Міністерство освіти і науки України

Національний університет харчових технологій

Кафедра інформаційних систем

#### Курсова робота

Основи програмування та алгоритмічні мови

на тему: структурований файл “Автомобілі”

студента групи АКС 3-9

напрям „ Комп’ютерні науки”

Грикуна Д. В.

Варіант № 7

навчальний шифр (номер залікової книжки)151497

Київ – 2014

# Анотації

Під час виконання курсової роботи було створено інформаційно-пошукову систему автомобілів. В системі реалізовано введення, виведення, редагування та пошуку даних в базі даних, яка представлена типізованим файлом з певною структурою записів даних. Поля запису – модель (до 15 символів), номер (код регіону, цифровий код, буквений код), колір, дані про власника (прізвище, ім’я, по-батькові), дата останнього техогляду. Система дозволяє створювати та відкривати файли бази даних. В системі забезпечено реалізацію трьох довідок з виведенням результатів пошуку на екран.

Для розробки даної системи було використано мову програмування C/C++ та інтегроване середовище розробки Qt Creator.

# Annotations

During the course work was created information and search engine car. Implemented in the system input, output, editing and searching data in a database, which is represented by typed files with a certain structure of data records. Fields - model (up to 15 characters), number (area code, digital code, code alpha), color data, wner (first name, middle name), date of last inspection. The system allows users to open files database. In the implementation of the system provided three references from breeding results on the screen.

To develop this system were used programming language C / C ++ and integrated development environment Qt Creator.

**Зміст**

[Анотації 2](#__RefHeading__3147_1192490275)

[Annotations 3](#__RefHeading__3149_1192490275)

[Вступ 5](#__RefHeading__3151_1192490275)

[Постановка задачі 6](#__RefHeading__3153_1192490275)

[Опис алгоритмів і програм 7](#__RefHeading__3155_1192490275)

[Клас запису у файлі: 7](#__RefHeading__3157_1192490275)

[Основні процедури: 8](#__RefHeading__3159_1192490275)

[Текст програми 9](#__RefHeading__3161_1192490275)

[Результати роботи 22](#__RefHeading__3167_1192490275)

[Інструкція користувача 24](#__RefHeading__3169_1192490275)

[Висновок 25](#__RefHeading__3171_1192490275)

[Список літератури 26](#__RefHeading__3173_1192490275)

# Вступ

Мова програмування − формалізована мова, призначена для описування алгоритмів розв’язування задач на ЕОМ. Мова програмування потрібна, щоб записувати алгоритм у термінах, зрозумілих компіляторові, який, своєю чергою, перекладе його на зрозумілу процесорові машинну мову. Як і кожна людська мова, мова програмування складається з алфавіту (вживаних символів), слів (елементарних конструкцій, які називають лексемами), словосполучень (виразів) та речень (операторів).

C++ (Сі-плюс-плюс) — мова програмування високого рівня з підтримкою декількох парадигм програмування: об'єктно-орієнтованої, узагальненої та процедурної. C++ є результатом еволюції мови С, створеної Денісом Рітчі та Брайаном Керніганом. Ще на початку 70-х років минулого сторіччя С набула широкої по-пулярності як мова розробляння операційної системи UNIX. У 1990-х роках С++ стала однією з найуживаніших мов програмування загального призначення. Мову використовують для системного програмування, розробки програмного забезпечення, написання драйверів, потужних серверних та клієнтських програм, а також для розробки розважальних програм таких як відеоігри. С++ суттєво вплинула на інші, популярні сьогодні, мови програмування: С# та Java. Програми, складені мовою С++, можуть бути виконані на ПК різних виробників і в різних операційних системах. У наступні два десятиліття мову С++ було збагачено й стандартизовано. Її втілено в кількох сучасних системах програму- вання, зокрема в Microsoft Visual C++. Нащадками мови С++ є мови Java (Джава) та C# (Сі-шарп або Сі-діез), спеціалізовані для програмування в сучасних комп'ю- терних мережах. Ці мови за структурою схожі на С++, тому, во- лодіючи С++, неважко перейти на Java або C#, і навпаки.

