Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

Механіко-технологічний факультет

Кафедра кібербезпеки та програмного забезпечення

ЗВІТ

ПРО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ № 11

з навчальної дисципліни “Базові методології та технології

програмування ”

**РЕАЛІЗАЦІЯ ПРОГРАМНИХ ЗАСОБІВ ОБРОБЛЕННЯ**

**ДИНАМІЧНИХ СТРУКТУР ДАНИХ ТА БІНАРНИХ ФАЙЛІВ**

ВИКОНАВ

студент академічної групи КІ-20

Небесний А.В.

ПЕРЕВІРИВ

викладач кафедри кібербезпеки

та програмного забезпечення

Л.І. Поліщук

Кропивницький – 2021

Мета роботи: полягає у набутті ґрунтовних вмінь і практичних навичок командної (колективної) реалізації програмного забезпечення, розроблення функцій оброблення динамічних структур даних, використання стандартних засобів С++ для керування динамічною пам’яттю та бінарними файловими потоками.

Завдання до лабораторної роботи:

1. У складі команди ІТ-проекта розробити програмні модулі оброблення динамічної структури даних.

2. Реалізувати програмний засіб на основі розроблених командою ІТ-проекта модулів.

Склад команди: Небесний Андрій, Рисований Максим, Кривохижа Віталій.

