

Содержание

Введение	4
1 Анализ задания и постановка задач	5
2 Проектирование программы	6
3 Реализация программы	10
4 Методика и результаты тестирования	16
Заключение	19
Список используемых источников	20
Приложения	21

					БИБ.15480103.03.ПЗ			
Изм.	Лист	№ докум.	Подпись	Дата				
Разраб.		Бурунов И.В.			Однофайловая база данных «Магазин продуктов» (пояснительная записка)	Лит.	Лист	Листов
Провер.		Макарычев М.Ю.					3	26
						Учреждение образования «Полоцкий государственный университет» гр. 17-ИТд		
Н. Контр.								
Утв.								

Введение

Язык C++ представляет собой набор команд, которые говорят компьютеру, что необходимо сделать. Этот набор команд, обычно называется исходный код или просто код. Командами являются или «функции» или «ключевые слова». Ключевые слова (зарезервированные слова C/C++) являются основными строительными блоками языка. Функции являются сложными строительными блоками, так как записаны они в терминах более простых функций — это будет представлено в разрабатываемой программе, которая будет описана в пояснительной записке. Структура функций напоминает содержание книги. Содержание может показывать главы книги, каждая глава в книге может иметь своё собственное содержание, состоящее из пунктов, каждый пункт может иметь свои подпункты. Хотя C++ предоставляет много общих функций и зарезервированных слов, которые можно использовать.

Темой курсовой работы является разработка простейшей однофайловой базы данных. База данных содержит информацию о товарах в магазине, а именно: штрихкод, наименование, массу, единицу измерения массы, цену, количество на складе. Разрабатываемая база данных должна включать в себя следующий функционал:

- запись и загрузка файла;
- добавление новых записей, редактирование и удаление старых;
- сортировать записи по любому из полей базы данных в любом направлении;
- фильтровать записи по значению любого поля;
- осуществлять поиск записей по значению любого поля.

В ходе выполнения работы проведен анализ задания, описана постановка задач, рассмотрены основные теоретические сведения по теме курсовой работы. Далее описан процесс проектирования и реализации программы. В конце приведена методика и результаты тестирования разработанной простейшей однофайловой базы данных.

1 Анализ задания и постановка задач

Курсовая работа представляет собой создание простейшей однофайловой базы данных в предметной области «Магазин продуктов». Программа должна осуществлять следующий функционал:

- запись и загрузка файла;
- добавление новых записей, редактирование и удаление старых;
- сортировать записи по любому из полей базы данных в любом направлении;
- фильтровать записи по значению любого поля;
- осуществлять поиск записей по значению любого поля.

Обработку данных производить в динамическом массиве. Выделение динамической памяти осуществляется автоматически. Чтение и запись данных в файл базы данных производится поэлементно. Программа должна обладать дружественным и интуитивно понятным интерфейсом и проводить проверку на корректность вводимых данных. При сдаче курсовой работы на проверку должен быть предоставлен файл базы данных, содержащий не менее 30 записей. Данные должны быть введены корректно, не допускается ввод «мусора». Содержание предоставляемой для проверки базы данных должно позволить осуществить проверку всех функций программы.

База данных содержит информацию о товарах в магазине: штрихкод (строка), наименование (строка), массу (целое число), единицу измерения массы (строка), цену (вещественное число), количество на складе (целое число).

Даже при наличии большого числа специализированных программных продуктов, используемых для создания и ведения баз данных, разработка новых не теряет своей актуальности. Это обусловлено тем, что не все из них отражают ту специфику, все те необходимые аспекты, которые зависят от предметной области и требований. К тому же постоянно возникают новые задачи, изменяются требования к существующим проектам, изменяется сам предмет. Поэтому уже существующие программы зачастую не могут применяться в полной мере. В связи с этим разработку базы данных можно считать актуальной и распространённой задачей.

2 Проектирование программы

Для реализации курсовой работы необходимо разработать простейшую однофайловую базу данных. База данных – это представленная в объективной форме совокупность самостоятельных материалов (статей, расчётов, нормативных актов, судебных решений и иных подобных материалов), систематизированных таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью электронной вычислительной машины (ЭВМ) [10].

В данной работе база данных представляет собой совокупность записей, хранящихся в типизированном файле.

Запись – это структура, состоящая из фиксированного числа компонент, называемых полями. Данные одного поля разных записей имеют один и тот же тип, а разных полей могут иметь разные типы [10].

Файл – это упорядоченная последовательность однотипных компонентов, расположенных на внешнем носителе. Файлы предназначены только для хранения информации, а обработка этой информации осуществляется программами [10].

Файл, определенный стандартным или пользовательским типом данных, называется типизированным.

Используемые в программировании данные можно разделить на две большие группы (рисунок 2.1):

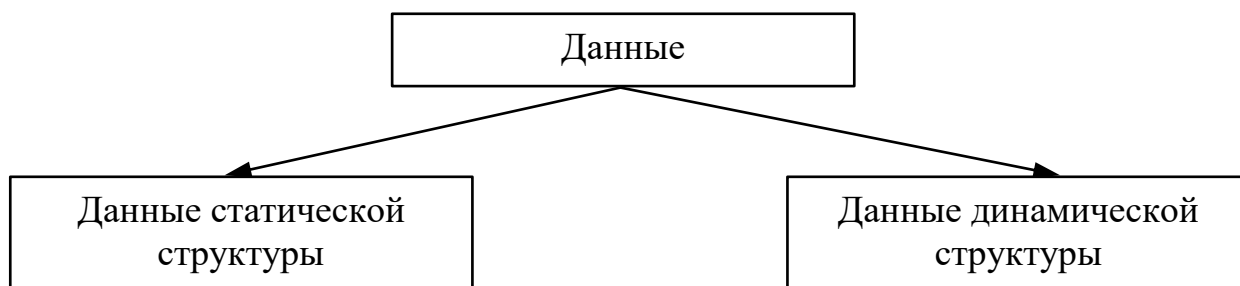


Рисунок 2.1 – Группы способов хранения данных в программировании

Данные статической структуры – это данные, взаиморасположение и взаимосвязи элементов которых всегда остаются постоянными.

Данные динамической структуры – это данные, внутреннее строение которых формируется по какому-либо закону, но количество элементов, их взаиморасположение и взаимосвязи могут динамически изменяться во время выполнения программы, согласно закону формирования [10].

Во время работы программы, данные, из типизированного файла перемещаются в динамический массив. Динамическим называется массив, размер которого может изменяться во время исполнения программы. Возможность изменения размера отличает динамический массив от статического, размер которого задаётся на момент компиляции программы. Динамический массив даёт возможность более гибкой работы с данными, так как позволяет не прогнозировать хранимые объёмы данных, а регулировать размер массива в соответствии с реально необходимыми объёмами.

Динамические массивы используются для обработки наборов однородных данных, размер которых неизвестен точно на момент написания

программы, но которые потенциально могут разместиться в доступной памяти. Использование динамических массивов позволяет выделить ровно столько памяти, сколько реально необходимо (сразу, если задача позволяет определить объём до загрузки данных, либо в процессе загрузки, расширяя массив по мере необходимости), загрузить все данные в один массив и единообразно их обработать. Недостатки, однако, имеются и у этой стратегии:

- снижение скорости работы из-за накладных расходов на изменение размера динамического массива;
- потенциальное снижение надёжности: при экстремально большом объёме входных данных попытка увеличить массив до соответствующих размеров может привести к внезапному существенному замедлению или даже отказу программы из-за недостатка свободной памяти [10].

