**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**ДВНЗ «УНІВЕРСИТЕТ БАНКІВСЬКОЇ СПРАВИ»**

**ІНСТИТУТ БАНКІВСЬКИХ ТЕХНОЛОГІЙ ТА БІЗНЕСУ**

**КАФЕДРА КІБЕРБЕЗПЕКИ ТА СОЦІАЛЬНИХ НАУК**

**КУРСОВА РОБОТА**

з дисципліни

**«Програмування (Рівень В- Об’єктно-орієнтовне програмування)»**

(назва дисципліни)

на тему: **КАТАЛОГ ЗАПЧАСТИН АВТОМОБІЛЯ**

Студента(ки) 2 курсу 203-Кб групи

спеціальності «Кібербезпека»

. Мамишев О.А .

(прізвище та ініціали)

Керівник

к.т.н., доцент .

(посада, вчене звання, науковий ступінь,

Гордєєв О.О. .

прізвище та ініціали)

Національна шкала

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів: \_\_\_\_

Оцінка: ECTS \_\_\_\_

Члени комісії

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище та ініціали)

Київ - 2019 рік

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**ДВНЗ «УНІВЕРСИТЕТ БАНКІВСЬКОЇ СПРАВИ»**

**ІНСТИТУТ БАНКІВСКЬИХ ТЕХНОЛОГІЙ ТА БІЗНЕСУ**

**КАФЕДРА КІБЕРБЕЗПЕКИ ТА СОЦІАЛЬНИХ НАУК**

Спеціальність: «Кібербезпека»

Курс 2 Група 203-Кб Семестр 3

**ЗАВДАННЯ**

**На курсову роботу студента/студентки**

Мамишева Олександра Андрійовича

1. Тема курсової роботи: Каталог запчастин автомобіля
2. Термін здачі курсової роботи студентом: 12.12.2019
3. Вихiднi данi до (роботи): інформаційно-аналітичні інтернет джерела, зразки рішень практичних завдань та задач, нормативно-правова база, щодо оформлення технічної документації програмних продуктів, початкові значення для обробки.
4. Змiст пояснювальної записки (перелiк питань, які належить розробити) Вступ. Формулювання вимог до програми. Проектування та практична реалізація розв’язку поставленої задачі. Тестування програми і результати її виконання. Висновки.
5. Дата видачі завдання «10» жовтня 2019 р.

**Зміст**

[Вступ 4](#_30j0zll)

[Розділ 1. Загальні відомості 6](#_1fob9te)

[1.1 Формулювання вимог до програми 6](#_3znysh7)

[1.2 Визначення та їх пояснення 6](#_2et92p0)

[Розділ 2. Проектування продукту 11](#_tyjcwt)

[2.1. Опис алгоритму програми 11](#_3dy6vkm)

[2.2 Створення об’єктів і розробка програми. Опис змінних та їх ідентифікаторів, що використовуються. 13](#_1t3h5sf)

[2.3 Опис програми 14](#_4d34og8)

[Розділ 3. Тестовий приклад роботи програми 18](#_1ksv4uv)

[Висновки 20](#_17dp8vu)

[Список використаних джерел 21](#_26in1rg)

[Додатки 22](#_lnxbz9)

Вступ

Мова програмування Сi була розроблена в 1972 р. Деннисом Рiтчi, який є одним з авторів операцiйноi системи UNIX. При створенні С++ прагнули зберегти сумісність з мовою С. Більшість програм на С справно працюватимуть і з компілятором С++. С++ має синтаксис, заснований на синтаксисі С.

Нововведеннями С++ порівняно з С є:

* підтримка об'єктно-орієнтованого програмування через класи;
* підтримка узагальненого програмування через шаблони;
* доповнення до стандартної бібліотеки;
* додаткові типи даних;
* обробка винятків;
* простори імен;
* вбудовані функції;
* перевантаження операторів;
* перевантаження імен функцій;
* посилання і оператори управління вільно розподіленою пам'яттю.

Назва «Сі++» була вигадана Ріком Масситті (Rick Mascitti) і вперше було використана в грудні 1983 року. Раніше, на етапі розробки, нова мова називалася «Сі з класами». Ім'я, що вийшло у результаті, походить від оператора Сі «++» (збільшення значення змінної на одиницю) і поширеному способу присвоєння нових імен комп'ютерним програмам, що полягає в додаванні до імені символу «+» для позначення поліпшень. Згідно Страуструпу, «ця назва указує на еволюційну природу змін Cі».

Переваги мови C++ :

* Продуктивність. Швидкість роботи програм на С++ практично не поступається програмам на С, хоча програмісти отримали в свої руки нові можливості і нові засоби.
* Масштабованість. На мові C++ розробляють програми для самих різних платформ і систем.
* Можливість роботи на низькому рівні з пам'яттю, адресами, портами. (Що, при необережному використанні, може легко перетворитися на недолік.)
* Можливість створення узагальнених алгоритмів для різних типів даних, їх спеціалізація, і обчислення на етапі компіляції, з використанням шаблонів [2].

**Розділ 1. Загальні відомості**

**1.1 Формулювання вимог до програми**

У нашому випадку є наступні вимоги до програми:

* Таблиці: країна, фірма-виробник, марка автомобіля, агрегат, вузол, деталь;
* Облік взаємозамінності;
* Облік користувачів;

**1.2 Визначення та їх пояснення**

Клас на C ++ - це графічний об'єкт, який описує правила, атрибути та об’єкти. Причина - ідеальне зображення символічного розчинення та його принципів. Матеріали, виготовлені на основі одного пристрою, називаються китайськими моделями. Вони можуть мати різні характеристики, але вони завжди матимуть однаковий зміст. В ООП є три основні правила побудови класу:

1. Інкапсуляція - функція, яка дозволяє поєднувати аудиторію та дані та методи, що працюють з ними, та приховати від користувача деталі реалізації.
2. Дотримуйтесь - це особливість, яка дозволяє створити новий дочірній клас на основі наявного, з усіма характеристиками батьківського класу, класу, призначеного дитині.
3. Паліморфізм - властивість класу, яка дозволяє використовувати об’єкти класу з тим самим інтерфейсом без будь-якої інформації про тип та внутрішню структуру об’єкта.

Ми пильно придивляємось до кожної вбудованої властивості класу за потребою, але просто пам’ятаємо ці три. Тепер поверніться до класів, щоб розпочати розробку структури класів. (Рис. 1)

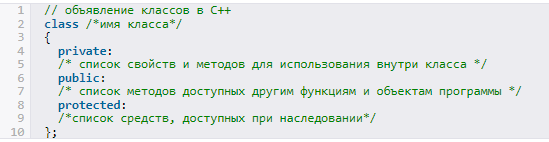
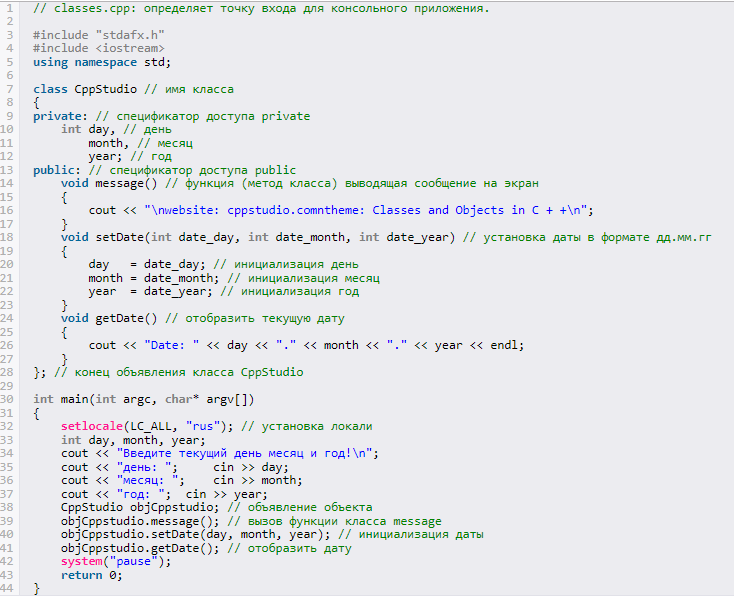


