**­­МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В. И. Ульянова (Ленина)**

**Кафедра Вычислительной техники**

**Курсовая работа**

**по дисциплине «Программирование»**

**Тема: Разработка электронной картотеки.**

|  |  |  |
| --- | --- | --- |
| Студент гр. 2305 |  | Решетняк А.К. |
| Преподаватель |  | Хахаев И.А. |

Санкт-Петербург

2023

**Содержание**

[**Введение.** 3](#_Toc137124648)

[**Задание.** 3](#_Toc137124649)

[**Описание общей архитектуры данных.** 4](#_Toc137124650)

[**Связи между функциями** 5](#_Toc137124651)

[**Описание структур.** 6](#_Toc137124652)

[**Описание функций** 8](#_Toc137124653)

[**Описание переменных (для каждой функции).** 11](#_Toc137124654)

[**Пример выполнения программы.** 29](#_Toc137124655)

[**Заключение.** 32](#_Toc137124656)

**Введение.**

Цель работы заключается в изучении языка программирования C и получение практических навыков программирования для создания электронной картотеки.

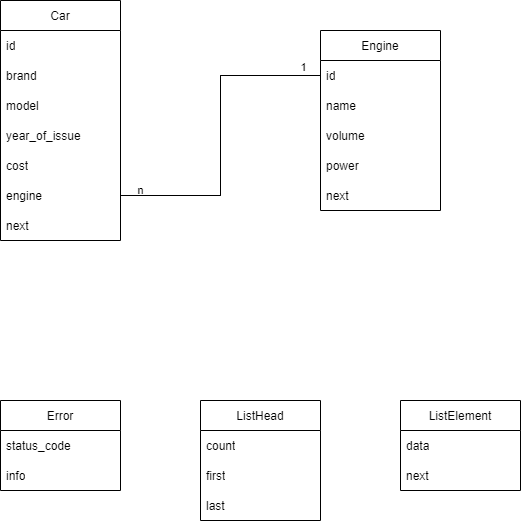
Задача работы представляет собой создание электронной картотеки автомобилей, содержащей таблицу с данными об автомобилях и таблицу с данными о двигателях

**Задание.**

Программа должна выполнять следующие действия:

* Выводить справку (вывод на экран краткой документации по пунктам меню).
* Добавлять новые карточки (внесения нового элемента).
* Редактирование карточки (изменение элемента, находящегося в списке)
* Удаление карточки (удаление элемента).
* Вывод картотеки (вывод на экран списка).
* Поиск карточки по параметру (поиск по введённым данным в заданном пользователем поле).
* Сортировка картотеки по параметру (сортировка по заданному пользователем полю)
* Выход (сохранение изменённой картотеки и завершение работы программы).

**Описание общей архитектуры данных.**



Структура Car – содержит данные об автомобилях, а также указатель на список двигателей

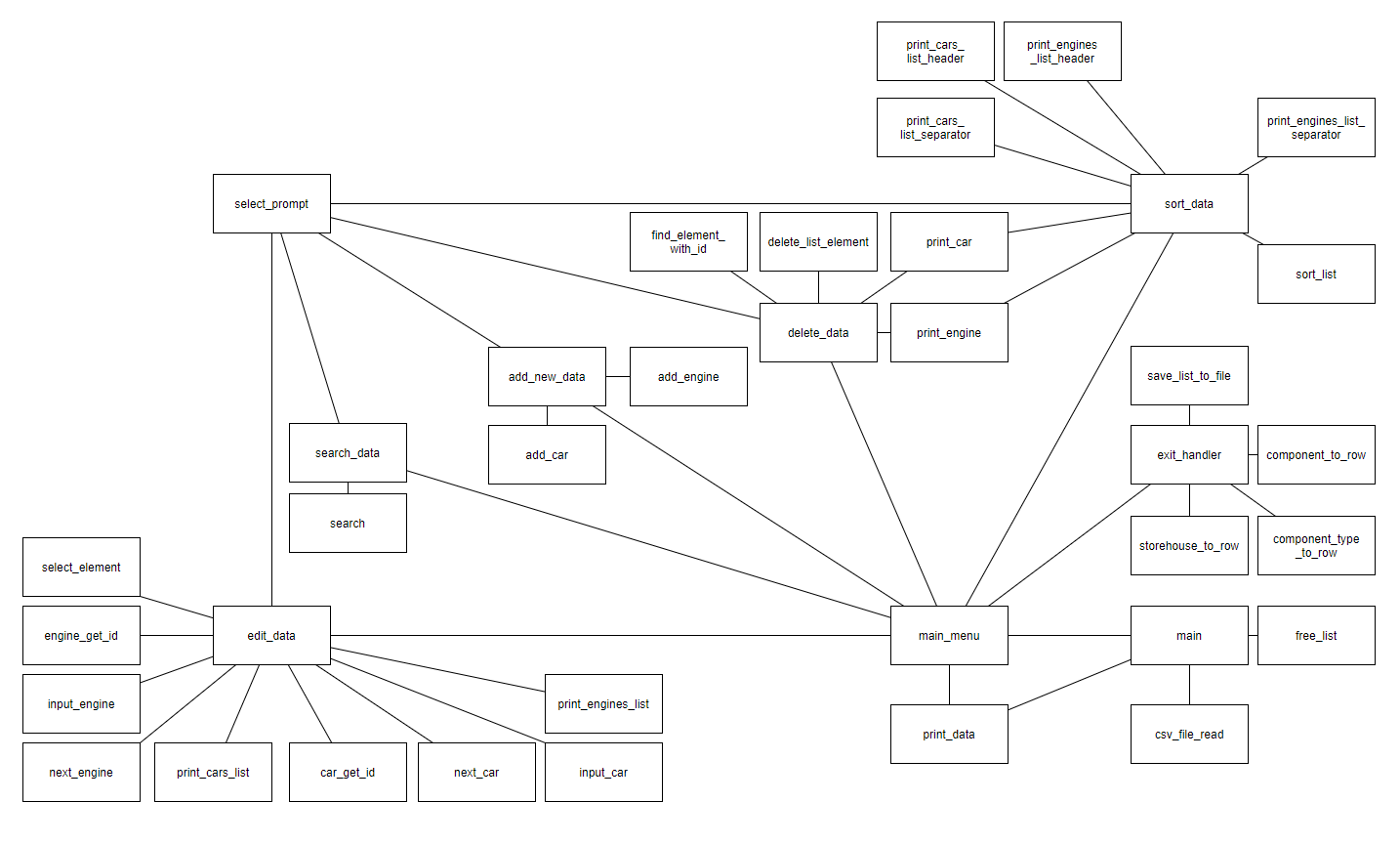
Структура Engine – содержит данные о двигателях