Данные сортируются при помощи алгоритма быстрой сортировки.

Алгоритм сортировки – это алгоритм для упорядочивания элементов в списке. В случае, когда элемент списка имеет несколько полей, поле, служащее критерием порядка, называется ключом сортировки.

Алгоритм быстрой сортировки состоит из трёх шагов:

1. Выбрать элемент из массива. Назовём его опорным.
2. Разбиение: перераспределение элементов в массиве таким образом, что элементы меньше опорного помещаются перед ним, а больше или равные после.
3. Рекурсивно применить первые два шага к двум подмассивам слева и справа от опорного элемента. Рекурсия не применяется к массиву, в котором только один элемент или отсутствуют элементы.

Структура программы имеет следующий вид (рисунок 2.2):

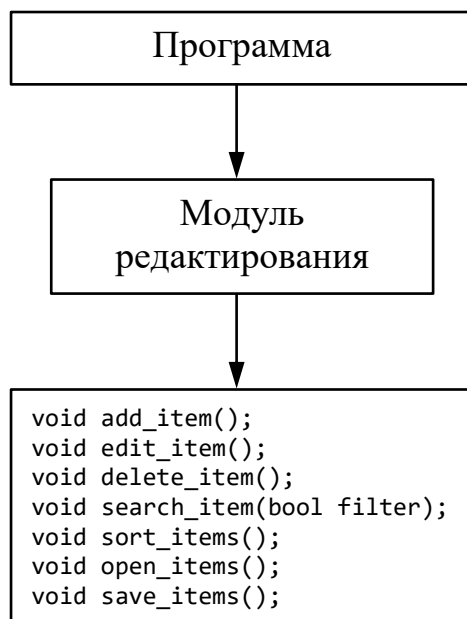


Рисунок 2.2 – Структура программы

Данные в программе реализованы в виде структуры: Product. Эта структура хранит в себе следующие поля:

- barcode – штрихкод товара;
- name – наименование товара;
- size – масса товара;
- unit – единица измерения массы товара;
- price – цена товара;
- amount – количество товара на складе.

Данные в программе хранятся в виде массива, который объявлен глобально.

Данные должны записываться в типизированный файл для хранения и переноса на другой компьютер. Корректность файла базы данных проверяется при помощи первой проверочной записи. В начало файла записывается строка с заранее описанной информацией. При считывании файла проверяется ее наличие. Отсутствие строки означает, что файл базы данных был создан не в рамках разрабатываемой в данной курсовой работе программе, и не может быть открыт. Все записи из файла, кроме проверочной, копируются в список, после чего они становятся доступны для редактирования, удаления и сортирования. После завершения работы со списком требуется сохранить файл, иначе данные будут утеряны.

Алгоритм решения задачи представлен на рисунке 2.3:

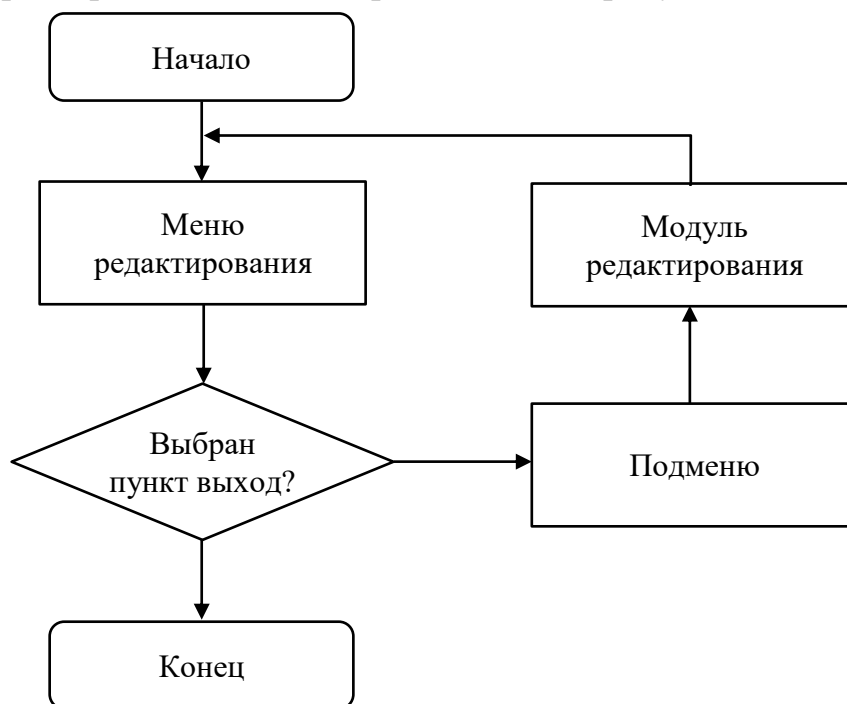


Рисунок 2.3 – Обобщённая схема алгоритма решения задачи

Алгоритм (рисунок 2.3) представляет собой вход в программу, после которого пользователю доступно меню редактирования с соответствующими пунктами. После выбора одно из них пользователь попадает в подменю программы для выполнения одной из предложенных

операций. При выборе пункта выхода из программы, процесс взаимодействия программы с пользователем прекращается.

3 Реализация программы

В программе предусмотрены следующие функции:

- `bool main_menu()` – меню редактирования;
- `void add_item()` – добавить запись;
- `void edit_item()` – редактировать запись;
- `void delete_item()` – удалить запись;
- `void search_item(bool filter)` – поиск и фильтрация записей;
- `void sort_items()` – сортировка записей;
- `void open_items()` – открыть базу данных;
- `void save_items()` – сохранить базу данных;
- `int get_int(int min, int max)` – получить целое число;
- `float get_float(float min, float max)` – получить вещественное число;
- `void get_str(string &str, unsigned &max)` – получить строку;
- `void get_item(unsigned i)` – вывести в консоль запись;
- `void get_list()` – вывести все записи в консоль;
- `void get_max_lenght()` – получить размеры самых длинных строк;
- `void read_string(ifstream &fin, string &str)` – читать строку из файла;
- `void write_string(ofstream &fout, string &str)` – записать строку в файл.

Функция `bool main_menu()` предоставляет меню редактирования пользователю для дальнейших действий.

Функция `void add_item()` добавляет новый элемент в массив. Новый элемент будет добавлен в конец.

Функция `void edit_item()` предлагает выбрать книгу, которую необходимо отредактировать. После ввода новых данных заменяет старые данные новыми у выбранной записи.

Функция `void delete_item()` удаляет запись по указанному пользователем номеру.

Функция `void search_item(bool filter)` которая в зависимости от значения указанного аргумента осуществляет поиск по данным или фильтрует записи массива. Пользователю предлагается ввести столбец и значение. В конце работы функция выводит результат.

Функция `void sort_items()` производит сортировку по выбранному пользователем столбцу и направлению. В данной функции применен метод быстрой сортировки.

Функция `void open_items()` открывает бинарный файл базы данных с последующей выгрузкой данных в массив. Пользователю предлагается ввести имя файла. Ранее хранящиеся данные из массива удаляются.

Функция `void save_items()` создает бинарный файл базы данных с именем указанным пользователем. В файл записываются данные хранящиеся в массиве.