Рис. 1.1. - Структура проголошення занять

Set - функції і get - функції класів

У кожного об'єкта є свої особливості або атрибути, що характеризують його протягом усього життя. Атрибути об'єкта зберігаються в змінних, оголошених в класі, якому належить об'єкт. Крім того, оголошення змінної повинно бути виконано за допомогою приватного доступу. Ці змінні називаються елементами даних. Оскільки елементи даних оголошуються приватними, тільки методи класу можуть отримати доступ до них, зовнішній доступ до елементів даних заборонений. Тому зазвичай оголошують в класах спеціальні методи - так званий набір і отримання функцій, за допомогою яких можна маніпулювати елементами даними (рис. 1.2).

Рис. 1.2 - Реалізація класу з додаванням функцій set та get

Конструктор - це спеціальна функція, яка виконує елементи даних ініціалізації часу, а ім'я конструктора повинно бути обов'язково відповідним імені класу. Важлива відмінність конструктора від інших особливостей полягає в тому, що він не повертає жодних значень, у тому числі недійсних. У будь-якому класі повинен бути конструктор, навіть якщо конструктор не вказано точно (як у попередньому класі), тоді компілятор надає конструктору за замовчуванням без параметрів [3].

Функції - це блоки коду, які виконують певні операції. При необхідності функція може визначити параметри введення, які дозволяють викликає об'єкт передавати йому аргументи. При необхідності функція може також повернути значення в якості вихідного. Функції корисні для інкапсуляції основних операцій в один, багаторазовий прилад. В ідеалі назва цього блоку повинна чітко описувати функцію функції. Наступна функція бере два цілих числа від сміливого об'єкта і повертає їх суму. a і b - параметри типу int.

Параметр функції - це змінна, яка використовується в функції і значення якої надає caller (що викликає об'єкт). Параметри вказуються при оголошенні функції в круглих дужках. Якщо їх багато, то вони перераховуються через кому.

У функції є список параметрів, в якому через кому перераховано необхідне (можливо, нульовий) число типів. Кожному параметру присвоюється ім'я, за яким до нього можна отримати доступ в тілі функції. У шаблоні функції можуть зазначатися додаткові типи або значення параметрів. Зухвалий об'єкт передає аргументи, що представляють собою конкретні значення, типи яких сумісні зі списком параметрів.

За замовчуванням аргументи передаються функції за значенням, тобто функція отримує копію переданого об'єкта. Копіювання великих об'єктів може бути ресурсозатратності і невиправданим.

Типи функцій повернення значень

Функція не може повертати інші функції і вбудовані масиви; Однак можливе повернення покажчиків на ці типи, або лямбда-вираз, що створює об'єкт функції. За винятком того, в цих випадках функція може повернути значення будь-якого типу, який знаходиться в області, або він не повертати ніякого значення, в цьому випадку повертається тип - void.

Всі типи функцій, крім типу [void](https://uk.wikipedia.org/w/index.php?title=Void&action=edit&redlink=1) повертають значення, визначене виразом в операторі return. Тобто, якщо є вказівка, що функція повертає якесь значення, то воно повинно бути заздалегідь зазначено для оператора return. У цьому випадку виконання функції, яка не належить до типу void, повертає довільне значення.

Також у мові C передбачена можливість повернення функцією вказівника. Щоб функція мала можливість повернути вказівник, вона повинна бути оголошена як функція, що повертає вказівник на потрібний тип. Тобто, тип покажчика, що повертається, повинен декларуватися явно. [4]

Змінна — це іменована область пам’яті, у якій зберігаються дані визначеного типу. Змінна має ім’я, розмір та інші атрибути, такі як видимість, час існування тощо. Ім’я змінної служить для звертання до області пам’яті, у якій зберігається її значення. Перед використанням будь-яка змінна повинна бути описана, при цьому для неї резервується деяка область пам’яті, розмір якої залежить від конкретного типу змінної. Під час виконання програми змінна може приймати рiзнi значення.

Ініціювання — це присвоювання змінній при описі початкового значення, яке записується зі знаком рівності — = значення або в круглих дужках — (значення). Зазначимо, що константа повинна бути ініційована при описі. Один оператор може містити опис декількох змінних одного типу, розділяючи їх комами [1].

Компілятоp (від англ. Compile - збирати разом, складати) - системна програма, що виконує перетворення програми, написаної на одній з алгоритмічних мов, у програму мовою, близькою до машинної, і в певному сенсі еквівалентною першій.

**Розділ 2. Проектування продукту**

**2.1. Опис алгоритму програми**

Програма написана за досить простим алгоритмом. Для початку ми вибираємо пункт меню “Додати деталь”. Далі виконується введення інформації про деталь. У нашому випадку їх десять. А саме: «країна», «фірма», «марка машини», «агрегат», «узел», «деталь», «корстувач автосервісу», «користувач магазину запчастин», «постачальник оптового». При виборі команди виконується відповідна дія. Тобто при виборі команди:

