**Міністерство освіти і науки України**

**Національний університет «Запорізька політехніка»**

Кафедра програмних засобів

**ЗВІТ**

з лабораторної роботи №6

дисципліна: Алгоритмізація та програмування

тема: **«Структури, перелiчення, об'єднання»**

Варіант 22

Виконав:

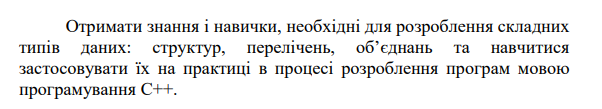
студентка гр. КНТ-210 Яценко А. А.

Перевірив:

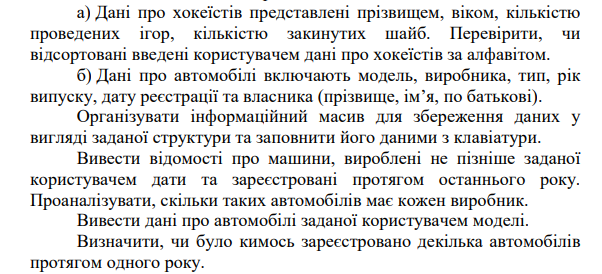
ст. викладач Качан О.І.

2020

**1.1 Мета роботи**



**1.2 Завдання:**



**1.4 Початкові коди програм**

#include <iostream>

#include <vector>

using namespace std;

struct Player

{

string surname;

int age;

int games\_count;

int goals;

};

int main()

{

int n;

cout << "Players count = ";

cin >> n;

cout << "Enter players in below format:" << endl;

cout << "Surname | Age | Games Count | Goals" << endl;

cout << "0. Petrov 22 100 606" << endl;

Player\* players = new Player[n];

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

{

cout << i+1 << ". ";

cin >> players[i].surname;

cin >> players[i].age;

cin >> players[i].games\_count;

cin >> players[i].goals;

}

for (int i = 1; i < n; i++)

{

string prev\_surname = players[i-1].surname;

string surname = players[i].surname;

//cout << prev\_surname << " " << surname << " " << (prev\_surname > surname);

if (prev\_surname > surname)

{

cout << "Error, " << i+1 << " line is not ordered(" << prev\_surname << ", " << surname << ")" << endl;

}

}

}

б)

#include <iostream>

#include <map>

using namespace std;

struct Date

{

int day;

int month;

int year;

string tostring()

{

string result;

if (day < 10)

{

result += '0';

}

result += to\_string(day);

result += '-';

if (month < 10)

{

result += '0';

}

result += to\_string(month);

result += '-';

result += to\_string(year);

return result;

}

};

enum carType {hatchback = 'H', universal = 'U', sedan = 'S'};

struct Car

{

string model;

string manufacturer;

carType type;

int year;

Date registrationDate;

string FIO;

};

/\* INPUT

5

Kalina Lada S 2008 11-12-2020 Kulikov Valerii Maksimovich

Benz Mercedes H 2015 05-10-2019 Grigorenko Alexey Andreevich

Kalina Lada S 2008 11-12-2020 Sergeev Alexey Victorovich

Kalina Lada H 2008 11-01-2021 Kaleno Georgiy Valerievich

Car BMW U 1999 01-12-2020 Kulikov Valerii Maksimovich

11-12-2020

Kalina

\*/

void print\_car(Car car)

{

cout << car.model << " | " << car.manufacturer << " | " << (char)car.type << " | " << car.year << " | "

<< car.registrationDate.tostring() << " | " << car.FIO << endl;

}

void print\_stats(map<string, int> stats)

{

cout << "~~~~~~Stats~~~~~~" << endl;

for ( const auto &p : stats ) {

cout << p.first << " - " << p.second << "\n";

}

}

Date str\_to\_date(string str)

{

Date date = Date();

date.day = stoi(str.substr(0,2));

date.month = stoi(str.substr(3,2));

date.year = stoi(str.substr(6,4));

return date;

}

int main()

{

int n;

cout << "Cars count = ";

cin >> n;

cout << "Enter players in below format:" << endl;

cout << "Model | Manufacturer | Type(H-hatchback, U-universal, S-sedan | Year | Registration Date | FIO" << endl;

cout << "0. Kalina Lada S 2008 11-12-2020 Kulikov Valerii Maksimovich" << endl;

Car\* cars = new Car[n];

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

{

cout << i+1 << ". ";

cin >> cars[i].model;

cin >> cars[i].manufacturer;

char type;

cin >> type;

switch (tolower(type))