Структура Error – содержит код ошибки и информацию об ошибке

Структура ListHead – содержит количество элементов, а также указатели на первый и последний элементы

Структура ListElement – содержит данные об элементе и указатель на следующий элемент

**Связи между функциями**



**Описание структур.**

Структура error

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | status\_code | StatusCode | Код ошибки |
| 2 | info | char\* | Информация об ошибке |

Структура Engine

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | id | unsigned | Номер id |
| 2 | name | char\* | Название двигателя |
| 3 | volume | float | Объем двигателя |
| 4 | power | int | Мощность двигателя |
| 5 | next | struct Engine\* | Указатель на следующий элемент |

Структура Car

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | id | unsigned | Номер id |
| 2 | brand | char\* | Марка |
| 3 | model | char\* | Модель |
| 4 | year\_of\_issue | int | Год производства |
| 5 | cost | float | Цена |
| 6 | engine | Engine\* | Двигатель |
| 7 | next | struct Car\* | Указатель на следующий элемент |

Структура ListHead

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | count | size\_t | Количество внесённых элементов |
| 2 | first | void\* | Указатель на первый элемент в списке |
| 3 | last | void\* | Указатель на последний элемент в списке |

Структура ListElement

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | data | void\* | Данные |
| 2 | next | void\* | Указатель на следующий элемент в списке |

**Описание функций**

**-** Error main\_menu(ListHead \*cars, ListHead \*engines)– главное меню

utils.h:

-FILE \*open\_or\_create\_file(char \*filename) – открытие файла или создание нового если файла нет

**-** Error csv\_file\_read(char \*filename, ListHead \*output, size\_t element\_size,

void \*\*(\*next\_getter)(void \*), void (\*free\_func)(void \*\*), Error (\*row\_converter)(CSVFileLine, void \*output, void \*), void \*additional\_args)– считывание файла и перевод информации в список

**-**void remove\_last\_symbol(char \*str) – удаление последнего символа строки

**-**void flush\_stdin() – удаление оставшихся символов переноса строки

structs\_to\_row.h:

**-** Error car\_to\_row(void \*car, CSVFileLine \*row) – перевод автомобиля в строку файла

**-** Error engine\_to\_row(void \*engine, CSVFileLine \*row) – перевод двигателя в строку файла

row\_to\_struct.h:

**-** Error row\_to\_engine(CSVFileLine row, void \*engine, ...) – перевод строки файла в структуру двигателя

**-** Error row\_to\_car(CSVFileLine row, void \*car, void \*additional\_args) – перевод строки файла в структуру автомобиля

printers.h:

- void print\_car(const void \*car) – вывод автомобиля

- void print\_cars\_list\_separator() – вывод разделителя таблицы

- void print\_cars\_list\_header() – вывод заголовка таблицы

- void print\_cars\_list(const ListHead cars) – вывод списка автомобилей

- void print\_engine(const void \*engine) – вывод двигателя

-void print\_engines\_list\_separator() – вывод разделителя таблицы

-void print\_engines\_list\_header() – вывод заголовка таблицы

-void print\_engines\_list(const ListHead engines) – вывод списка двигателей

menu\_functions.h:

-void about() – вывод меню помощи

- Error add\_new\_data(ListHead \*cars, ListHead \*engines) – добавление данных в картотеку

- Error edit\_data(ListHead cars, ListHead engines) – редактирует данные в картотеке

- Error delete\_data(ListHead \*cars, ListHead \*engines)– удаление данных из картотеки

- void print\_data(const ListHead cars, const ListHead engines) – вывод картотеки

- Error search\_data(ListHead cars, ListHead engines) – поиск данных в картотеке

- Error sort\_data(ListHead cars, ListHead engines) – сортировка данных картотеки

- Error exit\_handler(ListHead cars, ListHead engines) – выход из программы и сохранение данных в картотеку по желанию пользователя

-int select\_prompt(char \*prompt, int min\_value, int max\_value) – выбор пункта в меню

getters.h:

- unsigned \*car\_get\_engine\_id(void \*car) – получение id двигателя автомобиля

- char \*\*car\_get\_engine\_name(void \*car) – получение названия двигателя автомобиля

- float \*car\_get\_cost(void \*car) – получение цены автомобиля

- char \*\*car\_get\_model(void \*car) – получение модели автомобиля

- unsigned \*car\_get\_id(void \*car) – получение id автомобиля

- char \*\*car\_get\_brand(void \*car) – получение марки автомобиля

- int \*car\_get\_year\_of\_issue(void \*car) – получение года выпуска автомобиля