Функция `int get_int(int min, int max)` предлагает пользователю ввести целое число и проверяет его корректность. При вводе некорректного числа пользователю будет предложено повторить ввод.

Функция `float get_float(float min, float max)` предлагает пользователю ввести вещественное число и проверяет его корректность. При вводе некорректного числа пользователю будет предложено повторить ввод.

Функция `void get_str(string &str, unsigned &max)` предлагает пользователю ввести строку и проверяет её корректность. При вводе некорректной строки пользователю будет предложено повторить ввод.

Функция `void get_item(unsigned i)` выводит в консоль указанный элемент массива.

Функция `void get_item(unsigned i)` выводит в консоль все данные содержащиеся в массиве.

Функция `void get_max_lenght()` получает самую длинную строку в каждом содержащем строки столбце массива.

Функция `void read_string(istream &fin, string &str)` читает из бинарного файла строку и возвращает её для дальнейшей выгрузки в массив.

Функция `void write_string(ofstream &fout, string &str)` записывает в бинарный файл строку.

При запуске программы перед пользователем появляется меню редактирования с соответствующими пунктами. Данное меню представлено на рисунке 3.1.

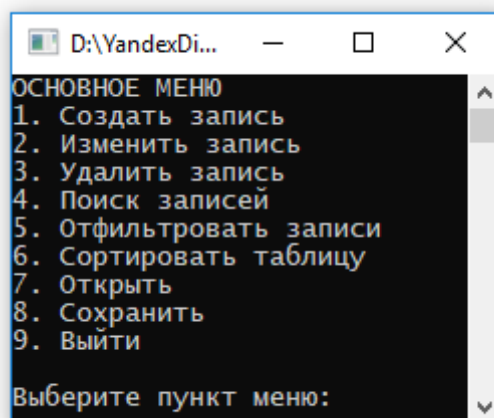


Рисунок 3.1 – Меню редактирования

После того как пользователь начал процесс взаимодействия с программой, ему необходимо открыть уже созданную базу данных выбрав 7-ой пункт меню или создать новую начав добавлять в нее записи при помощи 1-ого пункта меню. При создании записи пользователю будет предложено ввести необходимые данные. Ввод данных этих данных отображен на рисунке 3.2.

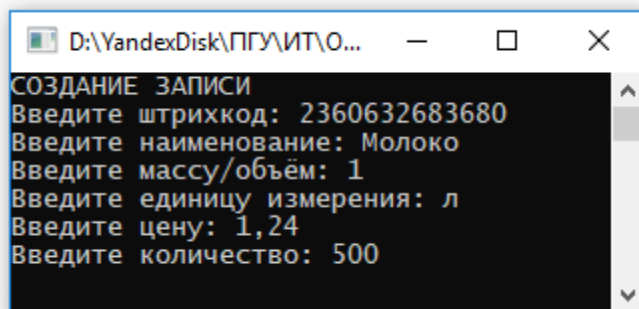


Рисунок 3.2 – Создание записи

Для редактирования записи в меню редактирования необходимо выбрать 2-ой пункт меню. Будет предложено ввести номер записи для редактирования и необходимые данные аналогичные тем, что вводились при создании записи. Выбор записи для редактирования отображен на рисунке 3.3.

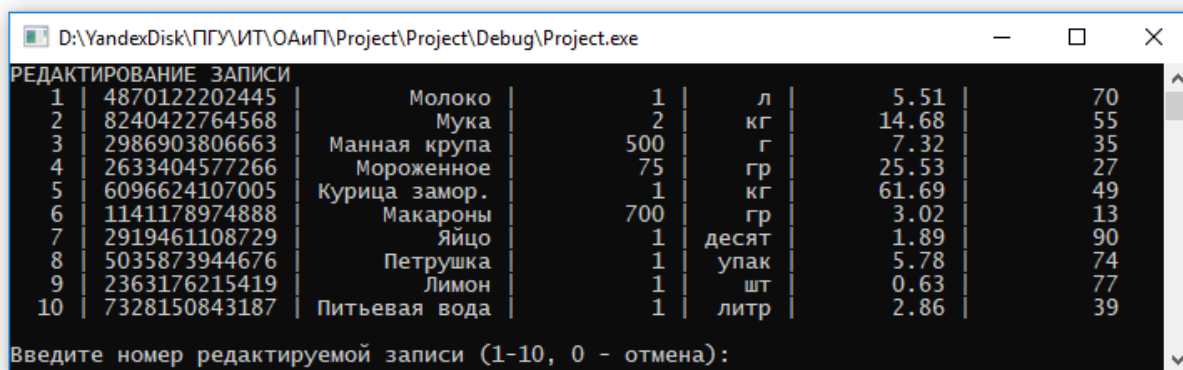


Рисунок 3.3 – Выбор записи для редактирования

Чтобы удалить запись из базы данных следует выбрать 3-ий пункт меню редактирования и указать номер записи из списка. Выбор записи для удаления отображен на рисунке 3.4.

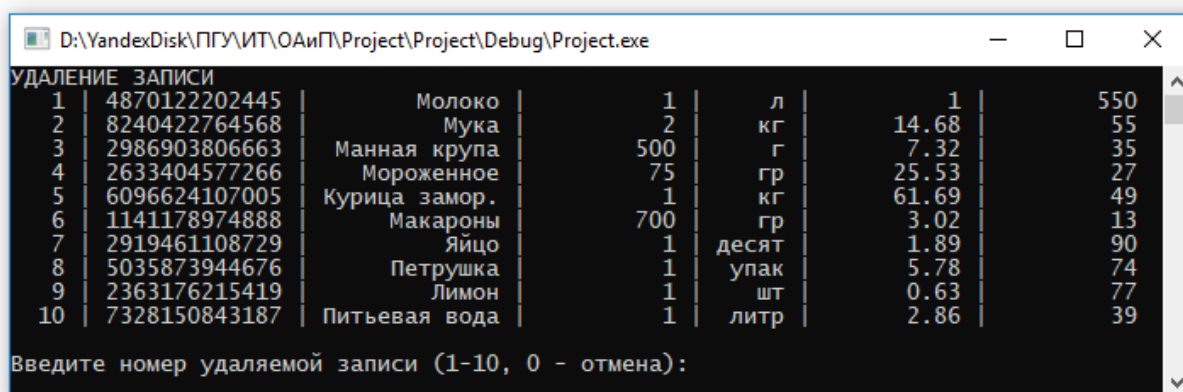


Рисунок 3.4 – Выбор записи для удаления

Чтобы осуществить поиск по записям необходимо выбрать 4-ый пункт меню редактирования. Пользователю будет предложено указать столбец для поиска и искомое значение. Процесс ввода данных для поиска и вывод результата показан на рисунках 3.5 и 3.6.