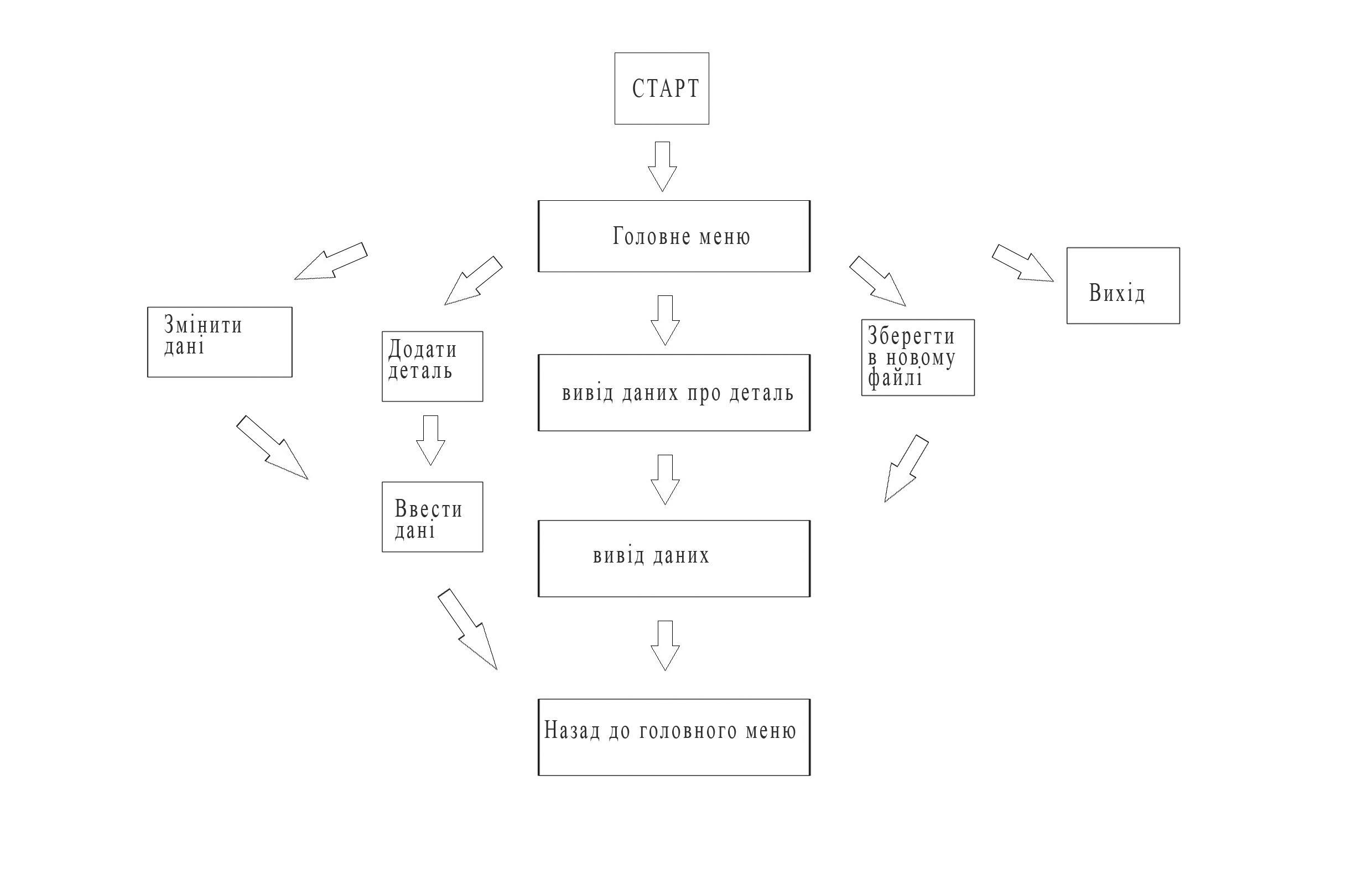
* «Вивід даних про деталь» виконується вивід данних про деталь;
* «Додати деталь» виконується дія «додати деталь»;
* «Змінити дані про деталь» виконується дія «зміна даних»;
* «Зберегти в новому файлі» виконується дія «Збереження даних про деталі та інформації по кожному з них»;
* «вихід» виконується завершення програми.

Наступним кроком є реалізація дій для перших трьох команд відповідно:

* Вивід даних про деталь;
* Додаток деталі;
* Зміна данних про деталь.

При виборі одного з перших п’яти кроків, після їх реалізації ми повертаємось до моменту, де обираємо команду. Такий алгоритм повторюється поки не буде обрано команду «вихід» після чого програма завершує свою роботу.

Рис. 2.1. - Алгоритм



**2.2 Створення об’єктів і розробка програми. Опис змінних та їх ідентифікаторів, що використовуються.**

Вказуємо дві бібліотеки **#include<iostream>** та **# include<string>.**

**Стандартна бібліотека «С++» -** це набір класів і функцій, які написані набазовій мові і є частиною стандарту «С++»ISO**.**

**#include<iostream> -** бібліотека і відповідний заголовний файл, які використовуються для організації введення-виведення в мові «С++».

**# include<string> -** клас з методами і змінними для організації роботи з рядками в мові програмування «С++».

Також прописано **using namespace std**.

**Namespace** – простір імен. Використовується для логічного угрупування класів, функцій і змінних.

**Функції** – це блоки коду, що виконують певні операції.

**Тип** – визначає, які значення може мати змінна, які операції з нею можна робити і скільки байт в пам’яті вона буде займати.

Створюємо функцію типу **void** з назвою *set-reserved(int reserved)*.

Покажчик типу **void –** спеціальний тип покажчика, який може вказувати на об’єкти будь-якого типу даних.

Потім створюємо функції типу **int** *get\_capacity(),get\_reserved(),get\_id().*

**Int** – цілочисельний тип даних.

С**out** – стандартний вихідний потік.

**Endl** – перехід на нову строку.

**Return** – оператор для повернення значен. (Використовується у всіх функціях, крім void).

За допомогою циклу **while** (цикл буде виконуватись до тих пір, поки буде виконуватись умова всередині умови циклу) перевіряємо на вибір команди.

Умова: якщо команда не «quit»

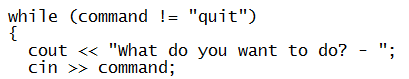


Рис.2.2. – Вибір команди

За допомогою оператора вибору **if**, що знаходиться всередині циклу, виконується дія відповідно команд.

В основній функції **int main()** за допомогою **cout** виводимо запитання про кількість місць на рейсі, та скільки з них зарезервовано.

Виконується цикл **while** поки не буде задано команду *«вихід»*.

У разі введення цієї команди програма завершує роботу.

**2.3 Опис програми**

////В автомобілі налічується кілька тисяч деталей; деякі використовуються в різних марках.Таблиці:

////країна, фірма - виробник, марка автомобіля, агрегат, вузол, деталь.Облік

////взаємозамінності.Користувачі : працівники автосервісу, магазинів запчастин; постачальники оптовики.

char name[100];

char firma[100];

char markacar[100];

char agregat[100];

char uzel[100];

char detal[100];

//Користувачі: працівники автосервісу, магазинів запчастин; постачальники оптовики.

char polzov\_avtocerv[100];

char polzov\_magazin[100];

char postach\_optovik[100];

char File[] = "countries.txt"; //Имя файла cantries

int Count;

FILE\* F\_co; //Логическое имя файла

void sort(state\*& a, int n) //Record\*&a= N;int size=i;

{

int min, j;

state temp;// для связи с полями

temp.name[0] = '\0';//обнуляем

for (int i = 0; i < n - 1; i++)//цикл по всему массиву

{

min = i;//первым этапом индекс минимума равен 0

for (j = i + 1; j < n; j++)//цикл для сравнения со следующей ячейкой масива

if (a[j] < a[min])//если следующ

min = j;//запоминаем индекс

swap(a[i], a[min]);//меняем местами

}

}

void Show(state\*& n)//n массив записей стран который надо отсортировать перед выводом

{/\*

1) После считывания очередной записи в State добавить копию в массив n

2) После сформирования массива n отсортировать его функцией сорт

3) вывести отсортированый масив n

\*/

//Если файл невозможно открыть для чтения, то завершение работы программы

if ((F\_co = fopen(File, "r")) == NULL)

{

fprintf(stderr, "\"%s\" : невозможно открыть\n", File);

}

//Чтение числа записей (Count) в файле

if (fread(&Count, sizeof(int), 1, F\_co) != 1)

{

fprintf(stderr, "\"%s\" : ошибка чтения\n", File);

}

//В цикле осуществляется вывод всех записей

for (i = 0; i < Count; i++)

{

fread(&State, 1, sizeof(State), F\_co); //Читается имя

if (ferror(F\_co)) //Проверяется отсутствие ошибки

if (code == 224)// проверка нажатия одной из управляющих клавиш

{

code = \_getch();

if (code == 80) key++;//cрелка вниз

if (code == 72) key--;//cрелка вверх

}

} while (code != 13);// нажат enter (сделан выбор)- выходим из цикла

//Если файл невозможно открыть для чтения, то завершение работы программы

if ((F\_co = fopen(File, "r+")) == NULL)