- void \*\*next\_car(void \*car) – переход к следующему автомобилю

- unsigned \*engine\_get\_id(void \*engine) – получение id двигателя

- char \*\*engine\_get\_name(void \*engine) – получение названия двигателя

- float \*engine\_get\_volume(void \*engine) – получение объема двигателя

- int \*engine\_get\_power(void \*engine) – получение мощности двигателя

- void \*\*next\_engine(void \*engine) – переход к следующему двигателю

- void \*\*list\_element\_get\_data(void \*list\_element) – получение данных элемента

- void \*\*next\_list\_element(void \*list\_element) – переход к следующему элементу

csv\_file\_to\_list.h:

-Error csv\_file\_to\_list(CSVFile csv, ListHead \*output, size\_t element\_size, void \*\*(\*next\_getter)(void \*), void (\*free\_func)(void \*\*), Error (\*row\_converter)(CSVFileLine, void \*output, void \*), void \*additional\_args) – перевод файла в структуру

csv\_file.h:

-void CSVFileFree(CSVFile \*file) – освобождение памяти файла

-int CSVFileRead(FILE \*fp, CSVFile \*output) – чтение файла

-void CSVFileWrite(FILE \*fp, CSVFile data) – запись в файл

-void \_CSVFileFreeLine(CSVFileLine \*line, size\_t fields\_count) – освобождение памяти строки файла

-int \_CSVFileAddCharToOutput(char character, char \*\*output, size\_t \*string\_len, size\_t \*allocated) – добавление символа в конец строки

-int \_CSVFileReadField(FILE \*csv\_file, char \*\*output) – чтение поля из файла

-int \_CSVFileCountFields(FILE \*csv\_file, size\_t \*fields\_count) – подсчет количества полей в файле

-int \_CSVFileSkipCRLF(FILE \*csv\_file) – пропуск окончания строки

-int \_CSVReadLine(FILE \*csv\_file, CSVFileLine \*output, size\_t fields\_count) – чтение строки файла

add\_funcs.h:

- Error add\_car(ListHead \*cars, const ListHead engines) – добавление автомобиля

- Error add\_engine(ListHead \*engines) – добавление двигателя

edit\_funcs.h:

-void \*select\_element(const ListHead list, unsigned \*(\*id\_getter)(void \*), void \*\*(next\_getter)(void \*), void (\*printer)(const void \*)) – выбор элемента

exit\_funcs.h:

-Error save\_list\_to\_file(ListHead list, char \*output\_filename, size\_t element\_fields\_count, Error (\*element\_to\_row)(void \*element, CSVFileLine \*row), void \*\*(\*next\_getter)(void \*element)) – сохранение картотеки в файл

input\_funcs.h:

- Car \*input\_car(const ListHead engines) – ввод автомобиля

- Engine \*input\_engine() – ввод двигателя

search\_funcs.h:

-char search(const ListHead list, void \*field\_value, size\_t field\_size, void \*(\*field\_getter)(void \*), void \*\*(\*next\_getter)(void \*), void (\*printer)(const void \*), char is\_str) – поиск

sort\_funcs.h:

-Error sort\_list(ListHead to\_sort, ListHead \*output, void \*(\*field\_getter)(void \*), void \*\*(\*next\_getter)(void \*), VariableType type) – сортировка списка

-int compare(void \*value1, void \*value2, VariableType type) – сравнение элементов

utils.h:

-void \*find\_element\_with\_id(ListHead list, unsigned id, unsigned \*(\*id\_getter)(void \*), void \*\*(\*next\_getter)(void \*)) – поиск элемента с заданным id

-void delete\_list\_element(ListHead \*list, void \*element, void \*\*(\*next\_getter)(void \*), void (\*free\_func)(void \*\*)) – удаление элемента

csv\_row\_converters.h:

-Error scan\_row(CSVFileLine row, const char \*format, ...) – считывание строки файла

-Error create\_row(CSVFileLine \*row, const char \*format, ...) – запись строки в файл

utils.h:

-Error format\_str\_to\_enum\_list(const char \*format, int \*\*output, size\_t \*count) – конвертация строки спецификаторов в массив, состоящий из элементов перечисления этих спецификаторов

-size\_t specifier\_to\_size(int int\_specifier) – конвертация элемента перечисления спецификаторов в размер требуемой памяти для хранения спецификатора такого типа

-int specifier\_to\_enum(char specifier[3]) – конвертация одного спецификатора в один элемент перечисления спефикаторов

**Описание переменных (для каждой функции).**

main

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | error | Error | Код ошибки |
| 2 | engines | ListHead | Список двигателей |
| 3 | cars | ListHead | Список автомобилей |

main\_menu

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | cars | ListHead\* | Указатель на список автомобилей |
| 2 | engines | ListHead\* | Указатель на список двигателей |
| 3 | error | Error | Код ошибки |
| 4 | menu\_selection | MenuSelection | Выбор пункта меню |

add\_new\_data

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | cars | ListHead\* | Указатель на список автомобилей |
| 2 | engines | ListHead\* | Указатель на список двигателей |
| 3 | error | Error | Код ошибки |
| 4 | table\_selection | int | Выбор таблицы |

edit\_data

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | cars | ListHead\* | Указатель на список автомобилей |
| 2 | engines | ListHead\* | Указатель на список двигателей |
| 3 | error | Error | Код ошибки |
| 4 | table\_selection | int | Выбор таблицы |
| 5 | element | void\* | Элемент |
| 6 | replacement | void\* | Замена элемента |

delete\_data

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | cars | ListHead\* | Указатель на список автомобилей |
| 2 | engines | ListHead\* | Указатель на список двигателей |
| 3 | error | Error | Код ошибки |
| 4 | id | unsigned | Номер id |
| 5 | iter | void\* | Вспомогательный элемент |
| 6 | found | char | Флаг нахождения |
| 7 | table\_selection | int | Выбор таблицы |