# Постановка задачі

Створити типізований файл записів, який містить дані про автомобілі. Поля запису – модель (до 15 символів), номер (код регіону, цифровий код, буквений код), колір, дані про власника (прізвище, ім’я, по-батькові), дата останнього техогляду. Довідки для пошуку:

1) даних про всі автомобілі, які не пройшли техогляд в цьому році;

2) інформації про всіх власників авто певного кольору;

3) всю інформації про авто та її власника за номером.

# Опис алгоритмів і програм

## **Клас** **запису у файлі:**

class Car

{

private:

std::string model;

std::string number;

std::string color;

std::string holder;

std::string date;

public:

std::string getNumber();

std::string getModel();

std::string getColor();

std::string getHolder();

std::string getYear();

std::string getDate();

void setNumber();

void setColor();

void setHolder();

void setDate();

void setModel();

void setNumber(std::string str);

void setColor(std::string str);

void setHolder(std::string str);

void setDate(std::string str);

void setModel(std::string str);

Car();

};

## Основні процедури:

1. addRecord() - додає новий запис в файл.
2. void viewRecords(std::vector<Car \*> arr) — відображає всі записи.
3. void findByDate() - пошук машин, які не пройшли техогляд за останній рік.
4. void findByColor() - пошук за кольором.
5. void findByNumber() - пошук за номером.
6. void writeToFile() - зберігання файлу.
7. void readFromFile() зчитування з файлу.

# Текст програми

**Файл car.h**

#ifndef CAR\_H

#define CAR\_H

#include <iostream>

class Car

{

private:

std::string model;

std::string number;

std::string color;

std::string holder;

std::string date;

public:

std::string getNumber();

std::string getModel();

std::string getColor();

std::string getHolder();

std::string getYear();

std::string getDate();

void setNumber();

void setColor();

void setHolder();

void setDate();

void setModel();

void setNumber(std::string str);

void setColor(std::string str);

void setHolder(std::string str);

void setDate(std::string str);

void setModel(std::string str);

Car();

};

#endif // CAR\_H

**Файл car.cpp**

#include "car.h"

#include <istream>

#include <string>

#include <iostream>

std::string Car::getNumber()

{

return number;

}

std::string Car::getModel()

{

return model;

}

std::string Car::getColor()

{

return color;

}

std::string Car::getHolder()

{

return holder;

}

std::string Car::getYear()

{

std::string temp;

for(int i = 0; i < 4; i++)

temp += date[6+i];

return temp;

}

std::string Car::getDate()

{

return date;

}

void Car::setNumber()

{

std::string temp;

std::cout << "Введіть номер автомобіля: \n";

std::getline(std::cin, temp);

number = temp;

fflush(stdin);

std::cin.clear();

}

void Car::setColor()

{

std::string temp;

std::cout << "Введіть колір автомобіля: \n";

std::getline(std::cin, temp);

color = temp;

fflush(stdin);

std::cin.clear();

}

void Car::setHolder()

{

std::string temp;

std::cout << "Введіть власника автомобіля: ";

std::getline(std::cin, temp);

holder = temp;

fflush(stdin);

std::cin.clear();

}

void Car::setDate()

{

std::string temp;

std::cout << "Введіть дату техогляду автомобіля: \n";

std::getline(std::cin, temp);

date = temp;

fflush(stdin);

std::cin.clear();

}

void Car::setModel()

{

std::string temp;

std::cout << "Введіть модель автомобіля: \n";

std::getline(std::cin, temp);

model = temp;

fflush(stdin);

std::cin.clear();

}

void Car::setModel(std::string str)

{

model = str;

}

void Car::setNumber(std::string str)

{

number = str;

}

void Car::setColor(std::string str)

{

color = str;

}

void Car::setDate(std::string str)

{

date = str;

}

void Car::setHolder(std::string str)

{

holder = str;

}

Car::Car()