{

fprintf(stderr, "\"%s\" : невозможно открыть\n", File);

}

//Чтение числа записей (Count) в файле

if (fread(&Count, sizeof(int), 1, F\_co) != 1)

{

fprintf(stderr, "\"%s\" : ошибка чтения\n", File);

}

//В цикле for осуществляется поиск нужной записи

for (i = 0; i < Count; i++)

{

fread(&State, 1, sizeof(State), F\_co); //Чтение имени

if (ferror(F\_co)) //Проверка отсутствия ошибки

{

fprintf(stderr, "\"%s\" : ошибка чтения\n", File);

return;

}

if (strcmp(s, State.name) == 0)//если такая деталь уже есть выводяться данные о ней

{ //и запрашивается ввод данных для редактирования

void Create(void) //Создает файл, если его не существует

return 0; //Строка str есть в файле

}

fclose(F\_co);

return 1; //Строки str нет в файле

}

void Add(char\* s) //Добавляет запись в файл

{ //

//Чтение числа записей (Count) в файле

if (fread(&Count, sizeof(int), 1, F\_co) != 1)

fwrite(&State, 1, sizeof(State), F\_co); //Запись в файл

if (ferror(F\_co)) //Проверка наличия ошибки

F\_co = fopen(File, "r+");

//Установка указателя в файле на первый байт

if (fseek(F\_co, 0L, SEEK\_SET) != 0)

++Count; //Увеличение числа записей на единицу

//Запись Count в файл

if (fwrite(&Count, sizeof(int), 1, F\_co) != 1)

FILE\* fp2 = fopen(s, "w");// для записи во второй

char cc[1024];

while (!feof(F\_co)) //построчно копируем файл

Find(s);//поиск искомой страны если деталь найдена вывод данных о ней и запрос на ввод новой детали

\*s = NULL; //обнуляем строку

write(s);// код считывания файла и запись под новым именем

\*s = NULL;

**Розділ 3. Тестовий приклад роботи програми**

Рис 3.1. - Приклад роботи програми з використанням усіх команд

Як ми можемо бачити на рис.3 самою першою виведеною фразою є Кількість місць на рейсі, яку ми вказуємо з клавіатури. Наступне речення – кількість зарезервованих місць, яку користувач теж вводить самостійно. Далі це вибір команди. Ось список слів, які відповідають за виклик тієї чи іншої команди:

* add – резервування місць
* remove – віднімання зарезервованих місць на рейсі
* create – додавання рейсу
* cancel – відміна рейсу
* print – виведення на екран
* quit – вихід з програми

Якщо ми обираєм «add», то на екрані з’являється запитання скільки місць ми хочем зарезервувати, потім результат, що вказану кількість місць зарезервовано, або, що рейс переповнений у разі помилки.

При обиранні «remove» висвічується запитання зі скількох місць користувач бажає зняти резерв.

Далі «create», при виборі цієї дії запитується скільки рейсів ми бажаємо створити, скільки у кожному з них всього місць, та скільки з них зарезервовані, для кожного рейсу окремо. Також виводить відсоток зарезервованих місць від загальної кількості на кожному рейсі.

«cancel», за допомогою цього ми можемо видалити будь-який рейс, просто вказавши його номер у відповіді на запитання який рейс потрібно видалити.

Наступним є «print» який виводить усю інформацію по одному з рейсів, номер якого користувач вказує самостійно, як відповідь на запитання з яким рейсом ми хочемо працювати.

Ну і на останок «quit», який завершує роботу програми.

**Висновки**

У цій роботі було наведено теоретичний та практичний варіант реалізації програми «Каталог запчастин для автомобіля». Код було створено за допомогою класів, методів та функцій. Розроблено у онлайн компіляторі Microsoft Visual Studio C++. Також додано блок-схему алгоритму роботи програми та розписано усі деталі покроково.

Користувач може задати кількість запчастин, також є можливість видалити та змінити інформацію. Ще однією можливістю є додавання і видалення збережених деталей.

**Список використаних джерел**

1. Змінні [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: http://cpp.dp.ua/zminni/.
2. Огляд і основи мови програмування C++ [Електронний ресурс] – Режим доступу до ресурсу: <http://www.znannya.org/?view=Cpp_basics>.
3. Класи в С++ [Електронний ресурс] – Режим доступу до ресурсу: <http://cppstudio.com/uk/post/439/>.
4. Functions (C++) [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.microsoft.com/ru-ru/cpp/cpp/functions-cpp?view=vs-2019>.
5. Геттер [Електронний ресурс] – Режим доступу до ресурсу: https://ru.wikipedia.org/wiki/%D0%93%D0%B5%D1%82%D1%82%D0%B5%D1%80\_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)#C++.

**Додатки**

**Додаток А. Лістинг програмного продукту**

////В автомобілі налічується кілька тисяч деталей; деякі використовуються в різних марках.Таблиці:

////країна, фірма - виробник, марка автомобіля, агрегат, вузол, деталь.Облік

////взаємозамінності.Користувачі : працівники автосервісу, магазинів запчастин; постачальники оптовики.

#define \_CRT\_SECURE\_NO\_////В автомобілі налічується кілька тисяч деталей; деякі використовуються в різних марках.Таблиці:

////країна, фірма - виробник, марка автомобіля, агрегат, вузол, деталь.Облік

////взаємозамінності.Користувачі : працівники автосервісу, магазинів запчастин; постачальники оптовики.

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <stdio.h>

#include <windows.h>

#include <conio.h>

using namespace std;

HANDLE hConsole;

HANDLE hStdOut = GetStdHandle(STD\_OUTPUT\_HANDLE);

void SetColor(int text, int background)

{

HANDLE hStdOut = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(hStdOut, (WORD)((background << 4) | text));

}

void GotoXY(int xpos, int ypos)

{

COORD scrn;

HANDLE hOuput = GetStdHandle(STD\_OUTPUT\_HANDLE);

scrn.X = xpos; scrn.Y = ypos;

SetConsoleCursorPosition(hOuput, scrn);

}

struct state

{

char name[100];

char firma[100];

char markacar[100];

char agregat[100];

char uzel[100];

char detal[100];

//Користувачі: працівники автосервісу, магазинів запчастин; постачальники оптовики.

char polzov\_avtocerv[100];

char polzov\_magazin[100];

char postach\_optovik[100];

bool operator <(state t) {

return strcmp(t.name, name) < 0 ? true : false;

}

}State;

char File[] = "countries.txt"; //Имя файла cantries

int Count;

FILE\* F\_co; //Логическое имя файла

char Fileх[40] = {};

template <class T>

void add\_array(T\*& a, int& n, T val)

{

T\* temp = new T[++n];

for (int i = 0; i < n; i++)

temp[i] = i < n - 1 ? a[i] : val;

if (a)

delete[]a;

a = temp;

}

void sort(state\*& a, int n) //Record\*&a= N;int size=i;

{

int min, j;

state temp;// для связи с полями

temp.name[0] = '\0';//обнуляем

for (int i = 0; i < n - 1; i++)//цикл по всему массиву

{

min = i;//первым этапом индекс минимума равен 0

for (j = i + 1; j < n; j++)//цикл для сравнения со следующей ячейкой масива

if (a[j] < a[min])//если следующ

min = j;//запоминаем индекс

swap(a[i], a[min]);//меняем местами

}

}

void Show(state\*& n)//n массив записей стран который надо отсортировать перед выводом

