|  |  |
| --- | --- |
| *voenmeh* | МИНОБРНАУКИ РОССИИ  федеральное государственное бюджетное образовательное учреждение  высшего образования  **«Балтийский государственный технический университет «ВОЕНМЕХ» им. Д.Ф. Устинова»**  **(БГТУ «ВОЕНМЕХ» им. Д.Ф. Устинова»)** |
| БГТУ.СМК-Ф-4.2-К5-01 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Факультет |  | И |  | Информационные и управляющие системы |
|  |  | шифр |  | наименование |
| Кафедра |  | И5 |  | Информационные системы и программная инженерия |
|  |  | шифр |  | наименование |
| Дисциплина |  | Программирование на ЯВУ | | |

КУРСОВАЯ РАБОТА

на тему

Разработка информационно-справочной системы

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Выполнил студент группы | | | |  | И494 |
| Нтирандекура Д.С. | | | | | |
| Фамилия И.О. | | | | | |
| **РУКОВОДИТЕЛЬ** | | | | | |
| Гаврютина А.А. | |  |  | | |
| Фамилия И.О. Подпись | | | | | |
| Оценка |  | | | |  |
| «\_\_\_\_\_» |  | | | | 2020г. |

САНКТ-ПЕТЕРБУРГ

2020 г.

СОДЕРЖАНИЕ

[**ВВЕДЕНИЕ 3**](#_Toc55435780)

[**1. Объединение union. 4**](#_Toc55435781)

# [**1.2. Память для хранения объединений………………………………………4**](#_Toc55435782)

# [**1.3. Общая форма объявления типа……………………………………………6**](#_Toc55435783)

[**1.4. Объявление тип объединения и переменную этого типа 6**](#_Toc55435784)

[**2 Программа**](#_Toc55435786) **8**

[**2.1 Состав программы**](#_Toc55435787) **8**

[**2.1.1 Файл Course\_Dan.cpp**](#_Toc55435788) **8**

[**2.1.2 Файл Car.h**](#_Toc55435789) **8**

**2.1.3 Файл Car.cpp………………………………………………………...……8**

**2.1.4 Файл OList.h………………………………………………………………8**

**2.1.5 Файл OList.cpp……………………………………………………………8**

[**3. Тестирование**](#_Toc55435790) **10**

[**3.1 Создание файла.**](#_Toc55435791) **10**

[**3.2 База данных загружена.**](#_Toc55435792) **10**

[**3.3 Главное меню.**](#_Toc55435793) **10**

[**3.4 Загрузить. 1**](#_Toc55435794)**1**

[**3.5 Сохранить. 1**](#_Toc55435795)**1**

[**3.6 Добавить. 11**](#_Toc55435796)

[**3.7 Удалить. 12**](#_Toc55435797)

[**3.8 Посмотреть. 12**](#_Toc55435798)

[**3.9 Редактировать. 13**](#_Toc55435799)

[**3.10 Фильтр по объему. 14**](#_Toc55435800)

[**3.11 Минимальный расход. 14**](#_Toc55435801)

[**3.12 Выход из программы. 15**](#_Toc55435802)

[**ЗАКЛЮЧЕНИЕ 16**](#_Toc55435803)

[**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 17**](#_Toc55435804)

# **ВВЕДЕНИЕ**

Цель работы – разработать информационно-справочную систему, используя знания о классах и работе с файлами.

Реферат – вариант № 6: Объединение union.

Вариант программы № 6: Поля данных: марка автомобиля, страна-производитель, год выпуска, объем двигателя, расход бензина на 100 км, цена, количество экземпляров. Скорректировать данные об определенном автомобиле при изменении на него цены. Вывести марку автомобиля с определенным объемом двигателя и наименьшим расходом бензина

В задании данные хранятся в бинарном файле записей. При выходе из программы обработанные данные сохраняются в том же файле. Для организации интерфейса в программе должно использоваться меню.

Уровень сложности - Средний. В задании список линейный односвязный. Обязательные методы класса: добавление элемента в упорядоченный список с сохранением упорядоченности (ключевое поле выбрать самостоятельно), просмотр списка, удаление произвольного элемента списка. Вывод данных осуществлять в табличном виде с графлением подходящими символами.

# **1 Объединение union**

Объединение (union) – это группирование переменных, которые разделяют одну и ту же область памяти. В зависимости от интерпретации осуществляется обращение к той или другой переменной объединения. Все переменные, что включены в объединение начинаются с одной границы.