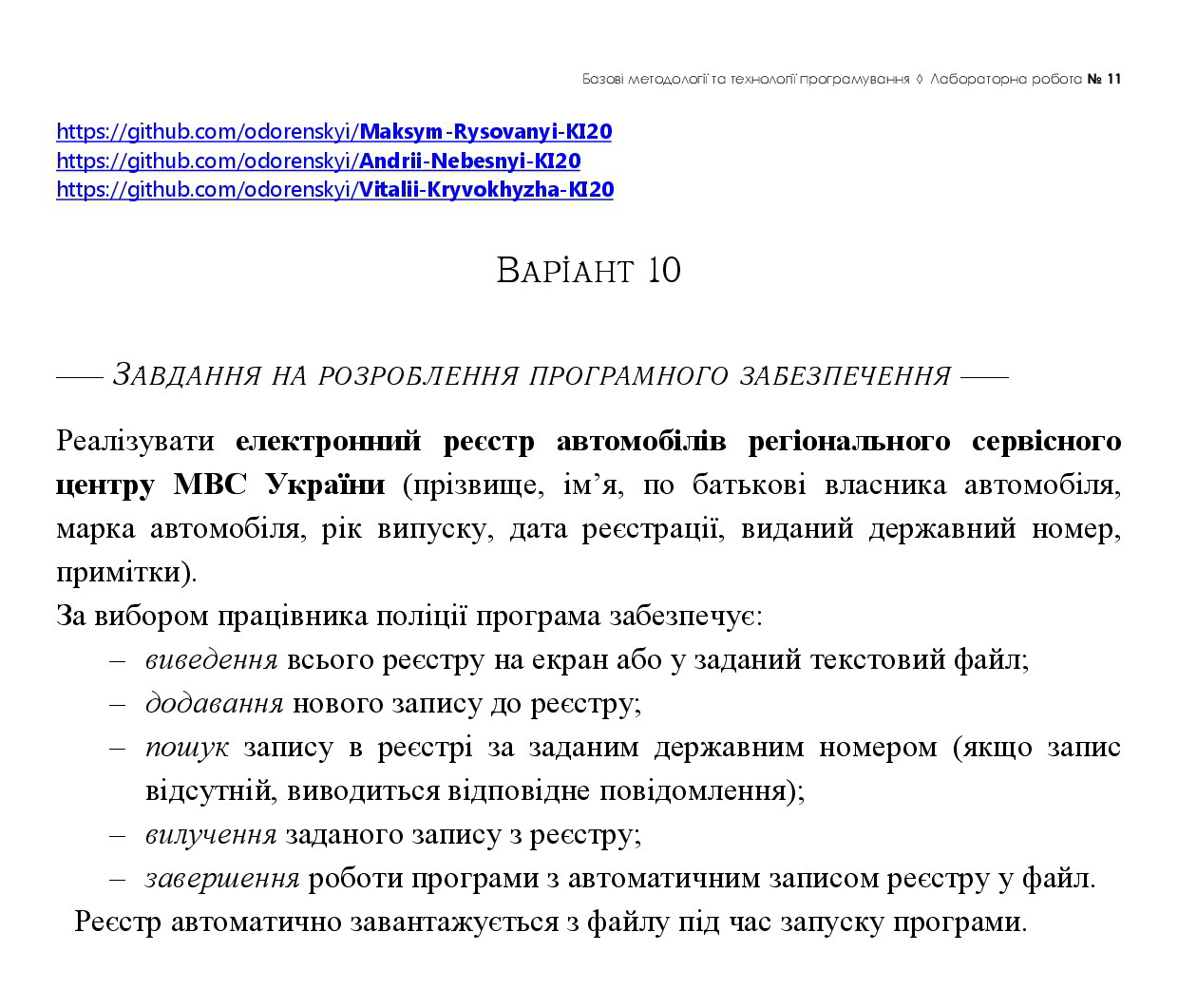


Рисунок 1 – Завдання за варіантом №10

Аналіз задачі IT-проекта:

**Вихідні змінні:** текст, який за допомогою функцій змінюється.

**Вхідні змінні:** база даних з трьох автомобілів, а саме прізвище, ім’я, по батькові власника автомобіля, марка автомобіля, рік випуску, дата реєстрації, виданий державний номер та примітки.

**Вимоги до ПЗ:**

1. Інтерфейс українською мовою.
2. Виконання валідації в усіх методах програми.
3. Список розробників і позначка авторського знаку (с).

**Алгоритм виконання ПЗ:**

1. У складі команди на мітингу виконання проектування програми, та розподіл праці між розробниками.
2. Написання функцій членами команди та тестування.
3. Написання інтерфейсу програми.
4. Повторне тестування, але вже з усіма функціями разом.
5. Усунення багів у програмі та написання звіту.

Задачу було поділено на **такі функції**:

1. greeting() – функція, у якій просто записано вітальне повідомлення, склад команди, та знак авторського права (с).
2. reg\_down() – функція, яка відкриває файл й завантажує усі данні у вектор структур.
3. reg\_reader() – функція, яка забезпечує вивід на екран увесь зміст вектора структур.
4. updater() – функція, яка активується при завершенні програми для запису інформації у файл.
5. nomers() – функція, для відображення усіх індексів автомобілів, які містяться у реєстрі.
6. new\_car() – функція для додавання нової машини у реєстр.
7. find\_car() – функція для пошуку автомобіля у реєстрі за автомобільним номером.
8. del\_car() – функція для видалення машини з реєстру, якщо такої машини не існує, відображається помилка про це.
9. У головній функції main реалізовано вибір функції через оператор switch.

Висновки

Під час цієї лабораторної роботи потрібно було поділитися на команди, щоб колективно реалізувати програмне забезпечення.

Склад нашої команди: Кривохижа Віталій, Небесний Андрій та Рисований Максим.

Зібравшись командою на мітинг у 501 кабінеті колективно ми виконали аналіз задач IT-проекта та вимог до програмного забезпечення, також обговорили специфікації ПЗ, сформулювали та узгодили архітектуру програмного засобу, основні алгоритми виконання та функціонування нашої програми, також узгодили макет інтерфейсу, який був ще не до кінця продуманий. Після цього, ми розподілили між собою усі підзадачі для реалізування програми відповідно до архітектури.

Розподіл праці між учасниками був такий:

- Рисований Максим: Додавання та видалення даних.

- Небесний Андрій: Обробка файлу та дизайн ПЗ.

- Кривохижа Віталій: Пошук даних та тестування ПЗ.

Для реалізації ПЗ було використано стэк та вектор, уся інформація записувалася у текстовий файл register.txt та виводилася у консоль, так як програму було поділено на функції, було вирішено, що через множинний оператор switch, буде зручніше викликати той чи інший метод.

Індивідуально кожний учасник виконував свою задачу та після написання частини ПЗ ми знову зібралися на мітинг та продемонстрували свою працю один одному, після коректування помилок та узгодження наступних дій у реалізації програмного забезпечення, кожен учасник знову отримав завдання та почав його виконання.