{/\*

1) После считывания очередной записи в State добавить копию в массив n

2) После сформирования массива n отсортировать его функцией сорт

3) вывести отсортированый масив n

\*/

int i, N = 0;

char str[100] = {};

strcpy(State.name, str);

strcpy(State.firma, str);

strcpy(State.markacar, str);

strcpy(State.agregat, str);

strcpy(State.uzel, str);

strcpy(State.detal, str);

strcpy(State.polzov\_avtocerv, str);

strcpy(State.polzov\_magazin, str);

strcpy(State.postach\_optovik, str);

//Если файл невозможно открыть для чтения, то завершение работы программы

if ((F\_co = fopen(File, "r")) == NULL)

{

fprintf(stderr, "\"%s\" : невозможно открыть\n", File);

}

//Чтение числа записей (Count) в файле

if (fread(&Count, sizeof(int), 1, F\_co) != 1)

{

fprintf(stderr, "\"%s\" : ошибка чтения\n", File);

}

//В цикле осуществляется вывод всех записей

for (i = 0; i < Count; i++)

{

fread(&State, 1, sizeof(State), F\_co); //Читается имя

if (ferror(F\_co)) //Проверяется отсутствие ошибки

{

fprintf(stderr, "\"%s\" : ошибка чтения\n", File);

return;

}

add\_array(n, N, State);

strcpy(State.name, str);

strcpy(State.firma, str);

strcpy(State.markacar, str);

strcpy(State.agregat, str);

strcpy(State.uzel, str);

strcpy(State.detal, str);

strcpy(State.polzov\_avtocerv, str);

strcpy(State.polzov\_magazin, str);

strcpy(State.postach\_optovik, str);

}

sort(n, N);

for (int i = 0; i < N; i++)

{

cout << n[i].name << endl;

cout << n[i].firma << endl;

cout << n[i].markacar << endl;

cout << n[i].agregat << endl;

cout << n[i].uzel << endl;

cout << n[i].detal << endl;

cout << n[i].polzov\_avtocerv << endl;

cout << n[i].polzov\_magazin << endl;

cout << n[i].postach\_optovik << endl;

cout << endl;

}

delete n;

n = nullptr;

fclose(F\_co);

\_getch();

}

int menuBegin()

{

int key = 0;

int code;

do {

system("cls");

GotoXY(22, 4);

SetColor(3, 0);

cout << "Выбор действия из меню";

GotoXY(20, 6);

SetColor(15, 0);

cout << "Выберите действие (используя стрелки):";

key = (key + 5) % 5;

if (key == 0)

{

GotoXY(24, 8);

cout << ">> ";

SetColor(0, 2);

cout << "Вывод данных о деталях" << endl;

SetColor(15, 0);

}

else

{

GotoXY(24, 8);

SetColor(15, 0);

cout << " Вывод данных о деталях" << endl;

}

if (key == 1)

{

GotoXY(24, 10);

cout << ">> ";

SetColor(0, 2);

cout << "Добавить деталь" << endl;

SetColor(15, 0);

}

else

{

GotoXY(24, 10);

SetColor(15, 0);

cout << " Добавить деталь" << endl;

}

if (key == 2)

{

GotoXY(24, 12);

cout << ">> ";

SetColor(0, 2);

cout << "Изменить данные о детали" << endl;

SetColor(15, 0);

}

else

{

GotoXY(24, 12);

cout << " Изменить данные о детали" << endl;

}

if (key == 3)

{

GotoXY(24, 14);

cout << ">> ";

SetColor(0, 2);

cout << "Сохранить в новом файле" << endl;

SetColor(15, 0);

}

else

{

GotoXY(24, 14);

cout << " Сохранить в новом файле" << endl;

}

if (key == 4)

{

GotoXY(24, 16);

cout << ">> ";

SetColor(0, 2);

cout << "Выход" << endl;

SetColor(15, 0);

}

else

{

GotoXY(24, 16);

cout << " Выход" << endl;

}

code = \_getch();

if (code == 224)// проверка нажатия одной из управляющих клавиш

{

code = \_getch();

if (code == 80) key++;//cрелка вниз

if (code == 72) key--;//cрелка вверх

}

} while (code != 13);// нажат enter (сделан выбор)- выходим из цикла

system("cls");

return key;

}

//В автомобілі налічується кілька тисяч деталей; деякі використовуються в різних марках.Таблиці:

//країна, фірма - виробник, марка автомобіля, агрегат, вузол, деталь.Облік

//взаємозамінності.Користувачі : працівники автосервісу, магазинів запчастин; постачальники оптовики.

void Find(char\* s)

{

int i;

char str[100] = {};

strcpy(State.name, str);

strcpy(State.firma, str);

strcpy(State.markacar, str);

strcpy(State.agregat, str);

strcpy(State.uzel, str);

strcpy(State.detal, str);

strcpy(State.polzov\_avtocerv, str);

strcpy(State.polzov\_magazin, str);

strcpy(State.postach\_optovik, str);

//Если файл невозможно открыть для чтения, то завершение работы программы

if ((F\_co = fopen(File, "r+")) == NULL)

{

fprintf(stderr, "\"%s\" : невозможно открыть\n", File);

}

//Чтение числа записей (Count) в файле

if (fread(&Count, sizeof(int), 1, F\_co) != 1)

{

fprintf(stderr, "\"%s\" : ошибка чтения\n", File);

}

//В цикле for осуществляется поиск нужной записи

for (i = 0; i < Count; i++)

{

fread(&State, 1, sizeof(State), F\_co); //Чтение имени

if (ferror(F\_co)) //Проверка отсутствия ошибки

{

fprintf(stderr, "\"%s\" : ошибка чтения\n", File);

return;

}

if (strcmp(s, State.name) == 0)//если такая деталь уже есть выводяться данные о ней

{ //и запрашивается ввод данных для редактирования

SetConsoleCP(866);

cout << State.name << endl;

cout << State.firma << endl;

cout << State.markacar << endl;

cout << State.agregat << endl;

cout << State.uzel << endl;

cout << State.detal << endl;

cout << State.polzov\_avtocerv << endl;

cout << State.polzov\_magazin << endl;

cout << State.postach\_optovik << endl;

cout << "Введите страну\n"; SetConsoleCP(1251);

gets\_s(State.name); SetConsoleCP(866);

cout << "Введите фирму\n"; SetConsoleCP(1251);

gets\_s(State.firma); SetConsoleCP(866);

cout << "Введите марку машины\n"; SetConsoleCP(1251);

gets\_s(State.markacar); SetConsoleCP(866);

cout << "Введите агрегат\n"; SetConsoleCP(1251);

gets\_s(State.agregat); SetConsoleCP(866);

cout << "Введите узел\n"; SetConsoleCP(1251);

gets\_s(State.uzel); SetConsoleCP(866);

cout << "Введите деталь\n"; SetConsoleCP(1251);

gets\_s(str);

gets\_s(State.detal); SetConsoleCP(866);

cout << "Введите пользователя автосервису\n"; SetConsoleCP(1251);

gets\_s(State.polzov\_avtocerv); SetConsoleCP(866);

cout << "Введите пользователя магазинов запчастей\n"; SetConsoleCP(1251);

gets\_s(State.polzov\_magazin); SetConsoleCP(866);

cout << "Введите поставщика оптовика\n"; SetConsoleCP(1251);

gets\_s(State.polzov\_magazin); SetConsoleCP(866);

if (fseek(F\_co, (sizeof(state) \* i + 4), SEEK\_SET) != 0)