Объединение позволяет представить в компактном виде данные, которые могут изменяться. Одни и те же данные могут быть представлены разными способами с помощью объединений. [1]

# **1.2 Память для хранения объединений**

# Размер оперативной памяти, требуемый для хранения объединений, определяется размером памяти, необходимым для размещения данных того типа, который требует максимального количества байт. Когда используется элемент меньшей длины, чем наиболее длинный элемент объединения, то этот элемент использует только часть отведенной памяти. Все элементы объединения хранятся в одной и той же области памяти, начиная с одного адреса.

# **1.3 Общая форма объявления типа.**

union имя\_типа\_объединения

{

тип переменная1;

тип переменная2;

...

тип переменнаяN;

};

где

* имя\_типа\_объединения – непосредственно имя новосозданного типа;
* переменная1, переменная2, переменнаяN – переменные, которые есть полями объединения. Эти переменные могут быть разных типов;
* тип – тип переменной, который есть полем объединения.

Тип переменной может быть:

1. [базовым типом](https://www.bestprog.net/ru/2016/08/16/03-%d0%b1%d0%b0%d0%b7%d0%be%d0%b2%d1%8b%d0%b5-%d1%82%d0%b8%d0%bf%d1%8b-%d0%b4%d0%b0%d0%bd%d0%bd%d1%8b%d1%85-visual-c/), принятым в языке C++;
2. тип [структура](https://www.bestprog.net/ru/2017/05/19/%d1%81%d1%82%d1%80%d1%83%d0%ba%d1%82%d1%83%d1%80%d1%8b-%d1%87%d0%b0%d1%81%d1%82%d1%8c-1-%d1%81%d0%be%d1%81%d1%82%d0%b0%d0%b2%d0%bd%d1%8b%d0%b5-%d1%82%d0%b8%d0%bf%d1%8b-%d0%b4%d0%b0%d0%bd%d0%bd%d1%8b/);
3. тип объединения;
4. тип класс.

# **1.4 Объявление тип объединения и переменную этого типа**

# Пусть задан тип объединения, которое содержит переменные типов с плавающей точкой

// объявление типа "объединение Floats"

union Floats

{

float f; // рассматривается 4 байта

double d; // рассматривается 8 байт

};

Тип объединения Floats содержит 2 переменные с именами f и d. Переменная f есть типа float, переменная d есть типа double. Для переменной f типа float рассматривается (принимается во внимание) 4 байта. Для переменной d типа double принимается во внимание 8 байт, так как компилятор выделяет для этого типа именно 8 байт. [2]

Поскольку размещение переменных в памяти условно начинается с одного адреса, то для переменной Fl типа Floats выделяется 8 байт памяти. Это связано с тем, что переменная типа double требует больше памяти для своего представления чем переменная типа float.

На рисунке 1 отображено размещение (интерпретация) переменных *f*, *d* из объединения Floats.



Рисунок 1 - Представление переменных f, d в объединении Floats.

# **2 Программа**

# **2.1 Состав программы**

В соответствии с индивидуальным заданием была написана программа, в состав которой вошли следующие наполнители:

* Файл Course\_Dan.cpp;
* Файл Car.h;
* Файл Car.cpp;
* Файл OList.h;
* Файл OList.cpp;

# **2.1.1 Файл Course\_Dan.cpp**

# Этот файл точка входа в программу. Реализует функционал меню и загружает файлы проекта.

# **2.1.2 Файл Car.h**

Этот файл заголовочный файл класса Car. Описывает типы полей и прототипы методов.

**2.1.3 Файл Car.cpp**

Это файл кода, описывающий реализацию полей и методов класса Car. Класс представляет совокупность характеристик автомобиля. Создан для удобной работы с базой данных.

**2.1.4 Файл OList.h**

Это заголовочный файл класса List. Описывает типы полей и прототипы методов.

**2.1.5 Файл OList.cpp**

Это файл кода, описывающий реализацию полей и методов класса List. Класс представляет программную реализацию односвязного списка.

# **3 Тестирование**

# **3.1 Создание файла**

Создание файла представлено на рисунке 1. Для загрузки или создания достаточно выбрать пункт.



Рисунок 1 – Создание файла

При нажатии клавиши «1» программа сохраняет данные в бинарный файл, чтобы в следующий раз не заполнять все заново.