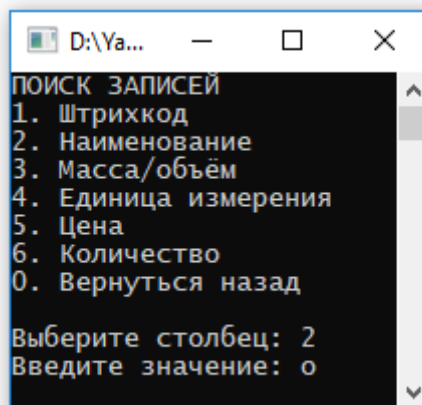


Рисунок 3.5 – Поиск записей

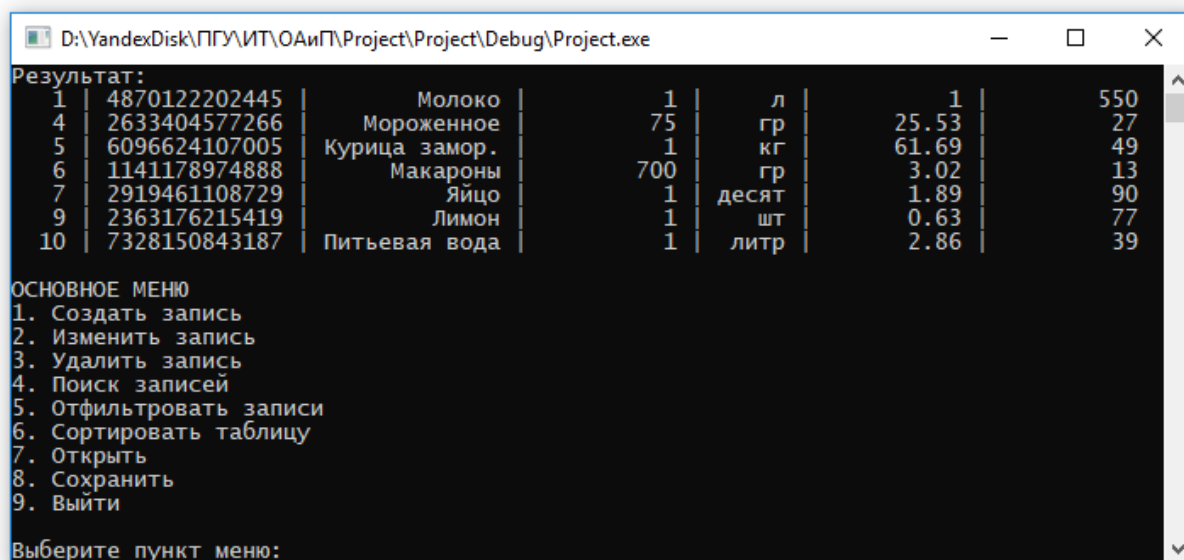


Рисунок 3.6 – Результат поиска

Фильтрация записей находится под 5-ым номером в меню редактирования. Ввод данных и отображение результата аналогичны поиску записей. Процесс ввода данных показан на рисунке 3.7.

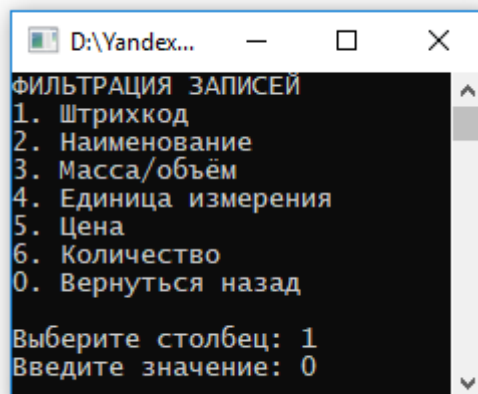


Рисунок 3.7 – Фильтрация записей

Выберите 6-ой пункт меню редактирования, чтобы отсортировать записи в базе данных. Будет предложено выбрать столбец и направление

сортировки. Процесс ввода данных для поиска и вывод результата показан на рисунках 3.8 и 3.9.

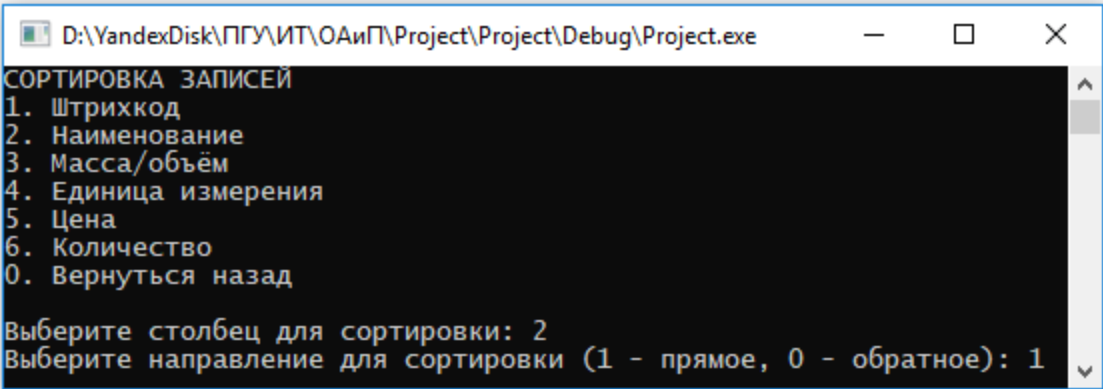


Рисунок 3.8 – Сортировка записей

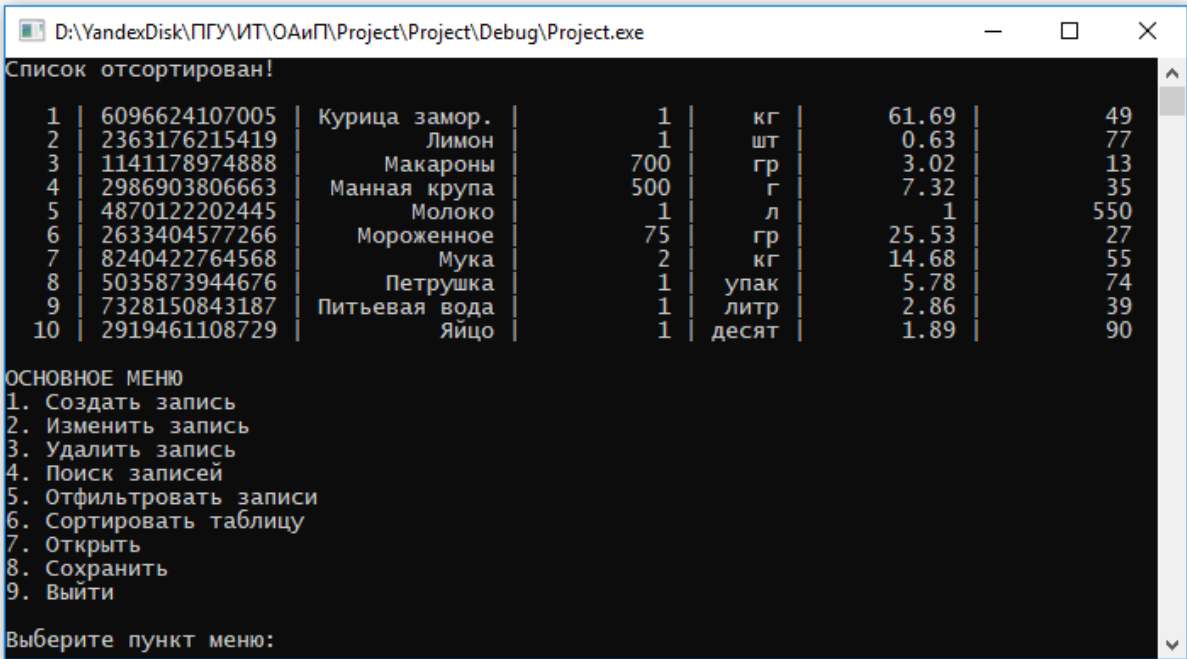


Рисунок 3.9 – Результат сортировки

Для открытия файла необходимо выбрать 7-ой пункт меню редактирования и указать имя файла без расширения, как показано на рисунке 3.10. Файл должен находиться в папке с программой. Данные находящиеся перед открытием в массиве будут удалены. Выгруженные из файла в базу данных данные будут отображены, как показано на рисунке 3.11.