{

fprintf(stderr, "\"%s\" : ошибка поиска\n", File);

}

fwrite(&State, 1, sizeof(state), F\_co);

fclose(F\_co);

return;

}

}

cout << "Такой запчасти в Вашем справочниек нет\n";

fclose(F\_co);

\_getch();

return;

}

void Create(void) //Создает файл, если его не существует

{

if ((F\_co = fopen(File, "w+")) == NULL)

{

fprintf(stderr, "\%s\" : невозможно открыть\n", File);

return;

}

Count = 0;

if (!fwrite(&Count, sizeof(Count), 1, F\_co))

{

fprintf(stderr, "\"%s\" : ошибка записи\n", File);

return;

}

fclose(F\_co);

}

//---------------------------------------------------------------------

int Choice(char\* str)

{

int i;

if ((F\_co = fopen(File, "r")) == NULL)

{

Create();

return 1;

}

if (fread(&Count, sizeof(int), 1, F\_co) != 1)

{

fprintf(stderr, "\"%s\" : ошибка чтения\n", File);

}

for (i = 0; i < Count; i++)

{

fread(&State, 1, sizeof(State), F\_co);

if (ferror(F\_co))

{

fprintf(stderr, "\"%s\" : ошибка чтения\n", File);

return 1;

}

if (strcmp(str, State.name) == 0)

{

fclose(F\_co);

return 0; //Строка str есть в файле

}

}

fclose(F\_co);

return 1; //Строки str нет в файле

}

void Add(char\* s) //Добавляет запись в файл

{ //

int i;

char str[100] = {};

strcpy(State.name, str);

strcpy(State.firma, str);

strcpy(State.markacar, str);

strcpy(State.agregat, str);

strcpy(State.uzel, str);

strcpy(State.detal, str);

strcpy(State.polzov\_avtocerv, str);

strcpy(State.polzov\_magazin, str);

strcpy(State.postach\_optovik, str);

if ((F\_co = fopen(File, "r+")) == NULL)

{

fprintf(stderr, "\"%s\" : невозможно открыть\n", File);

return;

}

//Чтение числа записей (Count) в файле

if (fread(&Count, sizeof(int), 1, F\_co) != 1)

{

fprintf(stderr, "\"%s\" : ошибка чтения\n", File);

return;

}

strcpy(State.name, s);

cout << State.postach\_optovik << endl;

cout << "Введите страну\n"; SetConsoleCP(1251);

gets\_s(State.name); SetConsoleCP(866);

cout << "Введите фирму\n"; SetConsoleCP(1251);

gets\_s(State.firma); SetConsoleCP(866);

cout << "Введите марку машины\n"; SetConsoleCP(1251);

gets\_s(State.markacar); SetConsoleCP(866);

cout << "Введите агрегат\n"; SetConsoleCP(1251);

gets\_s(State.agregat); SetConsoleCP(866);

cout << "Введите узел\n"; SetConsoleCP(1251);

gets\_s(State.uzel); SetConsoleCP(866);

cout << "Введите деталь\n"; SetConsoleCP(1251);

gets\_s(str);

gets\_s(State.detal); SetConsoleCP(866);

cout << "Введите пользователя автосервису\n"; SetConsoleCP(1251);

gets\_s(State.polzov\_avtocerv); SetConsoleCP(866);

cout << "Введите пользователя магазинов запчастей\n"; SetConsoleCP(1251);

gets\_s(State.polzov\_magazin); SetConsoleCP(866);

cout << "Введите поставщика оптовика\n"; SetConsoleCP(1251);

gets\_s(State.polzov\_magazin); SetConsoleCP(866);

if (fseek(F\_co, sizeof(state) \* Count + 4, SEEK\_SET) != 0)

{

fprintf(stderr, "\"%s\" : ошибка поиска\n", File);

}

fwrite(&State, 1, sizeof(State), F\_co); //Запись в файл

if (ferror(F\_co)) //Проверка наличия ошибки

{

fprintf(stderr, "\"%s\" : ошибка записи\n", File);

return;

}

fclose(F\_co);

F\_co = fopen(File, "r+");

//Установка указателя в файле на первый байт

if (fseek(F\_co, 0L, SEEK\_SET) != 0)

{

fprintf(stderr, "\"%s\" : ошибка позиционирования\n", File);

return;

}

++Count; //Увеличение числа записей на единицу

//Запись Count в файл

if (fwrite(&Count, sizeof(int), 1, F\_co) != 1)

{

fprintf(stderr, "\"%s\" : ошибка записи\n", File);

return;

}

fclose(F\_co);

return;

}

//---------------------------------------------------------------

void write(char\* s) {

F\_co = fopen(File, "r");

FILE\* fp2 = fopen(s, "w");// для записи во второй

char cc[1024];

while (!feof(F\_co)) //построчно копируем файл

{

fgets(cc, sizeof(cc), F\_co);

fputs(cc, fp2);

}

fclose(F\_co);

fclose(fp2);

}

void main()

{

setlocale(LC\_ALL, "");

char s[32] = { 0 };

int answerBegin = 0;

state\* F = nullptr;

while (answerBegin != 4)

{

answerBegin = menuBegin();

switch (answerBegin)

{

case 0:Show(F);

break;

case 1:

cout << "Введите деталь: ";

SetConsoleCP(1251);

if (\_kbhit())

gets\_s(s);

gets\_s(s);

SetConsoleCP(866);

if (Choice(s) == 0) //Нет ли такой записи в файле?

{

puts("Эта деталь есть в справочнике");

Sleep(1000);

}

else //если такой детали нет ввод данных и запись в файл

Add(s);

\*s = NULL;

break;

case 2:cout << "Введите имя: ";

SetConsoleCP(1251);

gets\_s(s);

SetConsoleCP(866);

Find(s);//поиск искомой страны если деталь найдена вывод данных о ней и запрос на ввод новой детали

\*s = NULL; //обнуляем строку

break;

case 3:cout << "Введите имя файла: ";

SetConsoleCP(1251);

gets\_s(s);

SetConsoleCP(866);

write(s);// код считывания файла и запись под новым именем

\*s = NULL;

break;

case 4:exit(0);

}

}

system("pauze");

delete F;

}

//

WARNINGS

#include <iostream>

#include <stdio.h>

#include <windows.h>

#include <conio.h>

using namespace std;

HANDLE hConsole;

HANDLE hStdOut = GetStdHandle(STD\_OUTPUT\_HANDLE);

void SetColor(int text, int background)

{

HANDLE hStdOut = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(hStdOut, (WORD)((background << 4) | text));

}

void GotoXY(int xpos, int ypos)

{

COORD scrn;

HANDLE hOuput = GetStdHandle(STD\_OUTPUT\_HANDLE);

scrn.X = xpos; scrn.Y = ypos;

SetConsoleCursorPosition(hOuput, scrn);

}

struct state

{

char name[100];

char firma[100];

char markacar[100];

char agregat[100];

char uzel[100];

char detal[100];

//Користувачі: працівники автосервісу, магазинів запчастин; постачальники оптовики.

