# **Содержание**

Изм

Лист

№ докум.

Подпись

Дата

Лист

2

Дубовский Н.В. 1610528

Ю.

Разраб.

Дубовский Н.В..

Провер.

Магеров В.В.

Б

Реценз.

Н. Контр.

Утверд.

«Модельный ряд Audi»

Лит.

Листов

24

35

УО «ПГУ», группа 16-ИТ-1

Содержание 2

Введение 3

[1. Анализ заданий и постановка задач 4](#_Toc468638283)

2. Теоретическая часть 6

2.1 Двусвязный список 6

2.2 Сортировка Шелла 6

2.3 Поиск 7

2.4 Редактирование 7

3. Проектирование программы 8

4. Реализация программы 10

5. Тестирование программы 14

Заключение 20

Список использованных источников 21

ПРИЛОЖЕНИЯ А 22

ПРИЛОЖЕНИЯ Б 23

ПРИЛОЖЕНИЯ В 24

введение

Изм.

Лист

№ докум.

Подпись

Дата

Лист

3

***Курсовая работа***

Тема курсовой работы – база данных «Модельного ряда Audi» реализованная на двусвязном списке.

В соответствии с предметной областью данной курсовой работой, база данных должна хранить, редактировать, сортировать, выполнять поиск по отдельно взятым записям данной темы курсовой работы.

Необходимо разработать приложение, которое могло бы обладало минимально необходимым функционалом в виде возможности хранения информации в базе данных, добавления новой информации, а также её редактирования и удаление. Также приложение должно иметь возможность вывода хранимой информации.

Для достижения поставленной цели необходимо решить следующие задачи:

* Редактирование базы данных в указанной предметной области;
* Выявить закономерности записей в проектируемой базе данных;
* Выявить различия между записями в проектируемой базе;
* Считывание и запись полученной информации в текстовый файлы.

Для создания информационной базы данных будет использоваться язык программирования С++. Для создания приложения – среда Microsoft Visual Studio 13.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

***Курсовая работа***

1. АНАЛИЗ ЗАДАНИЯ И ПОСТАНОВКА ЗАДАЧ

Основной задачей данного проекта является реализация консольного приложения, представляющего собой однофайловую базу данных, содержащую записи отдельных моделей автомобилей Audi.

К программе предъявляются следующие требования:

* Аппаратная архитектура – x86-64 совместимы компьютер на базе x64 совместимого процессора;
* Операционная система – MS Windows 7/8/8.1/10(разрядностью системы x64-бита);
* язык программирования – язык С++;
* Среда разработки - Microsoft Visual Studio 13;

Поля должны иметь следующие названия:

* Модель;
* Год;
* Двигатель;
* Кузов;
* Турбина;
* Тип двигателя;
* Диаметр колес;
* Модель Sline;
* Безопасность;
* Количество лошадиных сил;
* Количество подушек безопасности.

В консоли готового приложения должны быть реализованы следующие функции работы с записями базы данных:

Изм.

Лист

№ докум.

Подпись

Дата

Лист

4

***Курсовая работа***

1. Ввод;
2. Добавление новой записи в определенное место;
3. Вывод на экран;
4. Вывод на экран в виде таблицы;
5. Удаление определенной записи;
6. Очистка списка;
7. Сортировка;
8. Поиск;
9. Замена местами отдельно выбранных записей;
10. Редактирование записей;
11. Запись списка в файл;
12. Загрузка списка из файла;
13. Удаление файлов;
14. Работа с массивом;

Цель данной программы состоит в реализации приложение с менее оперативоемким функционалом и быстродействие.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

5

***Курсовая работа***

2. Теоретическая Часть

Изм.

Лист

№ докум.

Подпись

Дата

Лист

6

***Курсовая работа***

**2.1 Двусвязный список**

**Двусвязный список** - данные динамической структуры, представляющие собою совокупность линейно связанных однотипных элементов, для которого определены следующие фундаментальные операции:

1. Добавление элемента в любое место списка.
2. Удаление элемента из любой позиции списка.

Данный список обладает повышенной гибкостью, но также и более сложной программной реализацией, нежели простой односвязный линейный список. Гибкость достигается за счет того, что каждый элемент списка имеет два ссылочных поля: на левый узел и на правый узел, а так же указатель на начало списка и конец. В технологии обработки двусвязного списка не применяются рекурсивные алгоритмы, так как движение по элементам списка можно производить как из начала в конец, так и из конца в начало. Основная сложность связана с тем, чтобы правильно настроить взаимные связи между элементами списка, но при глубоком понимании анатомии структуры списочных данных это не создает особых сложностей.