print\_data

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | cars | ListHead | Список компонентов |
| 2 | engines | ListHead | Список типов компонентов |

search\_data

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | cars | ListHead\* | Указатель на список автомобилей |
| 2 | engines | ListHead\* | Указатель на список двигателей |
| 3 | error | Error | Код ошибки |
| 4 | car\_field\_getters[] | void\* | Получение данных с полей структуры |
| 5 | engine\_field\_getters[] | void\* | Получение данных с полей структуры |
| 7 | search\_criteria | int | Критерий поиска |
| 8 | is\_str | char | Флаг является ли элемент строкой |
| 9 | found | char | Флаг нахождения |
| 10 | value\_size | size\_t | Размер типа |
| 11 | buf[1024] | char | Буфер |
| 12 | table\_selection | int | Выбор таблицы |

sort\_data

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | cars | ListHead\* | Указатель на список автомобилей |
| 2 | engines | ListHead\* | Указатель на список двигателей |
| 3 | error | Error | Код ошибки |
| 4 | table\_selection | int | Выбор таблицы |
| 5 | sort\_criteria | int | Критерий поиска |
| 6 | sort\_type | VariableType | Тип данных для поиска |
| 7 | sorted | ListHead | Отсортированный список |
| 8 | element | ListElement\* | Элемент |
| 9 | car\_field\_getters[] | void\* | Получение данных с полей структуры |
| 10 | engine\_field\_getters[] | void\* | Получение данных с полей структуры |
| 11 | printer() | void\* | Вывод структуры |
| 12 | separator\_printer() | void\* | Вывод разделителей таблицы |
| 13 | header\_printer() | void\* | Вывод заголовка таблицы |

exit\_handler

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | cars | ListHead | Список автомобилей |
| 2 | engines | ListHead | Список двигателей |
| 3 | error | Error | Код ошибки |
| 4 | confirm | char | Подтверждение пользователем |

select\_prompt

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | prompt | char\* | Ответ пользователя |
| 2 | min\_value | int | Минимальное значение |
| 3 | max\_value | int | Максимальное значение |
| 4 | table\_selection | int | Выбор таблицы |

print\_car

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | car | void\* | Указатель на список автомобилей |
| 2 | car\_struct | Car\* | Переменная для вывода элементов списка |

print\_cars\_list

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | cars | ListHead | Список автомобилей |
| 2 | car | Car\* | Переменная для вывода списка |

print\_engine

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | engine | void\* | Указатель на список двигателей |
| 2 | engine\_struct | Engine\* | Переменная для вывода элементов списка |

print\_engines\_list

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | engines | ListHead | Список двигателей |
| 2 | engine | Engine\* | Переменная для вывода списка |

add\_car

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | cars | ListHead\* | Указатель на список автомобилей |
| 2 | engines | ListHead\* | Указатель на список двигателей |
| 3 | error | Error | Код ошибки |
| 4 | new\_car | Car\* | Новый автомобиль |

add\_engine

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | engines | ListHead\* | Указатель на список двигателей |
| 2 | error | Error | Код ошибки |
| 3 | new\_engine | ComponentType\* | Новый двигатель |

select\_element

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | list | ListHead | Список элементов |
| 2 | (\*id\_getter)(void\*) | unsigned\* | Получение id |
| 3 | (next\_getter)(void\*) | void\*\* | Получение следующего элемента |
| 4 | (\*printer)(const void\*) | void | Вывод элемента |
| 5 | confirm | char | Подтверждение пользователем |
| 6 | element | void\* | Элемент |
| 7 | id | unsigned | Номер id |

save\_list\_to\_file

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | list | ListHead | Список элементов |
| 2 | output\_filename | char\* | Имя выходного файла |
| 3 | element\_fields\_count | size\_t | Количество полей элементов |
| 4 | (\*element\_to\_row)(void \*element, CSVFileLine \*row) | Error | Преобразование элемента в строку файла |
| 5 | error | Error | Код ошибки |
| 6 | (\*next\_getter)(void \*element) | void\*\* | Получение следующего элемента |
| 7 | element | void\* | Элемент |
| 8 | csv | CSVFile | Файл |
| 9 | output | FILE\* | Выходной файл |
| 10 | i | size\_t | Переменная для цикла for |

input\_car

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | car | Car\* | Указатель на автомобиль |
| 2 | engines | ListHead | Список двигателей |

input\_engine

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | engine | Engine\* | Указатель на список двигателей |

search

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | list | ListHead | Список элементов |
| 2 | field\_value | void\* | Значение поля |
| 3 | field\_size | size\_t | Размер поля |
| 4 | (\*printer)(const void\*) | void | Вывод элемента |
| 5 | (\*next\_getter)(void \*) | void\*\* | Получение следующего элемента |
| 6 | (\*field\_getter)(void\*) | void\* | Получение поля |
| 7 | is\_str | char | Флаг является ли строкой |
| 8 | iter | void\* | Вспомогательная переменная |
| 9 | found | char | Флаг нахождения |

sort\_list

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | to\_sort | ListHead | Список для сортировки |
| 2 | output | ListHead\* | Указатель на выходной список |
| 3 | type | VariableType | Тип переменной |
| 4 | error | Error | Код ошибки |
| 5 | (\*next\_getter)(void \*) | void\*\* | Получение следующего элемента |
| 6 | (\*field\_getter)(void\*) | void\* | Получение поля |
| 7 | element | void\* | Элемент |
| 8 | element2 | void\* | Элемент 2 |
| 9 | temp | void\* | Временная переменная |

compare

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | value1 | void\* | Значение 1 |
| 2 | value2 | void\* | Значение 2 |
| 3 | type | VariableType | Тип переменной |
| 4 | result | int | Результат |