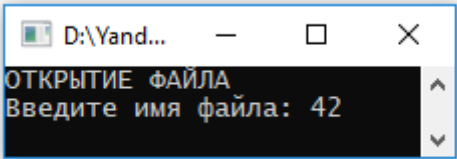


Рисунок 3.10 – Открытие файла

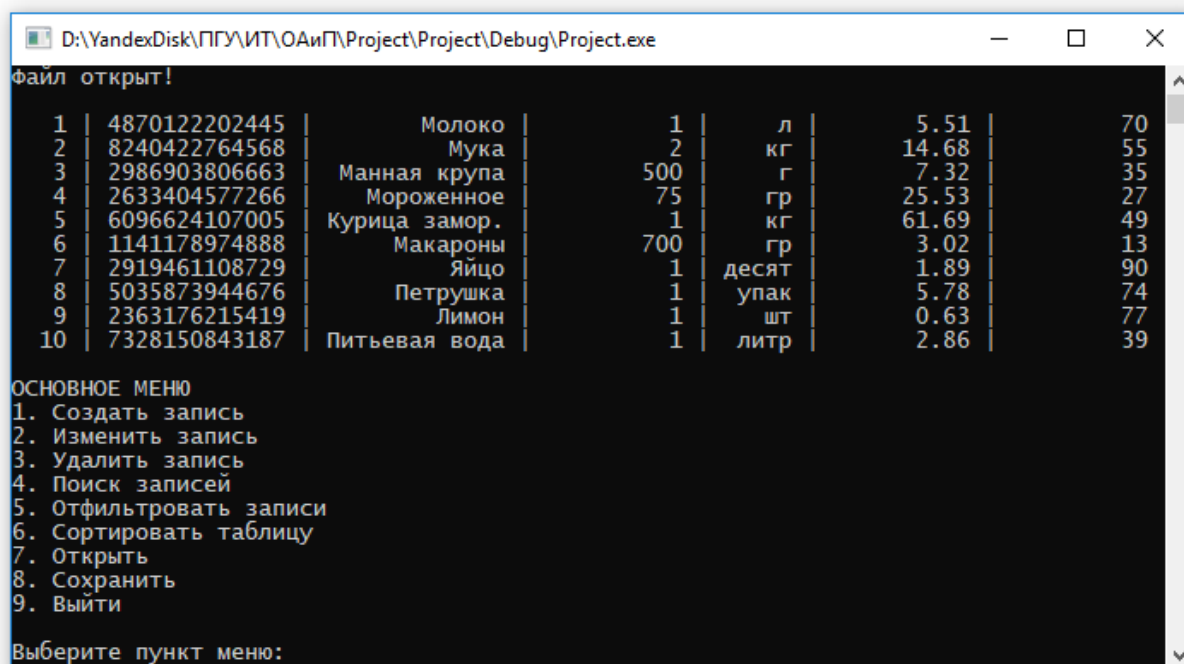


Рисунок 3.11 – Результат открытия файла

По окончании работы с базой данных необходимо сохранить данные, выбрав 8-ой пункт меню редактирования и введя имя файла без расширения, как показано на рисунке 3.12. Несохраненные данные после закрытия программы будут утеряны.

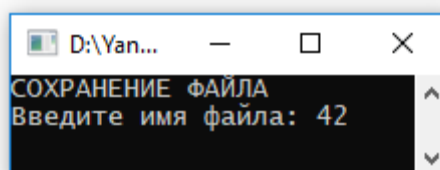


Рисунок 3.12 – Сохранение файла

4 Методика и результаты тестирования

Наименование программы – «Простейшая однофайловая база данных «Магазин продуктов»;

Обозначение программы – App.exe;

Требования к системе:

Поддерживаемые операционные системы:

- Windows 7 (архитектуры x86 и x64);
- Windows 8 (архитектуры x86 и x64);
- Windows 10 (архитектуры x86 и x64);

Поддерживаемые архитектуры:

- 32-разрядные (x86);
- 64-разрядные (x64).

Требования к оборудованию:

- процессор с тактовой частотой 1,6 ГГц или более быстрый;
- 1 ГБ ОЗУ;
- 20 МБ свободного места на жестком диске.

Планом испытаний является проведение тестирования программы. Для проверки работы программы будут созданы разного рода ситуации, например: указание несуществующего файла базы данных или добавление новых записей с некорректно введенными данными, проверка функциональных клавиш перехода между пунктами меню и т.д. Тестирования программы подробно описано ниже.

Тестирование программы многократно производилось методом ввода случайных данных. При вводе значений пунктов меню, добавлении, редактировании, удалении, поиске, фильтрации записей данные вводились как корректно, как этого требует программа, так и некорректно.

Результаты тестирования приведены в таблице 4.1.

Таблица 4.1 – Результаты тестирования

Описание теста	Выполняемые действия	Реакция программы	Вывод
Запуск программы	Активирован исполняемый файл программы	Появилось меню редактирования	Выполнено
Открытие базы данных	Указано имя файла	База загрузилась. Отобразился список загруженных данных.	Выполнено
Попытка загрузить базу из некорректного файла	Указано имя несуществующего файла	База не загрузилась. Система выдала сообщение о том, что файла с таким именем не существует.	Выполнено

Продолжение таблицы 4.1

Добавление новой записи	Корректно введены данные	Система выдала сообщение о том, что запись добавлена. Запись появилась в списке записей.	Выполнено
Добавление новой записи с некорректными данными	Данные введены некорректно	Запись не добавлена. Программа сообщает об ошибке. Предложено повторить ввод.	Выполнено
Редактирование записи	Указан корректный номер записи. Введены корректные данные.	Система выдала сообщение о том, что запись отредактирована. Запись отображается в списке с новыми данными.	Выполнено
Редактирование записи с некорректными данными	Данные введены некорректно	Запись не отредактирована. Программа сообщает об ошибке. Предложено повторить ввод.	Выполнено
Удаление записи	Указан корректный номер записи	Система выдала сообщение о том, что запись удалена. Запись отсутствует в списке.	Выполнено
Удаление записи с некорректным номером	Указан некорректный номер записи	Запись не удалена. Программа сообщает об ошибке. Предложено повторить ввод.	Выполнено
Поиск записей	Введен номер столбца и значение для поиска	Результаты, удовлетворяющие запросу, выводятся на экран	Выполнено
Поиск записей при некорректно введенных данных	Указан номер столбца с числами. В качестве значения для поиска задана строка.	Программа сообщает об ошибке. Предложено повторить ввод.	Выполнено

Продолжение таблицы 4.1

Фильтрация записей	Введен номер столбца и значение для фильтрации	Результаты, удовлетворяющие запросу, выводятся на экран	Выполнено
Фильтрация записей при некорректно введенных данных	Указан номер столбца с числами. В качестве значения для фильтрации задана строка.	Программа сообщает об ошибке. Предложено повторить ввод.	Выполнено
Сортировка	Выбран столбец и направление сортировки	Программа выполнила сортировку по указанному полю и в указанном направлении.	Выполнено
Сохранение файла	Введено имя файла	Система выдала сообщение о том, что файл сохранен. После открытия этого файла данные загружены корректно.	Выполнено
Выход из программы	В меню редактирования выбран выход из программы	Программа завершилась корректно	Выполнено

В ходе тестирования программы были найдены и устранены ошибки, которые приводили к неправильной работе всей программы или неправильным действиям. Были проверены следующие функции:

- добавление новой записи;
- редактирование существующей записи;
- удаление записи;
- поиск записей;
- фильтрация записей;
- сортировка записей;
- открытие файла базы данных;
- сохранение файла базы данных;
- выход из программы.