{

}

**Файл menu.h**

#ifndef MENU\_H

#define MENU\_H

#include <stdio.h>

#include <iostream>

#include <vector>

#include "car.h"

class Menu

{

public:

void mainMenu();

private:

void addRecord();

void viewRecords(std::vector<Car \*> arr);

void find(int selection);

void findByDate();

void findByColor();

void findByNumber();

void writeToFile();

void readFromFile();

bool keyX(unsigned char command);

bool keyEnter(unsigned char command);

bool keyV(unsigned char command);

bool keyF(unsigned char command);

std::vector<Car\*> cars;

};

#endif // MENU\_H

**Файл menu.cpp**

#include "menu.h"

#include <fstream>

void Menu::mainMenu()

{

bool exit = false;

unsigned char commandChar;

readFromFile();

printf("\t\tАвтомобілі\n\n");

printf("Натисніть Enter для додавання автомобіля, клавішу V для перегляду всіх автомобілів, F для пошуку та Х для виходу\n");

while(!exit)

{

commandChar = getchar();

if(keyEnter(commandChar))

{

addRecord();

}

else if(keyF(commandChar))

{

std::cout << "\n";

system("clear");

int i;

std::cout << "Натисніть 1 для виведення всіх автомобілів, що не пройшли техогляд в цьому році, "

"2 для пошуку автомобіля за кольором і 3 для пошуку за номером\n";

i = (int)getchar() - 48;

std::cout << "\n";

system("clear");

find(i);

}

else if(keyV(commandChar))// Enter

{

viewRecords(cars);

}

else if (!exit)

{

fflush(stdin);

printf("\n\t\tАвтомобілі");

system("clear");

printf("\nВи ввели неправильну команду.\n");

}

printf("Натисніть Enter для додавання автомобіля, клавішу V для перегляду всіх автомобілів, F для пошуку та Х для виходу\n");

if(keyX(commandChar))// X

{

writeToFile();

exit = true;

printf("\n");

system("clear");

}

}

}

void Menu::addRecord()

{

std::cout << "\n";

system("clear");

Car\* car = new Car;

car->setNumber();

car->setColor();

car->setHolder();

car->setDate();

car->setModel();

cars.push\_back(car);

//getchar();

fflush(stdin);

std::cout << "\n";

system("clear");

}

void Menu::viewRecords(std::vector<Car\*> arr)

{

std::cout << "\n";

system("clear");

for(int i = 0; i < arr.size(); i++)

{

std::cout << arr.at(i)->getNumber();

std::cout << " ";

std::cout << arr.at(i)->getModel();

std::cout << " ";

std::cout << arr.at(i)->getColor();

std::cout << " ";

std::cout << arr.at(i)->getHolder();

std::cout << " ";

std::cout << arr.at(i)->getDate();

std::cout << "\n\n";

}

}

void Menu::find(int selection)

{

switch (selection) {

case 1:

findByDate();

break;

case 2:

findByColor();

break;

case 3:

findByNumber();

break;

default:

break;

}

getchar();

std::cout << std::endl;

}

void Menu::findByDate()

{

std::string year;

std::cout << "Введіть рік: ";

std::cin >> year;

std::vector<Car\*> temp;

for(int i = 0; i < cars.size(); i++)

{

if(cars.at(i)->getYear() < year)

{

std::string dr = cars.at(i)->getDate();

temp.push\_back(cars.at(i));

}

}

viewRecords(temp);;

}

void Menu::findByColor()

{

std::string color;

std::cout << "Введіть колір для пошуку: ";

std::cin >> color;

std::vector<Car\*> temp;

for(int i = 0; i < cars.size(); i++)

{

if(cars.at(i)->getColor() == color)

{

temp.push\_back(cars.at(i));

}

}

viewRecords(temp);

}

void Menu::findByNumber()

{

std::string number;

std::cout << "Введіть номер для пошуку: ";

std::cin >> number;

std::vector<Car\*> temp;

for(int i = 0; i < cars.size(); i++)

{

if(cars.at(i)->getNumber() == number)

{

temp.push\_back(cars.at(i));