find\_element\_with\_id

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | list | ListHead | Список элементов |
| 2 | id | unsigned | Номер id |
| 3 | (\*id\_getter)(void\*) | unsigned\* | Получение id |
| 4 | (\*next\_getter)(void \*) | void\*\* | Получение следующего элемента |
| 5 | element | void\* | Элемент |
| 6 | found | char | Флаг нахождения |

delete\_list\_element

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | list | ListHead\* | Список элементов |
| 2 | element | void\* | Элемент |
| 3 | (\*free\_func)(void\*\*) | void | Освобождение функции |
| 4 | (\*next\_getter)(void \*) | void\*\* | Получение следующего элемента |
| 5 | iter | void\* | Вспомогательная переменная |
| 6 | prev | void\* | Предыдущий элемент |

car\_get\_engine\_id

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | car | void\* | Указатель на список автомобилей |

car\_get\_engine\_name

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | car | void\* | Указатель на список автомобилей |

car\_get\_cost

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | car | void\* | Указатель на список автомобилей |

car\_get\_model

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | car | void\* | Указатель на список автомобилей |

car\_get\_id

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | car | void\* | Указатель на список автомобилей |

car\_get\_brand

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | car | void\* | Указатель на список автомобилей |

car\_get\_year\_of\_issue

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | car | void\* | Указатель на список автомобилей |

next\_car

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | car | void\* | Указатель на список автомобилей |

engine\_get\_id

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | engine | void\* | Указатель на список двигателей |

engine\_get\_name

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | engine | void\* | Указатель на список двигателей |

engine\_get\_volume

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | engine | void\* | Указатель на список двигателей |

engine\_get\_power

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | engine | void\* | Указатель на список двигателей |

next\_engine

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | engine | void\* | Указатель на список двигателей |

list\_element\_get\_data

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | list\_element | void\* | Указатель на список элементов |

next\_list\_element

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | list\_element | void\* | Указатель на список элементов |

car\_to\_row

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | car | void\* | Указатель на автомобиль |
| 2 | row | CSVFileLine\* | Строка файла |
| 3 | car\_struct | Car\* | Указатель на список автомобилей |

engine\_to\_row

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | engine | void\* | Указатель на двигатель |
| 2 | row | CSVFileLine\* | Строка файла |
| 3 | engine\_struct | Engine\* | Указатель на список двигателей |

flush\_stdin

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | c | char | Получение символа |

remove\_last\_symbol

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | str | char\* | Строка |
| 2 | len | size\_t | Длина строки |

csv\_file\_read

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | filename | char\* | Имя файла |
| 2 | output | ListHead\* | Выходной список |
| 3 | element\_size | size\_t | Размер элемента |
| 4 | (\*next\_getter)(void\*) | void\*\* | Получение следующего элемента |
| 5 | (\*free\_func)(void\*\*) | void | Освобождение функции |
| 6 | (\*row\_converter)(CSVFileLine,  void\*, va\_list) | Error | Перевод строк файла |
| 7 | f | FILE\* | Файл |
| 8 | error | Error | Код ошибки |
| 9 | csv | CSVFile | CSV-файл |
| 10 | csv\_status | int | Статус файла |
| 11 | additional\_args | va\_list | Дополнительные аргументы |

open\_or\_create\_file

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | filename | char\* | Имя файла |
| 2 | error | StatusCode | Код ошибки |
| 3 | f | FILE\* | Файл |

free\_car

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | car | void\*\* | Указатель на автомобиль |
| 2 | car\_struct | Car\* | Указатель на список автомобилей |

free\_engine

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | engine | void\*\* | Указатель на двигатель |
| 2 | engine\_struct | Engine\* | Указатель на список двигателей |

free\_list\_element

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | list\_element | void\*\* | Указатель на элемент списка |

free\_list

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | list | ListHead\* | Указатель на список элементов |
| 2 | (\*free\_func)(void\*\*) | void | Освобождение функции |
| 3 | (\*next\_func)(void\*) | void\*\* | Переход к следующей функции |

row\_to\_engine

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | row | CSVFileLine | Строка файла |
| 2 | engine | void\* | Указатель на двигатель |
| 3 | pseudo | Engine\* | Указатель на вспомогательный двигатель |

row\_to\_car

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | row | CSVFileLine | Строка файла |
| 2 | car | void\* | Указатель на автомобиль |
| 3 | additional\_args | void\* | Дополнительные аргументы |
| 4 | engines | ListHead | Список двигателей |
| 5 | engine\_id | unsigned | Номер id двигателя |
| 6 | error | Error | Код ошибки |
| 7 | engine | Engine\* | Указатель на двигатель |
| 8 | pseudo | Car\* | Указатель на вспомогательный автомобиль |

csv\_file\_to\_list

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | csv | CSVFile | Файл |
| 2 | output | ListHead\* | Указатель на выходной список |
| 3 | element\_size | size\_t | Размер элемента |
| 4 | (\*next\_getter)(void \*) | void\*\* | Получение следующего элемента |
| 5 | (\*free\_func)(void \*\*) | void | Освобождение функции |
| 6 | (\*row\_converter)(CSVFileLine, void \*output, void \* | Error | Преобразование строк файла |
| 7 | additional\_args | void\* | Дополнительные аргументы |
| 8 | error | Error | Код ошибки |
| 9 | i | size\_t | Переменная для цикла for |

CSVFileFree

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | file | CSVFile\* | Файл |
| 2 | i | size\_t | Переменная для цикла for |

CSVFileRead

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | output | CSVFile\* | Указатель на выходной файл |
| 2 | fp | FILE\* | Файл |
| 3 | error | int | Код ошибки |
| 4 | continue\_loop | int | Флаг продолжения |
| 5 | allocated | size\_t | Переменная для выделения памяти |
| 6 | realloc\_result | CSVFileLine\* | Результат |
| 7 | next\_char | int | Следующий символ |