char polzov\_avtocerv[100];

char polzov\_magazin[100];

char postach\_optovik[100];

bool operator <(state t) {

return strcmp(t.name, name) < 0 ? true : false;

}

}State;

char File[] = "countries.txt"; //Имя файла cantries

int Count;

FILE\* F\_co; //Логическое имя файла

char Fileх[40] = {};

template <class T>

void add\_array(T\*& a, int& n, T val)

{

T\* temp = new T[++n];

for (int i = 0; i < n; i++)

temp[i] = i < n - 1 ? a[i] : val;

if (a)

delete[]a;

a = temp;

}

void sort(state\*& a, int n) //Record\*&a= N;int size=i;

{

int min, j;

state temp;// для связи с полями

temp.name[0] = '\0';//обнуляем

for (int i = 0; i < n - 1; i++)//цикл по всему массиву

{

min = i;//первым этапом индекс минимума равен 0

for (j = i + 1; j < n; j++)//цикл для сравнения со следующей ячейкой масива

if (a[j] < a[min])//если следующ

min = j;//запоминаем индекс

swap(a[i], a[min]);//меняем местами

}

}

void Show(state\*& n)//n массив записей стран который надо отсортировать перед выводом

{/\*

1) После считывания очередной записи в State добавить копию в массив n

2) После сформирования массива n отсортировать его функцией сорт

3) вывести отсортированый масив n

\*/

int i, N = 0;

char str[100] = {};

strcpy(State.name, str);

strcpy(State.firma, str);

strcpy(State.markacar, str);

strcpy(State.agregat, str);

strcpy(State.uzel, str);

strcpy(State.detal, str);

strcpy(State.polzov\_avtocerv, str);

strcpy(State.polzov\_magazin, str);

strcpy(State.postach\_optovik, str);

//Если файл невозможно открыть для чтения, то завершение работы программы

if ((F\_co = fopen(File, "r")) == NULL)

{

fprintf(stderr, "\"%s\" : невозможно открыть\n", File);

}

//Чтение числа записей (Count) в файле

if (fread(&Count, sizeof(int), 1, F\_co) != 1)

{

fprintf(stderr, "\"%s\" : ошибка чтения\n", File);

}

//В цикле осуществляется вывод всех записей

for (i = 0; i < Count; i++)

{

fread(&State, 1, sizeof(State), F\_co); //Читается имя

if (ferror(F\_co)) //Проверяется отсутствие ошибки

{

fprintf(stderr, "\"%s\" : ошибка чтения\n", File);

return;

}

add\_array(n, N, State);

strcpy(State.name, str);

strcpy(State.firma, str);

strcpy(State.markacar, str);

strcpy(State.agregat, str);

strcpy(State.uzel, str);

strcpy(State.detal, str);

strcpy(State.polzov\_avtocerv, str);

strcpy(State.polzov\_magazin, str);

strcpy(State.postach\_optovik, str);

}

sort(n, N);

for (int i = 0; i < N; i++)

{

cout << n[i].name << endl;

cout << n[i].firma << endl;

cout << n[i].markacar << endl;

cout << n[i].agregat << endl;

cout << n[i].uzel << endl;

cout << n[i].detal << endl;

cout << n[i].polzov\_avtocerv << endl;

cout << n[i].polzov\_magazin << endl;

cout << n[i].postach\_optovik << endl;

cout << endl;

}

delete n;

n = nullptr;

fclose(F\_co);

\_getch();

}

int menuBegin()

{

int key = 0;

int code;

do {

system("cls");

GotoXY(22, 4);

SetColor(3, 0);

cout << "Выбор действия из меню";

GotoXY(20, 6);

SetColor(15, 0);

cout << "Выберите действие (используя стрелки):";

key = (key + 5) % 5;

if (key == 0)

{

GotoXY(24, 8);

cout << ">> ";

SetColor(0, 2);

cout << "Вывод данных о деталях" << endl;

SetColor(15, 0);

}

else

{

GotoXY(24, 8);

SetColor(15, 0);

cout << " Вывод данных о деталях" << endl;

}

if (key == 1)

{

GotoXY(24, 10);

cout << ">> ";

SetColor(0, 2);

cout << "Добавить деталь" << endl;

SetColor(15, 0);

}

else

{

GotoXY(24, 10);

SetColor(15, 0);

cout << " Добавить деталь" << endl;

}

if (key == 2)

{

GotoXY(24, 12);

cout << ">> ";

SetColor(0, 2);

cout << "Изменить данные о детали" << endl;

SetColor(15, 0);

}

else

{

GotoXY(24, 12);

cout << " Изменить данные о детали" << endl;

}

if (key == 3)

{

GotoXY(24, 14);

cout << ">> ";

SetColor(0, 2);

cout << "Сохранить в новом файле" << endl;

SetColor(15, 0);

}

else

{

GotoXY(24, 14);

cout << " Сохранить в новом файле" << endl;

}

if (key == 4)

{

GotoXY(24, 16);

cout << ">> ";

SetColor(0, 2);

cout << "Выход" << endl;

SetColor(15, 0);

}

else

{

GotoXY(24, 16);

cout << " Выход" << endl;

}

code = \_getch();

if (code == 224)// проверка нажатия одной из управляющих клавиш

{

code = \_getch();

if (code == 80) key++;//cрелка вниз

if (code == 72) key--;//cрелка вверх

}

} while (code != 13);// нажат enter (сделан выбор)- выходим из цикла

system("cls");

return key;

}

//В автомобілі налічується кілька тисяч деталей; деякі використовуються в різних марках.Таблиці:

//країна, фірма - виробник, марка автомобіля, агрегат, вузол, деталь.Облік

//взаємозамінності.Користувачі : працівники автосервісу, магазинів запчастин; постачальники оптовики.

void Find(char\* s)

{

int i;

char str[100] = {};

strcpy(State.name, str);

strcpy(State.firma, str);

strcpy(State.markacar, str);

strcpy(State.agregat, str);

strcpy(State.uzel, str);

strcpy(State.detal, str);

strcpy(State.polzov\_avtocerv, str);

strcpy(State.polzov\_magazin, str);

strcpy(State.postach\_optovik, str);

//Если файл невозможно открыть для чтения, то завершение работы программы

if ((F\_co = fopen(File, "r+")) == NULL)

{

fprintf(stderr, "\"%s\" : невозможно открыть\n", File);

}

//Чтение числа записей (Count) в файле

if (fread(&Count, sizeof(int), 1, F\_co) != 1)

{

fprintf(stderr, "\"%s\" : ошибка чтения\n", File);

}

//В цикле for осуществляется поиск нужной записи

for (i = 0; i < Count; i++)

{

fread(&State, 1, sizeof(State), F\_co); //Чтение имени

if (ferror(F\_co)) //Проверка отсутствия ошибки

{

fprintf(stderr, "\"%s\" : ошибка чтения\n", File);

return;

}

if (strcmp(s, State.name) == 0)//если такая деталь уже есть выводяться данные о ней