Після наступного зібрання команди, ми об’єднали наші функції у основну програму і після цього Небесний Андрій розробив дизайн ПЗ. Після успішної компіляції програми, було затверджено виконати тестування ПЗ Віталієм.

Після виконання тестування, було виправлено помилки у деяких функціях програми.

Усю лабораторну роботу було виконано за три дні, після чого був написаний звіт. Вихідний код програми було додано до звіту як додаток.

Також варто зауважати основні проблеми, які виникли під час реалізації програмного забезпечення:

1. По-перше, це не до кінця розуміння якою динамічною структурою даних потрібно було використовувати, але при вивченні додаткового матеріалу та перегляд прикладів коду в Інтернеті на IT-форумах, було затверджено, що ми будемо використовувати стек та вектор.
2. По-друге, так як ми вперше працювали у команді, не було спочатку зрозуміло, як саме розподілити завдання між учасниками, але після спільного написання початку програми стало все ясно, хто який метод буде розробляти.

Під час цієї лабораторної роботи стало зрозуміло, що виконання великого проекту у команді набагато простіше ніж власноруч, адже правильний розподіл сил дає прекрасний результат у кінці.

Наша команда вважає, що таких лабораторних робіт повинно буде більше, адже IT-світ це по-перше виконання роботи у команді, тому окрім hard скіллів, потрібно авжеж практикування soft скіллів, які у подальшій роботі дуже знадобляться.

Додаток А. Лістинг задачі:

#include <iostream>

#include <vector>

#include <fstream>

#include <windows.h>

#include <clocale>

#include <string>

using namespace std;

struct Car{

// даем пустое значение на старте с целью анигиляции ошибок

string nomer = " ";

string owner = " ";

string model\_year = " ";

string reg\_date =" ";

} obj; // объект структуры

vector <struct Car> Reg; // создаем вектор и указываем его структуру

char filename[] = "register.txt"; // имя файла

string line = ""; // вспомагательная переменная

ifstream reg\_in;

ofstream reg\_out;

void greeting(){

cout << " Вас вітає застосунок обробки реєстру авто МВС України." << endl

<< " --------------------------------------------------------------------- " << endl

<< "| Застосунок розроблено студентами ЦНТУ групи КІ-20: |" << endl

<< "| <<DEVELOPERS>> |" << endl

<< "| Рисований Максим Олегович - Додавання та видалення даних. |" << endl

<< "| Кривохижа Віталій Юрійович - Пошук даних й тестування |" << endl

<< "| Небесний Андрій Вадимович - Обробка файлу й дизайн ПЗ |" << endl

<< " -----------------------(c)AllRightsAreReserved----------------------- " << endl;

system("pause");

system("cls");

}

void reg\_down(){

// открывает файл и загружает все данные в вектор структур

//

reg\_in.open(filename, ios\_base::in);

int iter = 0;

while(!reg\_in.eof()){

getline(reg\_in, line);

if(line.length() > 1){ // проверяем строку на пустоту для уничтожения пустых структур

Reg.push\_back(obj);

Reg[iter].nomer = line;

getline(reg\_in, Reg[iter].owner);

getline(reg\_in, Reg[iter].model\_year);

getline(reg\_in, Reg[iter].reg\_date);

}

iter++;

}

reg\_in.close();

}

void reg\_reader(){

// выводит на экран все содержимое вектора структур

//

cout << "Ви обрали виведення реєстру на екран." << endl

<< "-----------------------------------------------" << endl;

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

cout << Reg[i].nomer << endl;

cout << Reg[i].owner << endl;

cout << Reg[i].model\_year << endl;

cout << Reg[i].reg\_date << endl;

}

cout << "-----------------------------------------------" << endl;

system("pause");

system("cls");

}

void updater(){

// функция активируется по завершению

//

reg\_out.open(filename, ios\_base::trunc | ios\_base::out);

for(int j = 0; j < Reg.size(); j++){

reg\_out << Reg[j].nomer << endl

<< Reg[j].owner << endl

<< Reg[j].model\_year << endl

<< Reg[j].reg\_date << endl;

}

reg\_out.close();

}

void nomers(){

cout << "В реєстрі містяться такі авто (1, 2, 3... - індекси):" << endl;

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

cout << i << ". " << Reg[i].nomer << endl;

}

cout << "------------------------------------------------------" << endl;

}

void new\_car(){

// эта функция добавляет новый элемент в вектор

//

Reg.push\_back(obj);

cout << "Ви обрали додавання нового автомобіля до реєстру." << endl

<< "-------------------------------------------------------" << endl;

cout << "Будь ласка, введіть номер нового автомобіля (AA0000NA)."