CSVFileWrite

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | data | CSVFile | Файл данных |
| 2 | fp | FILE\* | Файл |
| 3 | i | size\_t | Переменная для цикла for |
| 4 | j | size\_t | Переменная для цикла for |
| 5 | CRLF | char | Переменная для записи в файл |

\_CSVFileFreeLine

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | line | CSVFileLine\* | Строка файла |
| 2 | fields\_count | size\_t | Количество полей |
| 3 | i | size\_t | Переменная для цикла for |

\_CSVFileAddCharToOutput

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | character | char | Символ |
| 2 | output | char\*\* | Выходная строка |
| 3 | string\_len | size\_t\* | Длина строки |
| 4 | allocated | size\_t\* | Переменная для выделения памяти |
| 5 | error | int | Код ошибки |
| 6 | realloc\_result | char\* | Результат |

\_CSVFileReadField

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | csv\_file | FILE\* | Файл |
| 2 | output | char\*\* | Выходная строка |
| 3 | string\_len | size\_t | Длина строки |
| 4 | allocated | size\_t | Переменная для выделения памяти |
| 5 | i | size\_t | Переменная для цикла for |
| 6 | quoted | char | Флаг, показывающий кавычки |
| 7 | continue\_loop | char | Флаг продолжения |
| 8 | realloc\_result | char\* | Результат |
| 9 | current\_char | int | Текущий символ |
| 10 | error | int | Код ошибки |

\_CSVFileCountFields

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | csv\_file | FILE\* | Файл |
| 2 | fields\_count | size\_t\* | Количество полей |
| 3 | start | long | Старт |
| 4 | current\_char | int | Текущий символ |
| 5 | previous\_char | int | Предыдущий символ |
| 6 | quoted | char | Флаг, показывающий кавычки |
| 7 | continue\_loop | char | Флаг продолжения |
| 8 | error | int | Код ошибки |
| 9 | quotes\_count | size\_t | Количество кавычек |

\_CSVFileSkipCRLF

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | csv\_file | FILE\* | Файл |
| 2 | error | int | Код ошибки |
| 3 | next | int | Переход к следующему символу |

\_CSVReadLine

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | csv\_file | FILE\* | Файл |
| 2 | output | CSVFileLine\* | Выходная строка |
| 3 | fields\_count | size\_t | Количество полей |
| 4 | error | int | Код ошибки |
| 5 | i | size\_t | Переменная для цикла for |
| 6 | next\_char | int | Следующий символ |
| 7 | continue\_loop | char | Флаг продолжения |

scan\_row

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | row | CSVFileLine | Строка файла |
| 2 | format | char\* | Формат |
| 3 | specifiers | COURSE\_TASK\_  SPECIFIER\_TYPE\* | Спецификаторы |
| 4 | count | size\_t | Количество |
| 5 | i | size\_t | Переменная для цикла for |
| 6 | error | Error | Код ошибки |
| 7 | args | va\_list | Аргументы |
| 8 | str | char\*\* | Строка |
| 9 | temp | int | Временная переменная |

create\_row

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | row | CSVFileLine\* | Строка файла |
| 2 | format | char\* | Формат |
| 3 | specifiers | COURSE\_TASK\_  SPECIFIER\_TYPE\* | Спецификаторы |
| 4 | count | size\_t | Количество |
| 5 | i | size\_t | Переменная для цикла for |
| 6 | error | Error | Код ошибки |
| 7 | length | size\_t | Длина |
| 8 | args | va\_list | Аргументы |
| 9 | str | char\* | Строка |
| 10 | temp | int | Временная переменная |

format\_str\_to\_enum\_list

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | output | int\*\* | Выходное значение |
| 2 | format | char\* | Формат |
| 3 | buf | char[3] | Буфер |
| 4 | count | size\_t\* | Количество |
| 5 | i | size\_t | Переменная для цикла for |
| 6 | error | Error | Код ошибки |
| 7 | allocated | size\_t | Переменная для выделения памяти |
| 8 | j | size\_t | Переменная для цикла for |
| 9 | temp\_specifiers\_ptr | int\* | Временные спецификаторы |
| 10 | temp | int | Временная переменная |