{ //и запрашивается ввод данных для редактирования

SetConsoleCP(866);

cout << State.name << endl;

cout << State.firma << endl;

cout << State.markacar << endl;

cout << State.agregat << endl;

cout << State.uzel << endl;

cout << State.detal << endl;

cout << State.polzov\_avtocerv << endl;

cout << State.polzov\_magazin << endl;

cout << State.postach\_optovik << endl;

cout << "Введите страну\n"; SetConsoleCP(1251);

gets\_s(State.name); SetConsoleCP(866);

cout << "Введите фирму\n"; SetConsoleCP(1251);

gets\_s(State.firma); SetConsoleCP(866);

cout << "Введите марку машины\n"; SetConsoleCP(1251);

gets\_s(State.markacar); SetConsoleCP(866);

cout << "Введите агрегат\n"; SetConsoleCP(1251);

gets\_s(State.agregat); SetConsoleCP(866);

cout << "Введите узел\n"; SetConsoleCP(1251);

gets\_s(State.uzel); SetConsoleCP(866);

cout << "Введите деталь\n"; SetConsoleCP(1251);

gets\_s(str);

gets\_s(State.detal); SetConsoleCP(866);

cout << "Введите пользователя автосервису\n"; SetConsoleCP(1251);

gets\_s(State.polzov\_avtocerv); SetConsoleCP(866);

cout << "Введите пользователя магазинов запчастей\n"; SetConsoleCP(1251);

gets\_s(State.polzov\_magazin); SetConsoleCP(866);

cout << "Введите поставщика оптовика\n"; SetConsoleCP(1251);

gets\_s(State.polzov\_magazin); SetConsoleCP(866);

if (fseek(F\_co, (sizeof(state) \* i + 4), SEEK\_SET) != 0)

{

fprintf(stderr, "\"%s\" : ошибка поиска\n", File);

}

fwrite(&State, 1, sizeof(state), F\_co);

fclose(F\_co);

return;

}

}

cout << "Такой запчасти в Вашем справочниек нет\n";

fclose(F\_co);

\_getch();

return;

}

void Create(void) //Создает файл, если его не существует

{

if ((F\_co = fopen(File, "w+")) == NULL)

{

fprintf(stderr, "\%s\" : невозможно открыть\n", File);

return;

}

Count = 0;

if (!fwrite(&Count, sizeof(Count), 1, F\_co))

{

fprintf(stderr, "\"%s\" : ошибка записи\n", File);

return;

}

fclose(F\_co);

}

//---------------------------------------------------------------------

int Choice(char\* str)

{

int i;

if ((F\_co = fopen(File, "r")) == NULL)

{

Create();

return 1;

}

if (fread(&Count, sizeof(int), 1, F\_co) != 1)

{

fprintf(stderr, "\"%s\" : ошибка чтения\n", File);

}

for (i = 0; i < Count; i++)

{

fread(&State, 1, sizeof(State), F\_co);

if (ferror(F\_co))

{

fprintf(stderr, "\"%s\" : ошибка чтения\n", File);

return 1;

}

if (strcmp(str, State.name) == 0)

{

fclose(F\_co);

return 0; //Строка str есть в файле

}

}

fclose(F\_co);

return 1; //Строки str нет в файле

}

void Add(char\* s) //Добавляет запись в файл

{ //

int i;

char str[100] = {};

strcpy(State.name, str);

strcpy(State.firma, str);

strcpy(State.markacar, str);

strcpy(State.agregat, str);

strcpy(State.uzel, str);

strcpy(State.detal, str);

strcpy(State.polzov\_avtocerv, str);

strcpy(State.polzov\_magazin, str);

strcpy(State.postach\_optovik, str);

if ((F\_co = fopen(File, "r+")) == NULL)

{

fprintf(stderr, "\"%s\" : невозможно открыть\n", File);

return;

}

//Чтение числа записей (Count) в файле

if (fread(&Count, sizeof(int), 1, F\_co) != 1)

{

fprintf(stderr, "\"%s\" : ошибка чтения\n", File);

return;

}

strcpy(State.name, s);

cout << State.postach\_optovik << endl;

cout << "Введите страну\n"; SetConsoleCP(1251);

gets\_s(State.name); SetConsoleCP(866);

cout << "Введите фирму\n"; SetConsoleCP(1251);

gets\_s(State.firma); SetConsoleCP(866);

cout << "Введите марку машины\n"; SetConsoleCP(1251);

gets\_s(State.markacar); SetConsoleCP(866);

cout << "Введите агрегат\n"; SetConsoleCP(1251);

gets\_s(State.agregat); SetConsoleCP(866);

cout << "Введите узел\n"; SetConsoleCP(1251);

gets\_s(State.uzel); SetConsoleCP(866);

cout << "Введите деталь\n"; SetConsoleCP(1251);

gets\_s(str);

gets\_s(State.detal); SetConsoleCP(866);

cout << "Введите пользователя автосервису\n"; SetConsoleCP(1251);

gets\_s(State.polzov\_avtocerv); SetConsoleCP(866);

cout << "Введите пользователя магазинов запчастей\n"; SetConsoleCP(1251);

gets\_s(State.polzov\_magazin); SetConsoleCP(866);

cout << "Введите поставщика оптовика\n"; SetConsoleCP(1251);

gets\_s(State.polzov\_magazin); SetConsoleCP(866);

if (fseek(F\_co, sizeof(state) \* Count + 4, SEEK\_SET) != 0)

{

fprintf(stderr, "\"%s\" : ошибка поиска\n", File);

}

fwrite(&State, 1, sizeof(State), F\_co); //Запись в файл

if (ferror(F\_co)) //Проверка наличия ошибки

{

fprintf(stderr, "\"%s\" : ошибка записи\n", File);

return;

}

fclose(F\_co);

F\_co = fopen(File, "r+");

//Установка указателя в файле на первый байт

if (fseek(F\_co, 0L, SEEK\_SET) != 0)

{

fprintf(stderr, "\"%s\" : ошибка позиционирования\n", File);

return;

}

++Count; //Увеличение числа записей на единицу

//Запись Count в файл

if (fwrite(&Count, sizeof(int), 1, F\_co) != 1)

{

fprintf(stderr, "\"%s\" : ошибка записи\n", File);

return;

}

fclose(F\_co);

return;

}

//---------------------------------------------------------------

void write(char\* s) {

F\_co = fopen(File, "r");

FILE\* fp2 = fopen(s, "w");// для записи во второй

char cc[1024];

while (!feof(F\_co)) //построчно копируем файл

{

fgets(cc, sizeof(cc), F\_co);

fputs(cc, fp2);

}

fclose(F\_co);

fclose(fp2);

}

void main()

{

setlocale(LC\_ALL, "");

char s[32] = { 0 };

int answerBegin = 0;

state\* F = nullptr;

while (answerBegin != 4)

{

answerBegin = menuBegin();

switch (answerBegin)

{

case 0:Show(F);

break;

case 1:

cout << "Введите деталь: ";

SetConsoleCP(1251);

if (\_kbhit())

gets\_s(s);

gets\_s(s);

SetConsoleCP(866);

if (Choice(s) == 0) //Нет ли такой записи в файле?

{

puts("Эта деталь есть в справочнике");

Sleep(1000);

}

else //если такой детали нет ввод данных и запись в файл

Add(s);

\*s = NULL;

break;

case 2:cout << "Введите имя: ";

SetConsoleCP(1251);

gets\_s(s);

SetConsoleCP(866);

Find(s);//поиск искомой страны если деталь найдена вывод данных о ней и запрос на ввод новой детали

\*s = NULL; //обнуляем строку

break;

case 3:cout << "Введите имя файла: ";

SetConsoleCP(1251);

gets\_s(s);

SetConsoleCP(866);

write(s);// код считывания файла и запись под новым именем

\*s = NULL;

break;

case 4:exit(0);

}

}

system("pauze");

delete F;

}

//