}

}

viewRecords(temp);

}

void Menu::writeToFile()

{

std::ofstream file ("cars.txt");

if(file.is\_open())

{

for(int i = 0; i < cars.size(); i++)

{

file << cars.at(i)->getModel() << "~";

file << cars.at(i)->getColor() << "~";

file << cars.at(i)->getNumber() << "~";

file << cars.at(i)->getHolder() << "~";

file << cars.at(i)->getDate() << "~\n";

}

file.close();

}

else

{

std::cout << "Can't open file!";

}

}

void Menu::readFromFile()

{

std::string line;

std::ifstream file ("cars.txt");

if(file.is\_open())

{

while (std::getline(file, line))

{

std::string temp;

Car\* car = new Car;

int i = 0;

while (line[i] != '~')

{

temp += line[i];

i++;

}

car->setModel(temp);

temp = "";

i++;

while (line[i] != '~')

{

temp += line[i];

i++;

}

car->setColor(temp);

temp = "";

i++;

while (line[i] != '~')

{

temp += line[i];

i++;

}

car->setNumber(temp);

temp = "";

i++;

while (line[i] != '~')

{

temp += line[i];

i++;

}

car->setHolder(temp);

temp = "";

i++;

while (line[i] != '~')

{

temp += line[i];

i++;

}

car->setDate(temp);

temp = "";

cars.push\_back(car);

}

file.close();

}

else

{

std::cout << "Can't open file!";

}

}

bool Menu::keyX(unsigned char command)

{

if(command == 135 || command == 120 || command == 88 || command == 167)

{

return true;

}

return false;

}

bool Menu::keyEnter(unsigned char command)

{

if(command == 10)

{

return true;

}

return false;

}

bool Menu::keyV(unsigned char command)

{

if(command == 118 || command == 86 || command == 188 || command == 156)

{

return true;

}

return false;

}

bool Menu::keyF(unsigned char command)

{

if(command == 102 || command == 70 || command == 176 || command == 144)

{

return true;

}

return false;

}

**Файл main.cpp**

#include <iostream>

#include <termios.h>

#include <menu.h>

//7. Створити типований файл записів, який містить дані про автомобілі. Поля #запису – модель (до 15 символів),

//номер (код регіону, цифровий код, буквенний #код), колір, дані про власника (прізвище, ім’я, по-батькові),

//дата останнього #техогляду. Довідки для пошуку:

//даних про всі автомобілі, які не пройшли техогляд в цьому році;

//інформації про всіх власників авто певного кольору;

//всю інформації про авто та її власника за номером.

using namespace std;

struct TerminalOpt {

TerminalOpt() {

termios new\_settings;

tcgetattr(0,&stored\_settings);

new\_settings = stored\_settings;

new\_settings.c\_lflag &= (~ICANON);

new\_settings.c\_cc[VTIME] = 0;

new\_settings.c\_cc[VMIN] = 1;

tcsetattr(0,TCSANOW,&new\_settings);

}

~TerminalOpt() {

tcsetattr(0,TCSANOW,&stored\_settings);

}

termios stored\_settings;

};

int main()

{

setlocale(LC\_ALL,"Russian");

TerminalOpt term;

Menu menu;

menu.mainMenu();

return 0;