<< endl << ">> ";

getline(cin, line);

bool flag = true;

while(true){

getline(cin, line);

if(line.length() != 8){

cout << "Невірно введений номер, спробуйте ще раз! (AA0000NA)" << endl << ">> ";

flag = false;

}

else

break;

}

Reg[Reg.size() - 1].nomer = line;

cout << "Введіть ПІП власника. "

<< endl << ">> ";

getline(cin, line);

Reg[Reg.size() - 1].owner = " " + line;

cout << "Введіть модель та рік випуску авто. "

<< endl << ">> ";

getline(cin, line);

Reg[Reg.size() - 1].model\_year = " " + line;

cout << "Введіть дату реєстрації. "

<< endl << ">> ";

getline(cin, line);

Reg[Reg.size() - 1].reg\_date = " " + line;

cout << "-------------------------------------------------------" << endl;

system("pause");

system("cls");

}

int find\_car(){

// эта функция ищет элемент в структуре за автомобильным номером

// выводит сообщение о наличии или отсутствии элемента

// Коментруйте свой код

cout << "Ви обрали пошук авто за індексом в реєстрі." << endl;

cout << "-------------------------------------------------------" << endl;

nomers();

int find\_Elem;

cout << "Будь ласка, введіть індекс номеру автомобіля, який ви бажаєте знайти." << endl << ">> ";

cin >> find\_Elem;

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

{

if(find\_Elem == i)

{

cout << "Результати пошуку в реєстрі: " << endl;

cout << Reg[i].nomer << endl;

cout << Reg[i].owner << endl;

cout << Reg[i].model\_year << endl;

cout << Reg[i].reg\_date << endl;

break;

}

else if(find\_Elem >= Reg.size()){

cout << "Авто за даним індексом відсутнє." << endl;

break;

}

}

cout << "-------------------------------------------------------" << endl;

system("pause");

system("cls");

}

int del\_car(){

// ищет обьект за индексом автомобильного номера и стирает его

// если обькта не было в векторе, возвращает сообщение об ошибке

//

cout << "Ви обрали видалення автомобіля з реєстру." << endl

<< "-------------------------------------------------------" << endl;

nomers();

int del\_elem;

cout << "Будь ласка, введіть індекс номеру автомобіля, який ви бажаєте видалити." << endl << ">> ";

cin >> del\_elem;

if(del\_elem < Reg.size()){

Reg.erase(Reg.begin() + del\_elem);

cout << "Автомобіль було видалено з реєстру!" << endl;

}

else{

cout << "Помилка, спробуйте ще раз! (Можливо за даним індексом не існує автомобіля)" << endl;

}

cout << "-------------------------------------------------------" << endl;

system("pause");

system("cls");

}

int main(){

SetConsoleCP(::GetACP());;

SetConsoleOutputCP(::GetACP());;

char key = ' ';

greeting();

reg\_down();

while(true){

cout << "Список ключів:" << endl

<< "N - Додати автомобіль;" << endl

<< "S - Вивести на екран реєстр;" << endl

<< "D - Видалити авто з реєстру;" << endl

<< "F - Знайти авто в реєстрі" << endl

<< "E - Завершення роботи зі збереженням змін." << endl

<< "------------------------------------------------------" << endl;

if(Reg.size() == 0){

cout << "На даний момент реєстр пороженій, що ви бажаєте зробити?" << endl;

}else{

nomers();

cout << "Що ви бажаєте зробити?" << endl;

}

cout << ">> ";

cin >> key;

switch(key){

case 'S':

system("cls");

reg\_reader();

break;

case 'N':

system("cls");

new\_car();

break;

case 'D':

system("cls");

del\_car();

break;

case 'F':

system("cls");

find\_car();

break;

case 'E':

system("cls");

cout << "Збереження реєстру...." << endl;

updater();

system("pause");

return 0;

default:

cout << "Невірно введено дані, спробуйте інший ключ" << endl;

system("pause");

system("cls");

break;

}

}

}