**2.2 Сортировка Шелла**

Сортировка данного списка производиться путем выгрузки списка в массив данных(через определенную функцию), после следует сама сортировка, реализованная алгоритмом сортировки Шелла, и в последующем совершается выгрузка из массива в прежде очищенный список данных, с последующей очисткой массива данных.

Сортировка Шелла  —  [алгоритм сортировки](https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D1%81%D0%BE%D1%80%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%BA%D0%B8), являющийся усовершенствованным вариантом [сортировки вставками](https://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D1%80%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%BA%D0%B0_%D0%B2%D1%81%D1%82%D0%B0%D0%B2%D0%BA%D0%B0%D0%BC%D0%B8). Идея метода Шелла состоит в сравнении элементов, стоящих не только рядом, но и на определённом расстоянии друг от друга(с определенным шагом). Иными словами — это сортировка вставками с предварительными «грубыми» проходами.

**2.3 Поиск**

Поиск реализован через сравнение отдельно выбранного поля и введенного пользователем выражения. Так же прежде чем начинать поиск функция выгружает список в массив данных, после чего производится поиск и вывод позиции, если элемент найден, в ином случае программа нам сообщает что такого элемента в записях не найдено, после чего программа очищает ранее выгруженный массив.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

7

***Курсовая работа***

* 1. **Редактирование**

Редактирование производиться через выбор отдельной записи, прежде выведенной на экран и выгруженного списка данных в массив данных. Совершается выбор поля в выбранной пользователем записи, после происходит считывание выражения введенного пользователем и запись его в дальнейшем в определенную запись, после происходить очистка списка и выгрузка массива данных, с последующей очисткой массива данных.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

***Курсовая работа***

Изм.

Лист

№ докум.

Подпись

Дата

Лист

***Курсовая работа***

1. Проектирование программы

Изм.

Лист

№ докум.

Подпись

Дата

Лист

8

***Курсовая работа***

Основной задачей при проектировании программы является простота использования для пользователя и работоспособность программы, способность ее обнаруживать некорректный ввод данных и информировать об этом пользователя.

Главным этапом проектирования программы является реализация двусвязного списка и возможность корректной работы с ним.

В программе будет главная функция, управляющая остальными. Она будет отвечать за работу вывода меню и за запуск остальных функций согласно выбору пользователя.

Так же будет реализована функция отвечающая за ввод данных в определенных участок памяти, с последующей расстановкой указателей и преждевременной проверкой на ввод. Подобным обратом реализована функция добавления записи на любое место в списке данных.

Будут реализована функция вывода списка данных в прямом и обратным выводом данных, путем перехода от головы к концу списка и наоборот.

Однако функция вывода структуры данных в табличном виде будет реализована иным обратом. Данная функция прежде выгружает в массив данных введенный(считаный) пользователем список данных, после выполняет вывод путем перебора массива, в окончании вывода массив данных очищается.

Реализация функции удаления отдельной записи из списка, выполнена через ввод пользователем номера записи, с последующим переопределением указателей. Функция Очистки списка реализована подобным обратом. Очистка списка производиться через удаление головы списка пока в структуре данных не останется один элемент, о чем нам будет свидетельствовать глобальная переменная счетчика количества записей в структуре, после того как остается один элемент, мы обнуляем все указатели которые ведут от этого элемента, а так же голову и хвост, и очищаем память элемента.

Функция обмена местами записей списка реализована через замену метами в массиве данных, с преждевременной выгрузкой в массив и последующей очисткой памяти структуры и очисткой массива.

Также будет реализована функция, которая будет отвечать за чтения базы из файла. Она будет считывать данные и заносить их в массив с последующей выгрузкой и очисткой массива, после чего с этими данными можно будет работать: сортировать, просматривать, редактировать, удалять.

Отдельная функция будет отвечать за запись данных в файл. База, которая будет находиться в программе, может быть записана в файл, который может быть прочитан функцией чтения файла.

Также будет предусмотрен корректный выход из программы.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

3

***Курсовая работа***

Изм.

Лист

№ докум.

Подпись

Дата

Лист

9

***Курсовая работа***

1. РЕАЛИЗАЦИЯ ПРОГРАММЫ

Изм.