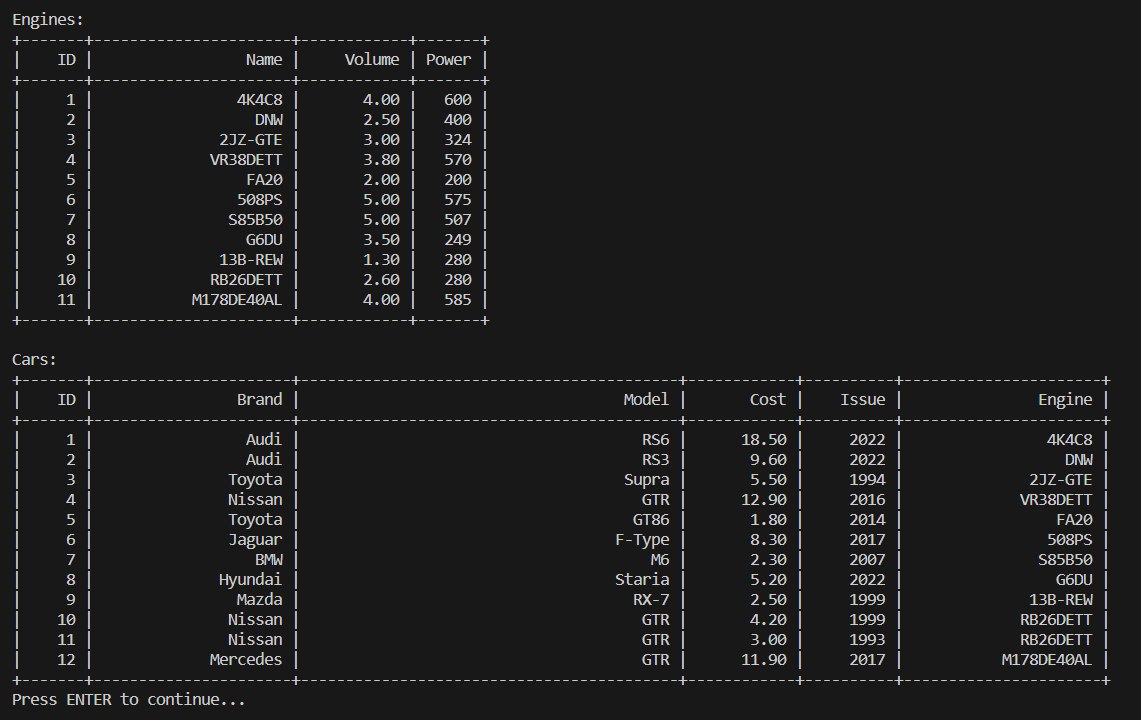
specifier\_to\_size

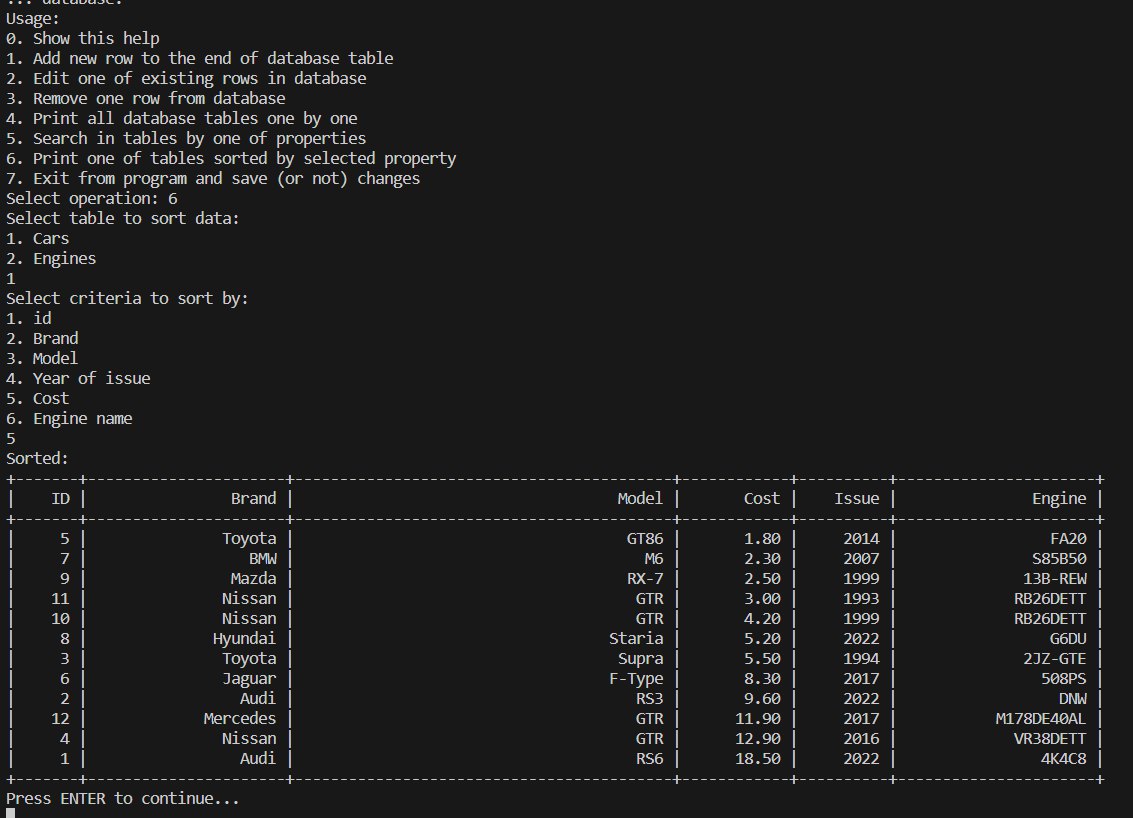
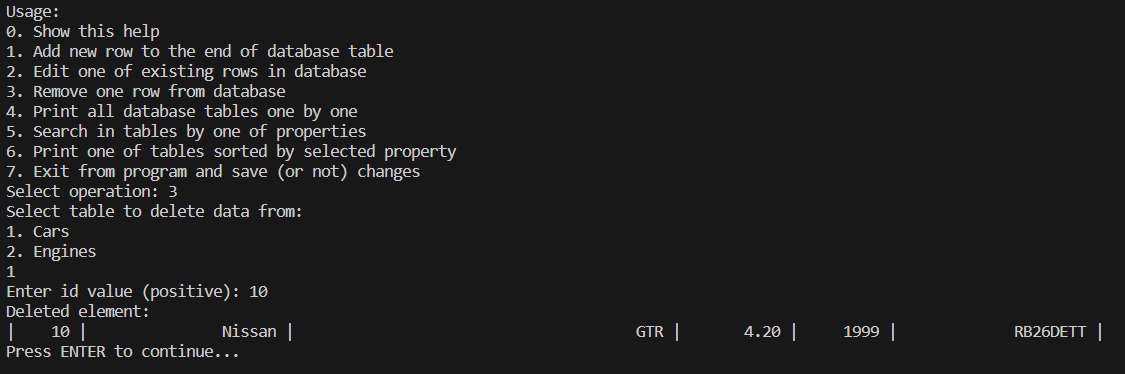
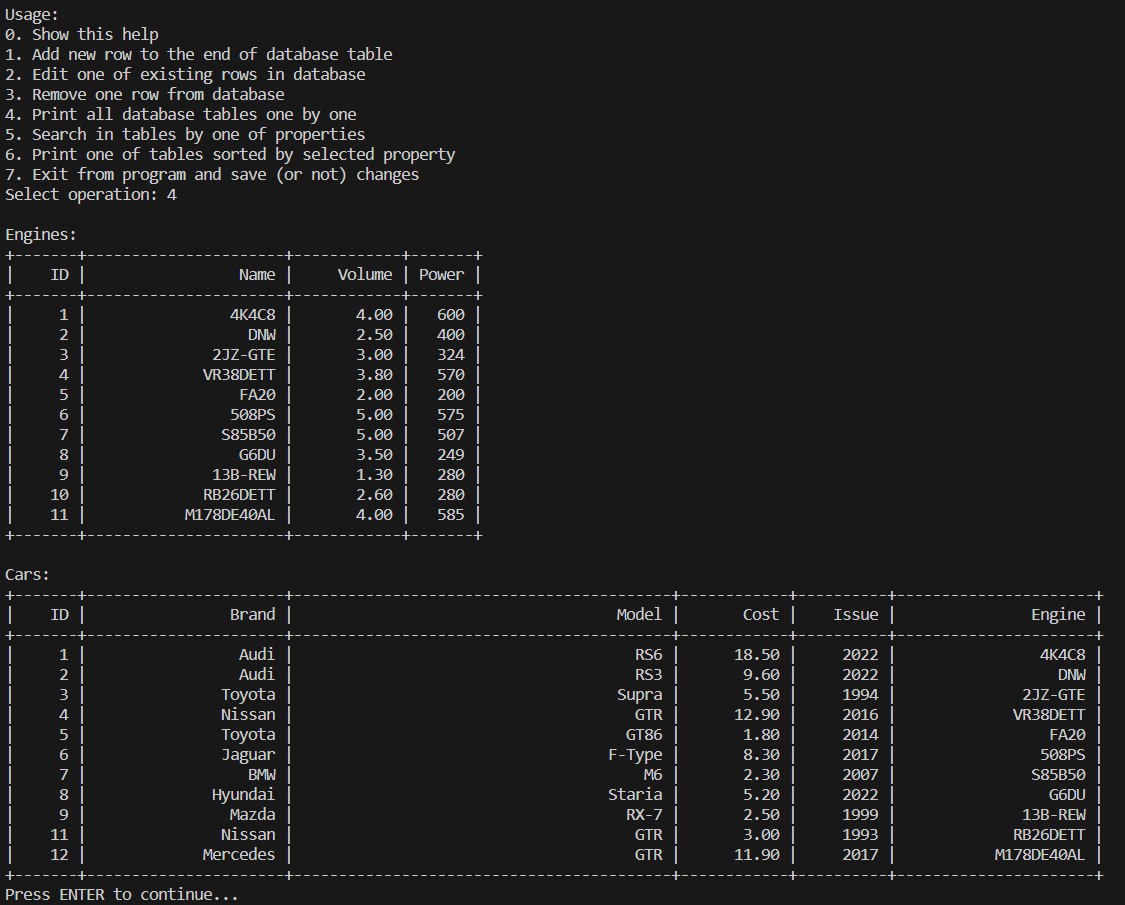
|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | int\_specifier | int | Целочисленный спецификатор |
| 2 | specifier | COURSE\_TASK\_ SPECIFIER\_TYPE | Спецификатор |
| 3 | size | size\_t | Размер |