# **3.2 База данных загружена**

Вводим имя нужного нам файла в бинарном формате. Представлено на рисунке 2

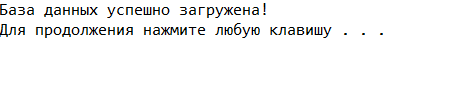


Рисунок 2 – База данных загружена

# **3.3 Главное меню**

Встреча главного меню. Представлено на рисунке 3

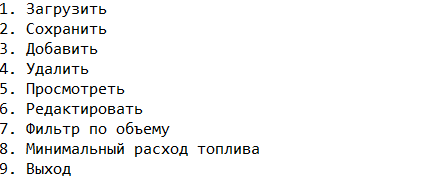


Рисунок 3 – Главное меню

# **3.4 Загрузить**

Пункт 1(Загрузить). Данная пункт нужен в случае, если пользователь при загрузке файла нажал “Нет”. Представлено на рисунке 4

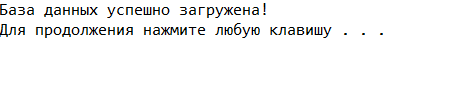


Рисунок 4 – Загрузить

# **3.5 Сохранить**

Пункт 2(Сохранить). Данная пункт нужен для сохранения возможных изменений в файл. Представлено на рисунке 5

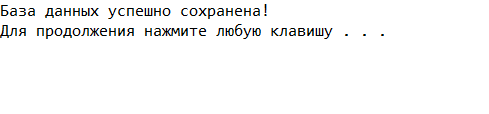


Рисунок 5 – Сохранить

# **3.6 Добавить**

Пункт 3(Добавить). Данная пункт нужен для добавления элементов в список. Представлено на рисунке 6

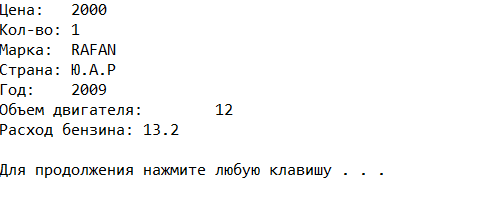


Рисунок 6 – Добавить

# **3.7 Удалить**

Пункт 4(Удалить). Данная пункт нужен для удаления элементов из списка. Ввод представлено на рисунке 7.1. Вывод представлено на рисунке 7.2.

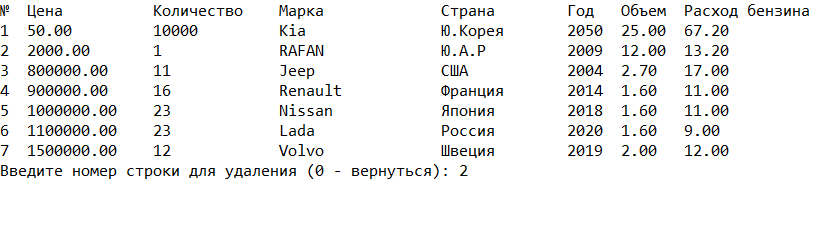


Рисунок 7.1 – Удалить

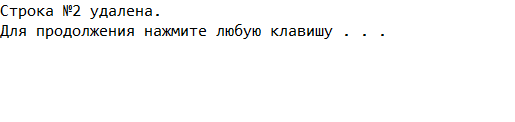


Рисунок 7.2 – Удалить

# **3.8 Посмотреть**

Пункт 5(Посмотреть). Данная пункт нужен для просмотра элементов списка. Представлено на рисунке 8.

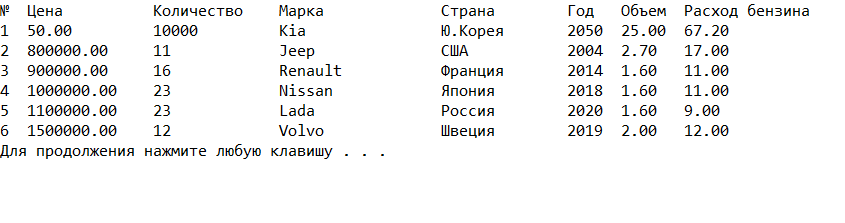


Рисунок 8 – Посмотреть

# **3.9** **Редактировать**

Пункт 6(Редактировать). Данная пункт нужен для редактирования определённого элемента списка. Ввод представлено на рисунке 9.1, вывод представлено на рисунке 9.2.