Лист

№ докум.

Подпись

Дата

Лист

10

***Курсовая работа***

Далее будут представлены реализации некоторых функций приложения:

Функция List\* createList() которая входит в модуль in\_out.h, в ней представлена инициализация списка данных, с использованием метода malloc:

|  |
| --- |
|  |

Рис. 1 – Инициализация списка.

Функция void file\_remove() входящая в модуль file.h, представлена с введением имени файла с расширением, проверкой на открытие файла через цикл и возможностью выходи из данной функции без совершения действий с файлами.

|  |
| --- |
|  |

Рис. 2 – Удаление файла.

Функция Del\_List(List \*list) входящая в модуль Delete.h, использует проверку на глобальную переменную n, которая ведет подсчет количества записей в списке, если же n равно нулю, то производиться выход из функции, с объявлением того что в нашем списке нет записей, после выполняется удаление всего списка с постоянным удалением головы и переопределением указателей.

|  |
| --- |
| Рис. 3 – Очистка списка. |

функция List \*Array\_save(List \*list) входящая в модуль array.h, так же выполняет проверку на пустоту списка, однако в сравнение с функцией инициализации списка(createList()) уже использует new для создания динамической памяти для массива структур, после чего в этот массив структур записываются указатели на участки памяти из нашего списка данных, после этого функция возвращает массив с данными.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

11

***Курсовая работа***

|  |
| --- |
|  |

Рис. 4 – Выгрузка в массив.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

***Курсовая работа***

Функция List \*Array\_Load(List \*list, audi \*array) входит в модуль описанный в предыдущей функции и так же выполняет работу с массивом, однако данная функция выполняет считывание элементов массива и передает его в функцию описанную нижу, после данная функция очищает память массива.

|  |
| --- |
| Рис. 5 – Загрузка из массива. |
| Функция void putDequeRight(List \*list, audi \*array) входящая в модуль array.h, создает указатели из массива в пустой список начиная с головы записывая их в структуру данных, либо добавляет все данные из массива в конец |

Рис. 6 – Создание списка из переданной памяти.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

12

***Курсовая работа***

Изм.

Лист

№ докум.

Подпись

Дата

Лист

***Курсовая работа***

Функция menu(List \*list, List \*&array) входящая в модуль menu.h, организовывает и визуализирует меню, а так же обращение и вызов функций через оператор множественного выбора(switch).

Изм.

Лист

№ докум.

Подпись

Дата

Лист

13

***Курсовая работа***

Изм.

Лист

№ докум.

Подпись

Дата

Лист

***Курсовая работа***

|  |
| --- |
|  |

Рис. 7 – Меню.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

***Курсовая работа***

**5. ТЕСТИРОВАНИЕ ПРОГРАММЫ**

|  |
| --- |
|  |

Рис. 8 – Выбор первого пункта меню(Ввод).

Изм.

Лист

№ докум.

Подпись

Дата

Лист

14

***Курсовая работа***

Изм.

Лист

№ докум.

Подпись

Дата

Лист

***Курсовая работа***

|  |
| --- |
|  |

Рис. 9 – Выбор второго пункта меню(Ввод в выбранную позицию).

|  |
| --- |
|  |

Рис. 9 – Выбор третьего пункта меню(Вывод структуры на экран).

|  |
| --- |
|  |

Рис. 9 – Выбор четвертого пункта меню(Вывод структуры в виде таблицы).

Изм.

Лист

№ докум.

Подпись

Дата

Лист

15

***Курсовая работа***

|  |
| --- |
|  |

Рис. 10 – Выбор девятого пункта меню(обмен местами записей структуры).

|  |
| --- |
|  |

Рис. 11 – Выбор седьмого пункта меню(сортировка структуры).

Изм.

Лист

№ докум.

Подпись

Дата

Лист

16

***Курсовая работа***

Изм.

Лист

№ докум.

Подпись

Дата

Лист

***Курсовая работа***

|  |
| --- |
|  |

Рис. 12 – Выбор восьмого пункта меню(поиск по структуре).

Изм.

Лист

№ докум.

Подпись

Дата

Лист

17

***Курсовая работа***

|  |
| --- |
|  |

Рис. 13 – Выбор десятого пункта меню(редактирование записи).

Изм.

Лист

№ докум.

Подпись

Дата

Лист

***Курсовая работа***

Изм.

Лист

№ докум.