{

case 'h':

cars[i].type = hatchback;

break;

case 'u':

cars[i].type = universal;

break;

case 's':

cars[i].type = sedan;

break;

}

cin >> cars[i].year;

string reg\_date;

cin >> reg\_date;

cars[i].registrationDate = str\_to\_date(reg\_date);

string F, I, O;

cin >> F >> I >> O;

cars[i].FIO = F + " " + I + " " + O;

}

cout << "Enter date in this format: 31-12-2020(dd-mm-yyyy)\n";

string str\_date;

cin >> str\_date;

Date date = str\_to\_date(str\_date);

map<string, int> manu\_stats;

cout << "~~~~~~Found Cars~~~~~~" << endl;

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

{

if ((cars[i].registrationDate.year <= date.year && cars[i].registrationDate.month <= date.month && cars[i].registrationDate.day <= date.day) || cars[i].registrationDate.year == 2021)

{

print\_car(cars[i]);

if (manu\_stats.count(cars[i].manufacturer) == 0)

{

manu\_stats.insert(pair<string,int>(cars[i].manufacturer, 1));

}

else

{

manu\_stats[cars[i].manufacturer]++;

}

}

}

print\_stats(manu\_stats);

string model;

cout << "Enter model: ";

cin >> model;

cout << "~~~~~~Found Cars~~~~~~" << endl;

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

{

if (cars[i].model == model)

{

print\_car(cars[i]);

}

}

cout << "Registered cars in one year by one human? ";

bool isRegistered = false;

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

{

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

{

if (cars[i].FIO == cars[j].FIO && cars[i].registrationDate.year == cars[j].registrationDate.year)

{

isRegistered = true;

break;

}

}

}

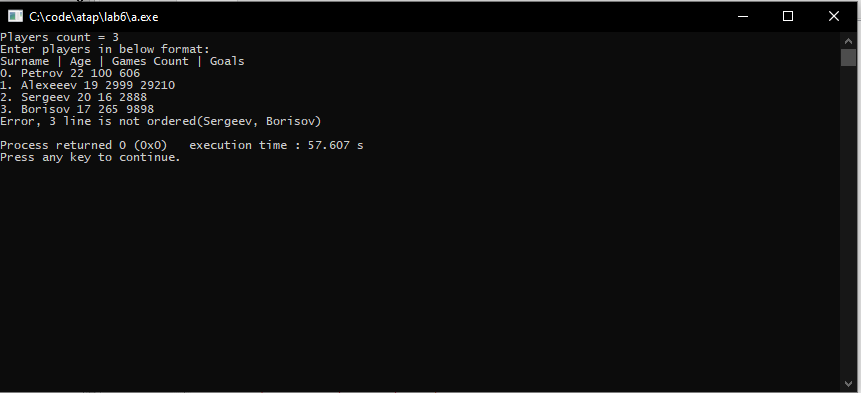
cout << (isRegistered ? "YES" : "NO") << endl;

}

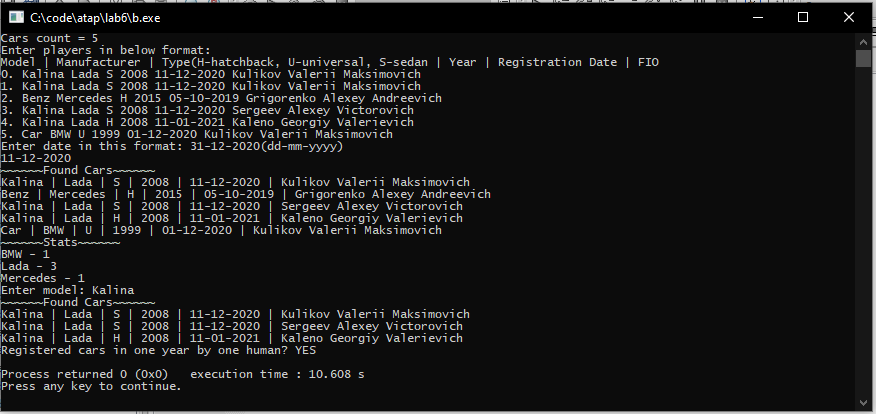
**1.5 Результат роботи програм**

Скриншоти

а)



б)



**1.7 Висновки**

Отримав знання і навички, необхідні для програмування з викроистанням cтруктур, перелiченнь, об'єднаннь, та навчився використовувати їх на практиці в процесі розроблення програм мовою програмування С++.