После выполнения тестирования программа готова к использованию.

Заключение

В результате выполнения курсовой работы, была разработана простейшая однофайловая база данных в предметной области «Магазин продуктов». Результат работы имеет консольный вариант, который выполняет весь необходимый функционал в соответствии с поставленной задачей. Пользователь с лёгкостью сможет ознакомиться со всей необходимой ему информацией из приложения, процесс работы законченного приложения был продемонстрирован в пояснительной записке.

При выполнении курсовой работы были закреплены навыки работы на языке программирования C++ и приобретены навыки написания консольных приложений. Сделаны соответствующие выводы о необходимости внедрения информационных технологий в различные сферы жизнедеятельности, о том, как это действительно упрощает работу с информацией.

Программа имеет специфическую направленность – вся работа ведётся над информацией о продуктах в магазине. Поэтому использовать её сможет узкий круг людей, но для них она окажется весьма полезной.

Список используемых источников

1. В. В. Подбельской, С. С. Фомин. Программирование на языке Си++. М.: ФиС, 2012, 600 с. (рекомендуемый учебник по направлению «Информатика и ВТ»).
2. Х. М. Дейтел., П. Дж. Дейтел. Как программировать на Си++. М.: ЗАО БИНОМ, 2007, 1000 с. (первые 600 страниц - изложение обычного Си, наиболее полно соответствует структуре курса для 1-2семестра).
3. У. Топп, У. Форд. Структуры данных в Си++. М.: ЗАО БИНОМ, 2011, 800 с. (наиболее полно соответствует структуре курса для 3-4 семестра).
4. Климова Л. М. Основы практического программирования на языке Си++. М.: Издательство ПРИОР, 2014, 464 с.
5. Романов Е.Л. Язык Си. Структуры данных. Конспект лекций. Тестовые вопросы и задания к лабораторным работам. – Новосибирск, Изд-во НГТУ, 2012, -80 с. («методичка» для 2,3-го семестра, библиотека НГТУ – 73Р693)
6. Т. Фейсон. Объектно-ориентированное программирование на Borland C++ 4.5: Пер. с англ. - Киев: Диалектика, 1996. 544с.
7. Т. Сван. Освоение Borland C++ 4.5: Пер. с англ. - Киев: Диалектика, 1996. 544с.
8. Г. Шилдт. Самоучитель C++: Пер. с англ. - Санкт-Петербург: BHV-Санкт-Петербург, 1998. 620с.
9. К. Джамса. Учимся программировать на языке C++: Пер. с англ. - Москва: Мир, 1997. 320с.
10. Свободная энциклопедия [Электронный ресурс] – 2014. - Режим доступа: <http://ru.wikipedia.org> – Дата доступа: 15.05.2018.