Подпись

Дата

Лист

***Курсовая работа***

|  |
| --- |
|  |

Рис. 14 – Выбор пятого пункта меню(очистка списка).

|  |
| --- |
|  |

Рис. 15 – Выбор двенадцатого пункта меню(считывание из файла).

Изм.

Лист

№ докум.

Подпись

Дата

Лист

18

***Курсовая работа***

При тестировании была проведена проверка на работоспособность программы «Модельного ряда Audi». В процессе тестирования были выявлены следующие достоинства:

* В полях, которые допускается ввод только определенных значений, обрабатывают некорректный ввод и программа не завершается аварийно, а предоставляет пользователю возможность повторного ввода значения.
* Организован корректный выход из программы.
* Программа не завершалась ни одного разу при тестирование, таким обратом мы можем утверждать что данная программа является работоспособной и готова к использованию.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

19

***Курсовая работа***

**Заключение**

В результате выполненной курсовой работы была разработана и реализована база данных «Модельный ряд Audi».

Разработанная база данных удовлетворяет всем требованиям, предъявленным в задании, и позволяет без проблем хранить, редактировать и удалять требуемую информацию.

Созданное приложение позволяет упростить работу с информацией, а также её систематизировать. Также приложение упрощает операции по удалению, изменению, добавлению данных. В приложении реализованы запросы, позволяющие пользователю выбрать всю необходимую информацию по заданным критериям.

Разработанная система реагирует на ошибочный ввод данных, а так же информирует пользователя об этом.

В процессе выполнения данной курсовой работы были закреплены навыки в программировании на языке C++, проектировании баз данных и реализации функций работы с двусвязным списком.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

20

***Курсовая работа***

Изм.

Лист

№ докум.

Подпись

Дата

Лист

***Курсовая работа***

# **Список использованных источников**

* Карпов Б., Баранов Т. «С++: специальный справочник» - СПб. : Питер, 2001. — 480 с.;
* Культин Н. «С/С++ в задачах и примерах» - СПб. :БХВ-Петербург, 2002. — 288 с.;
* Павловская Т.А. “С/с++. Программирование на языке высокого уровня. - СПб.: Лидер, 2010. — 461 с.;
* http://it.kgsu.ru/PasDin/dnpas035.html;
* http://saod.narod.ru/saod3/List003.html;
* http://programm.ws/page.php?id=600;
* Культин, Н. С/С++ в задачах и примерах / Н. Культин. – СПб. : БХВ-Петербург, 2007. – 288 c.;
* Фридман, А. С/С++. Алгоритмы и приемы программирования / А. Фридман. – М. : Бином, 2007. – 560 с.;
* Шилдт, Г. Полный справочник по С. – 4-е изд. / Г. Шилдт. – М. : Изд. Дом «Вильямс», 2007. – 704 с.;
* Костюкова, Н. Язык СИ и особенности работы с ним / Н. Костюкова. – М. : Би-ном, 2006. – 207 с.;

Изм.

Лист

№ докум.

Подпись

Дата

Лист

21

***Курсовая работа***

# **Приложение А**

# **Блок-схемы алгоритмов**

|  |  |
| --- | --- |
| **Тело цикла while:http://www.kafiitbgau.narod.ru/Metod/Pascal/pascal-3.files/pascal4.jpg** | **Тело цикла for:http://gospodaretsva.com/wp-content/uploads/2011/12/Visual-Basic.NET-Urok12-1.jpg** |
| **Тело оператора множественного выбора switch:http://itandlife.ru/wp-content/uploads/2012/02/%D0%A1%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%BD%D0%B0%D1%8F-%D1%81%D1%85%D0%B5%D0%BC%D0%B0-%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%B0-switch.jpg** | **Тело условного оператора if:** |

Изм.

Лист

№ докум.

Подпись

Дата

Лист

22

***Курсовая работа***

# **Приложение б**

# **Интерфейс программы**

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

Изм.

Лист

№ докум.

Подпись

Дата

Лист

23

***Курсовая работа***

# **ПРИЛОЖЕНИЕ В**

# **Листинг программы**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

24

***Курсовая работа***

# **Приложение А**

# **Блок-схемы алгоритмов**

main

menu.h

array.h Delete.h file.h in\_out.h work.h

struct.h

Изм.

Лист

№ докум.

Подпись

Дата

Лист

***Курсовая работа***

Изм.

Лист

№ докум.

Подпись

Дата

Лист

22

***Курсовая работа***