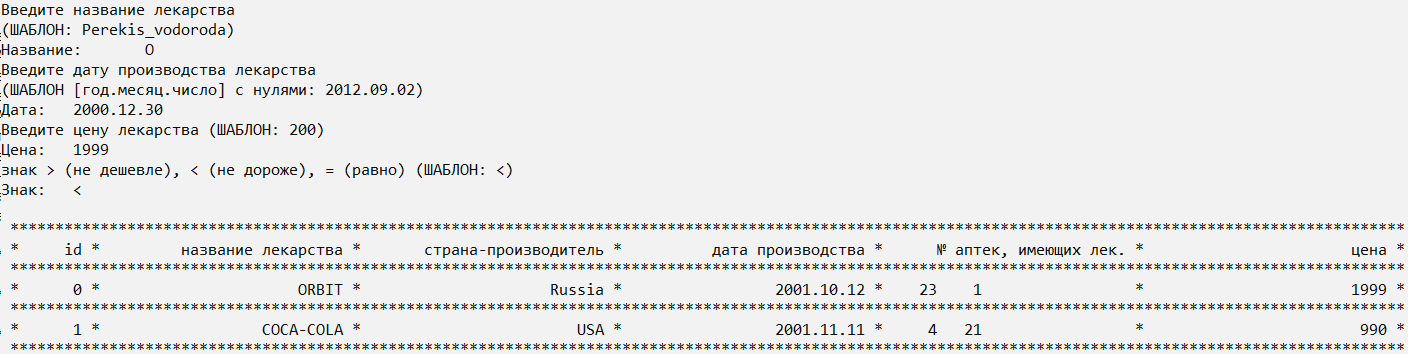
Главное меню/1/0 (возвращение к предыдущему шагу)



Главное меню/3



После выбор полей, вносим интересующие нас данные и получаем ответ:



Главное меню/0 (завершение работы)