Приложение А
(обязательное)
Блок-схемы алгоритмов

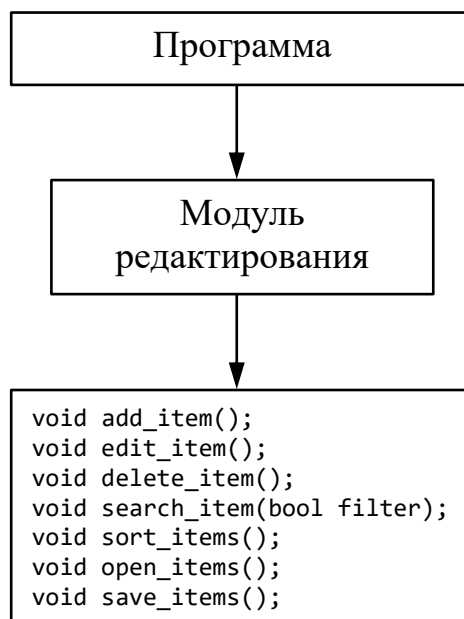


Рисунок А.1 – Структура программы

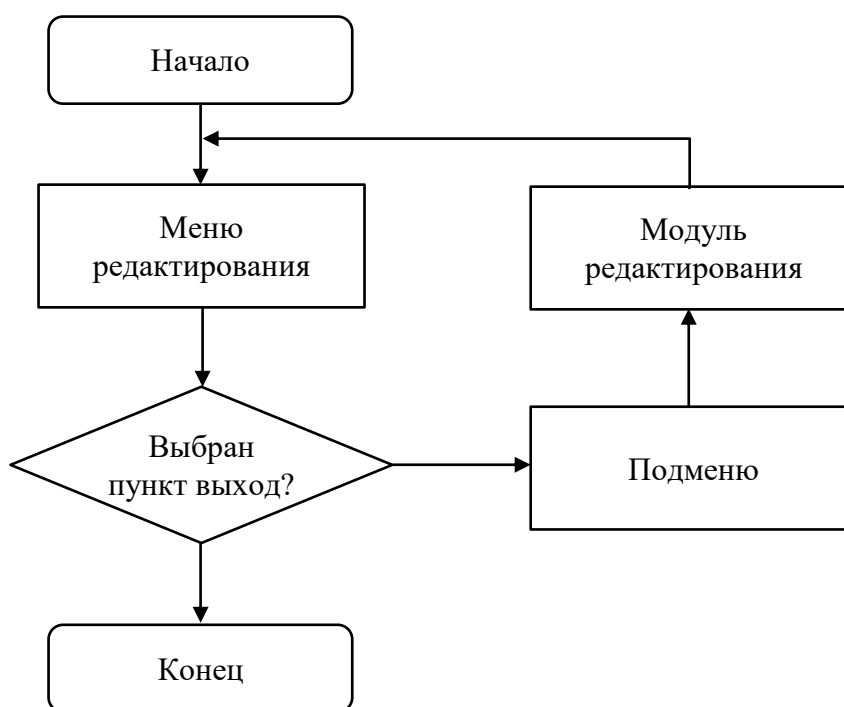


Рисунок А.2 – Обобщенная схема алгоритма решения задачи

Приложение Б (обязательное) Интерфейс программы

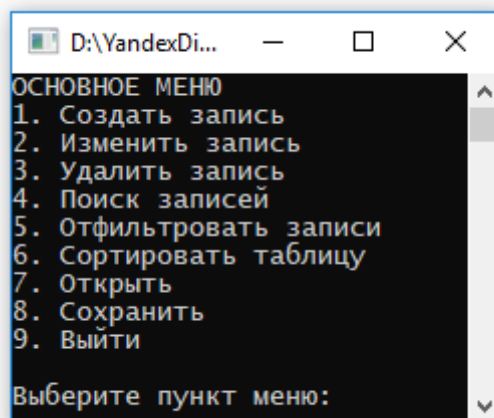


Рисунок Б.1 – Меню редактирования

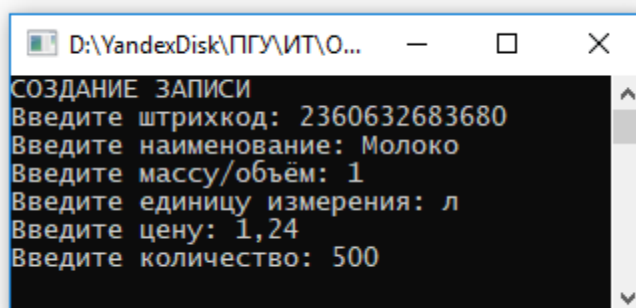


Рисунок Б.2 – Создание записи

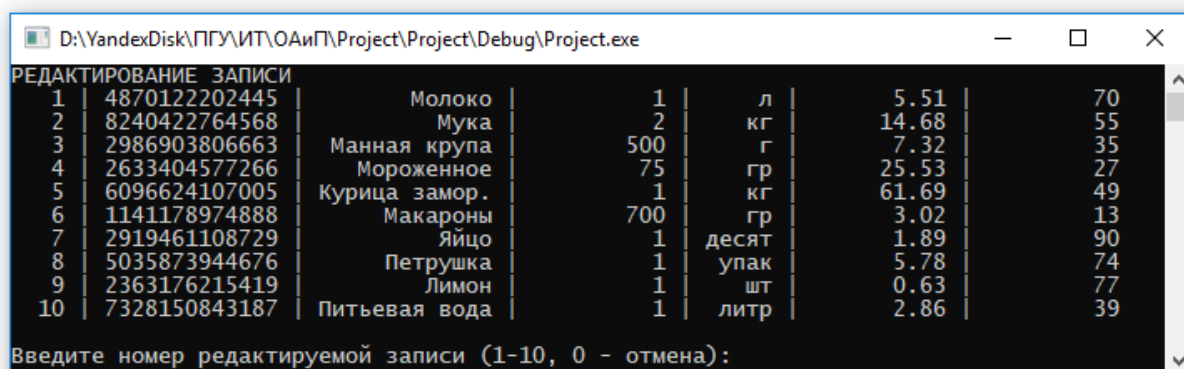


Рисунок Б.3 – Выбор записи для редактирования

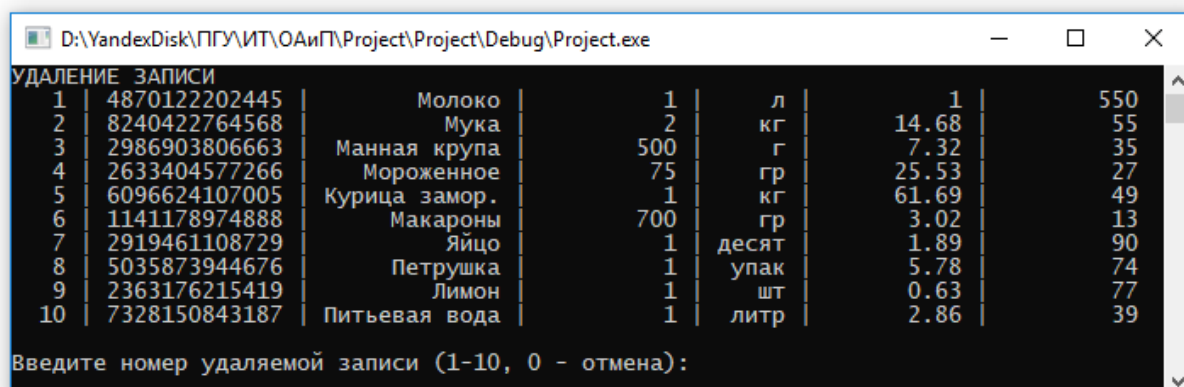


Рисунок Б.4 – Выбор записи для удаления

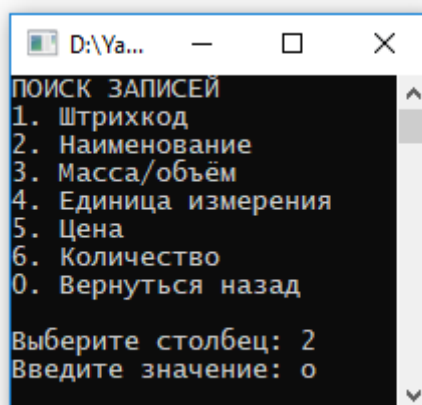


Рисунок Б.5 – Поиск записей