specifier\_to\_enum

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| 1 | specifier\_enum | COURSE\_TASK\_ SPECIFIER\_TYPE | Спецификатор перечислений |
| 2 | specifier | char[3] | Спецификатор |
| 3 | i | size\_t | Переменная для цикла for |

**Пример выполнения программы.**

**Заключение.**

Используемые заголовочные файлы стандартной библиотеки:

1. stdio.h
   1. fputs – записывает строку в файл
   2. fputc – записывает символ в файл
   3. scanf – считывает данные
   4. fgetc – считывает символ из файла
   5. fopen – открывает файл
   6. fseek – ищет в файле
   7. printf – выводит аргументы
   8. puts – выводит строку
   9. fwrite – записывает объекты в файл
   10. fgets – считывает символы из файла
   11. sscanf – считывает данные из массива
2. stdlib.h
   1. system – передает команду командному процессору
   2. free – освобождает память
   3. malloc – возвращает адрес на первый байт области памяти
   4. realloc – изменяет величину выделенной памяти на новую
3. string.h
   1. strstr – ищет по строке
   2. memcmp – сравнивает символы
   3. strcmp – сравнивает строки
   4. strcpy – копирует из одной строки в другую
   5. strlen – считает длину строки
4. stdarg.h
   1. va\_arg – осуществляет передачу функции переменного числа параметров
   2. va\_start – осуществляет передачу функции переменного числа параметров
5. stddef.h
   1. size\_t – определяет тип size\_t
6. float.h
   1. DBL\_MAX\_10\_EXP – максимальное целое число, при котором число 10, возведенное в степень этого числа, является представимым числом с плавающей запятой.
   2. FLT\_MAX\_10\_EXP – максимальное целое число, при котором число 10, возведенное в степень этого числа, является представимым числом с плавающей запятой.
7. limits.h
   1. ULONG\_MAX – максимальное значение для беззнакового long int
8. math.h
   1. log10 – считает десятичный логарифм числа
   2. ceil – считает наименьшее целое, которое не меньше заданного числа

В результате выполнения работы изучена работа со структурами в языке C и получены практические навыки в создании электронных картотек.