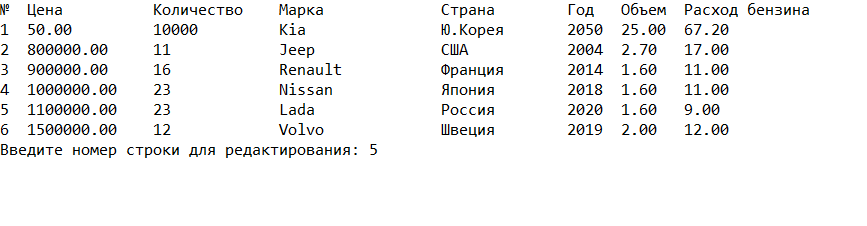


Рисунок 9.1 – Редактировать

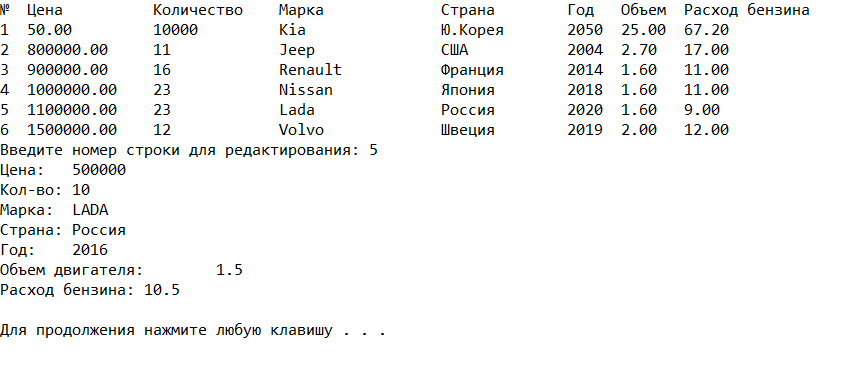


Рисунок 9.2 – Редактировать

# **3.10 Фильтр по объему**

Пункт 7(Фильтр по объему). Данная пункт нужен для поисков в элементах списка по объему ДВС автомобилей. Ввод представлено на рисунке 10.1, вывод представлено на рисунке 10.2.

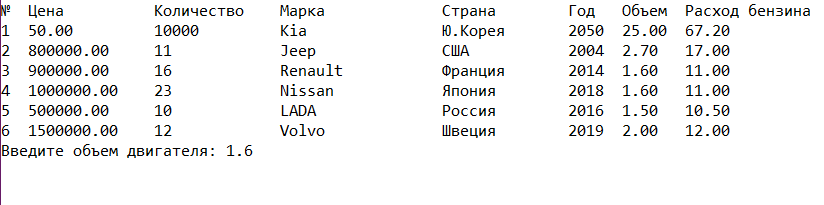


Рисунок 10.1 – Фильтр по объему

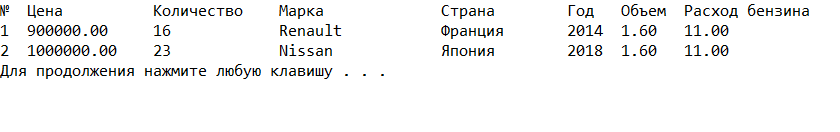


Рисунок 10.2 – Фильтр по объему

# **3.11 Минимальный расход топлива**

Пункт 8(Минимальный расход топлива). Данная пункт нужен для поиска в элементах списка по наименьшему расходу топлива автомобилей. Представлено на рисунке 11

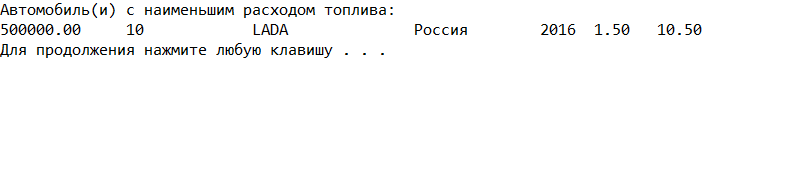


Рисунок 11 – Минимальный расход топлива

# **3.12 Выход из программы**

При нажатии клавиши «9» происходит выход из программы. Представлено на рисунке 12

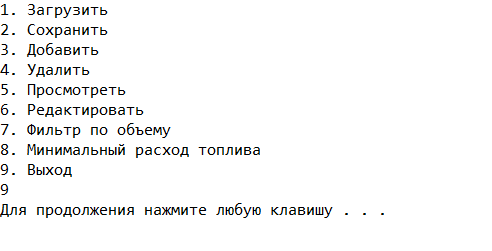


Рисунок 12 – Выход из программы

**ЗАКЛЮЧЕНИЕ**

Таким образом, была написана и протестирована программа в соответствии с индивидуальным вариантом задания, в процессе выполнения курсовой работы изучены динамические структуры данных, абстрактные типы данных,классы и их методы.

Программа выполнена на среднем уровне сложности. Содержит интерфейс и состоит из трёх файлов. В программе реализованы следующие элементы:

* Запись данных в бинарный файл.
* Просмотр списка в прямом направлении.
* Добавление элемента в список.
* Удаление элемента из списка.
* Поиск элементов по заданному ключу.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Программирование на языках C и C++ [Электронный ресурс] URL: <https://prog-cpp.ru/cpp-operator/> (дата обращения 11.11.20)
2. Программирование на языке C++. Теория [Электронный ресурс] URL: https://www.bestprog.net/ru/sitemap\_ru/c/ (дата обращения 11.11.20)
3. Павловская Т.А. С/С++. Программирование на языке высокого уровня. СПб.: «Питер», 2017.