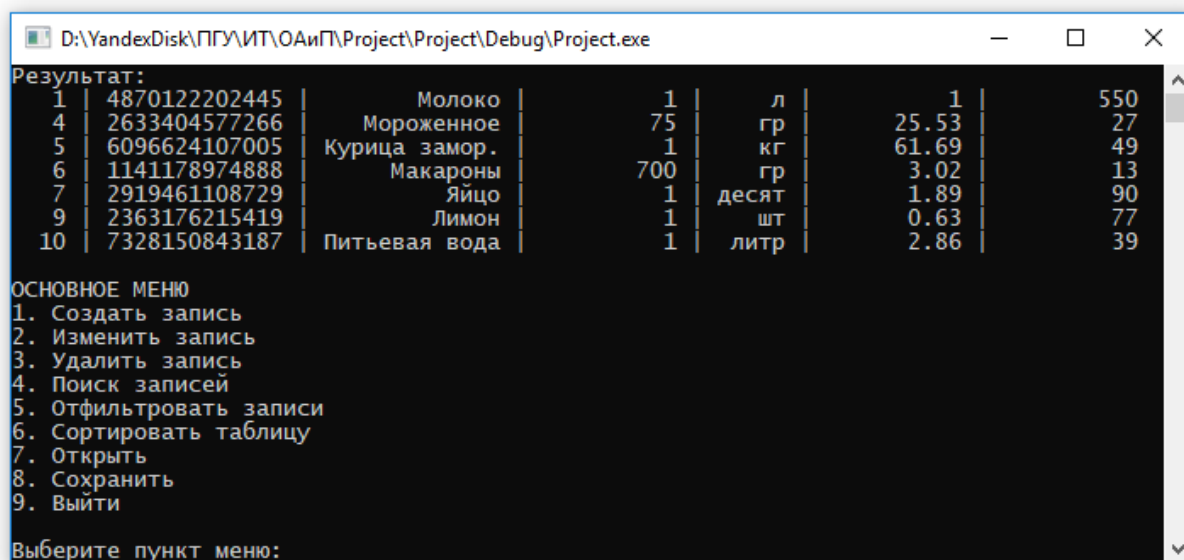


Рисунок Б.6 – Результат поиска

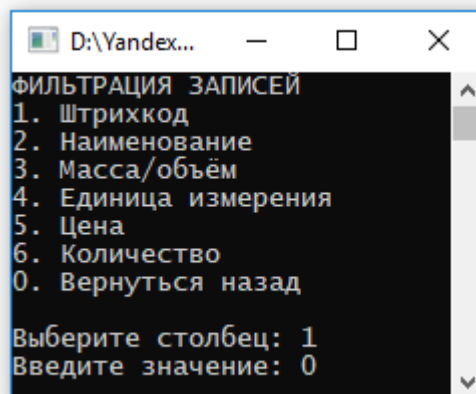


Рисунок Б.7 – Фильтрация записей

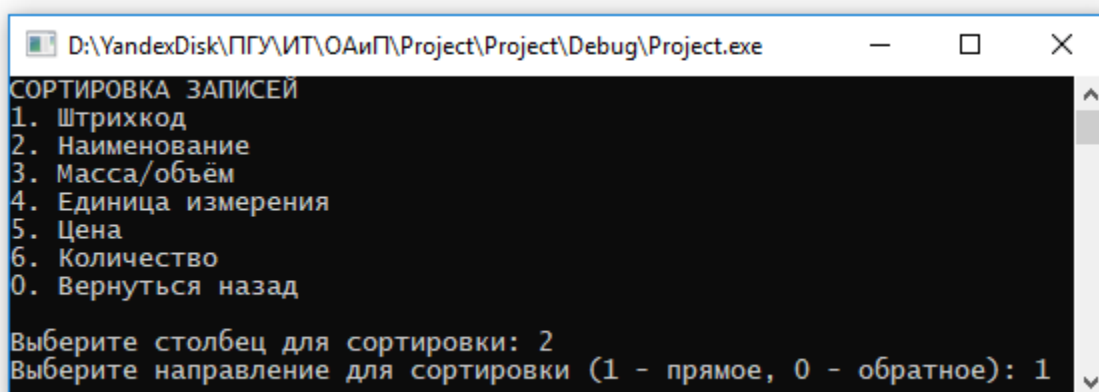


Рисунок Б.8 – Сортировка записей

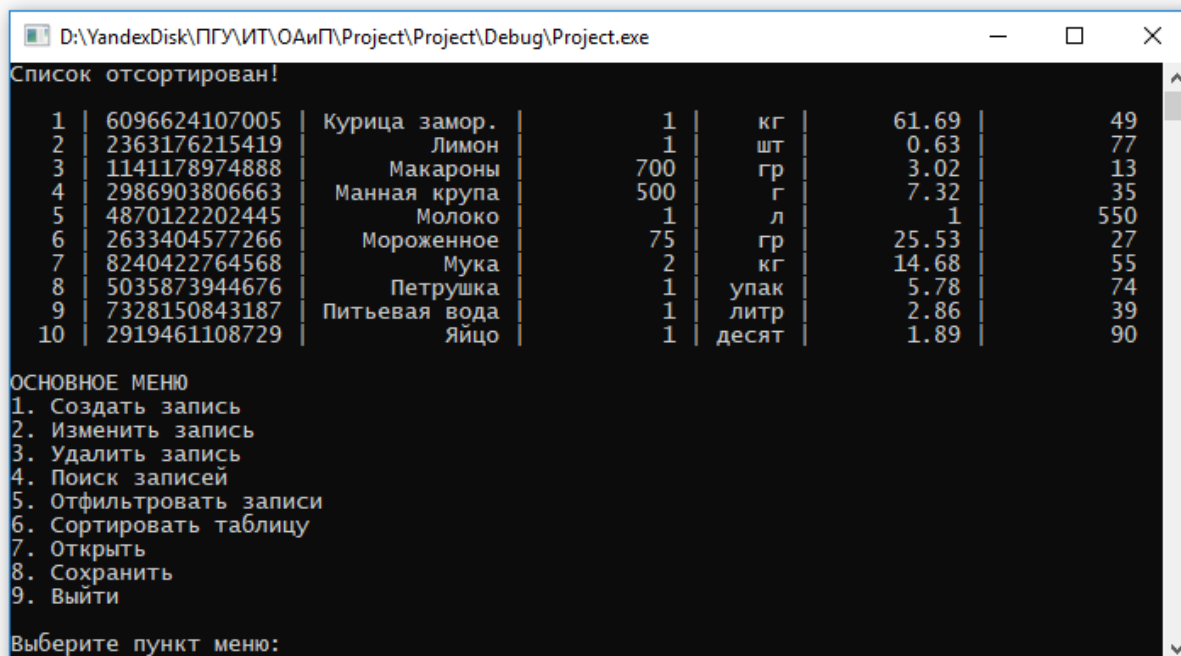


Рисунок Б.9 – Результат сортировки

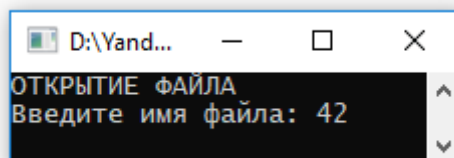


Рисунок Б.10 – Открытие файла

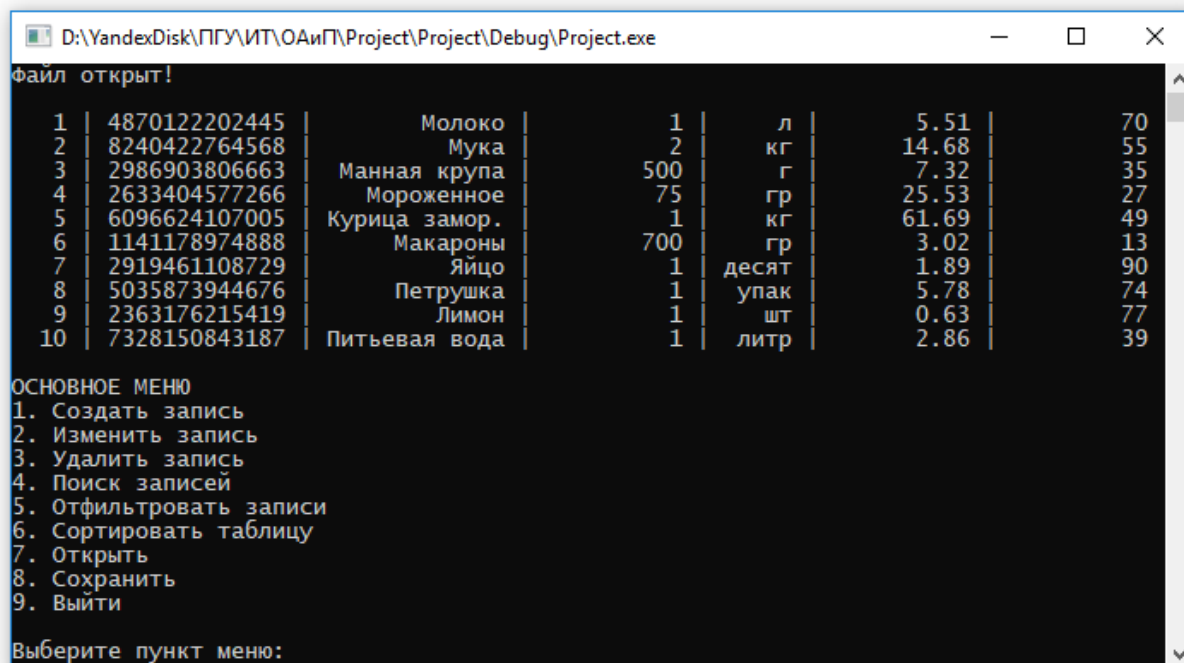


Рисунок Б.11 – Результат открытия файла

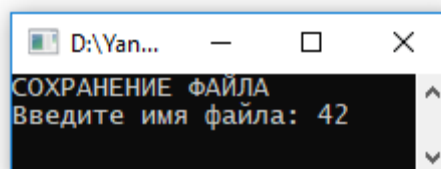


Рисунок Б.12 – Сохранение файла

Приложение В
(обязательное)
Листинг программы

Листинг программы находится на диске в файле file.docx

					БИВ.15480103.03.ПЗ	Лист
						26
Изм.	Лист	№ докум.	Подпись	Дата		