}

# Результати роботи

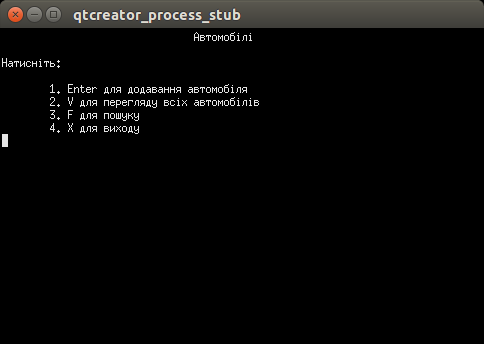


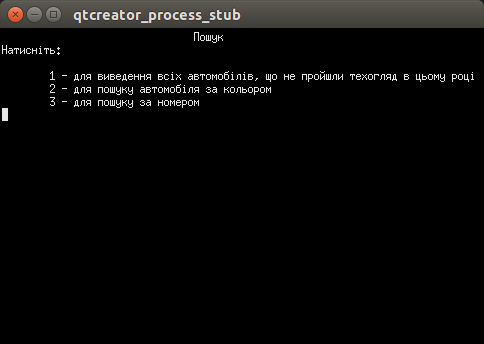
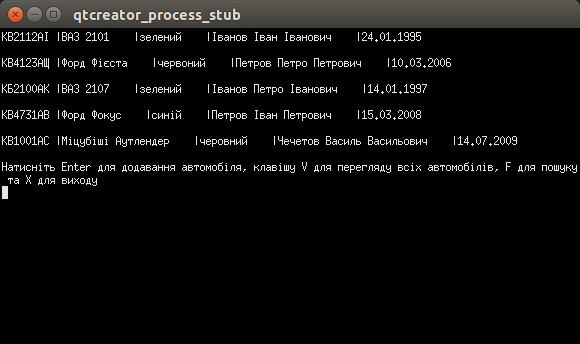
Рис.1 Головне меню

Рис.2 Меню пошуків

Рис.3 Усі запити

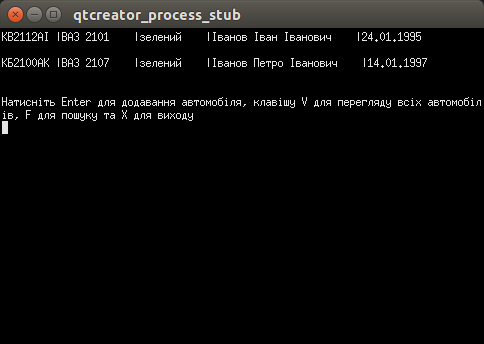


Рис.4 Пошук за кольором

# Інструкція користувача

Після запуску програми у вікні з'являється головне меню з 4 варіантами.

Вибравши пункт "Додати автомобіль", користувач створює новий запис вводячи дані про техніку. Після введення користувача буде повернуто до головного меню.

Клавіша V дозволяє вивести на екран всі автомобілі в базі даних.

Пункт 3 "Пошук" зберігає в собі друге меню в якому користувач може вибрати 3 види пошуку автомобілів за певними критеріями. Після вибору пункту користувач повинен буде ввести додаткові дані для пошуку авто за певними критеріями,після чого отримає наявну інформацію.

Якщо користувач вибере пункт "Вихід",то відбудеться вихід з програми.

# Висновок

В курсовій роботі було розроблено інформаційно-пошукову систему в інтегрованому середовищі Qt Creator. В системі реалізована робота з типованим файлом записів, який містить дані про автомобілі. Поля запису – модель (до 15 символів), номер (код регіону, цифровий код, буквений код), колір, дані про власника (прізвище, ім’я, по-батькові), дата останнього техогляду. В програмі реалізовано пошук даних за різними умовами, додання записів. Уся інформація виводиться у вигляді таблиць, що має зручне представлення. Інтерфейс програми представлено двома меню з 7 пунктами. Для вибору пункту меню використано оператор SWITCH, який вибирає потрібний на фрагмент коду, в залежності від значення змінної, у яку заздалегідь заноситься код натиснутої клавіші у системі ASCII, за допомогою функції getch(). Повернення до пунктів меню після виконання запиту реалізовано за допомогою виклику процедури потрібного меню після виконання запиту. Для розробки даної системи було використано мову C++.

Під час виконання курсової роботи було закріплено уміння та навички програмування на прикладі розробки інформаційно-пошукової системи для заданої предметної області.

# Список літератури

1. «Основи програмування мовою С++» авторів Є.П. Путятін, В.П. Степанов, В.П. Пчелінов, Т.Г. Долженкова, О.О. Матат
2. <http://rghost.ru/8rF8j4Yhl> - Методичні вказівки
3. <https://msdn.microsoft.com/ru-ru/default.aspx> - Мережа розробників Майкрософт.