Міністерство освіти і науки України

Національний лісотехнічний університет України

Кафедра інформаційних технологій

**Звіт до лабораторної роботи №2**

з навчальної дисципліни

**«Об’ктно-орієнтоване програмування»**

на тему:

«РОБОТА З КЛАСАМИ, МАСИВИ ОБ’ЄКТІВ»

**Виконав:**

студент групи КНС-11

Ватаманюк Р.

**Перевірив:**

          Волинець Є.

Львів – 2021

Мета роботи: освоїти технологію створення програм в консолі з використанням класів, розв’язувати поставлені задачі з масивами об’єктів.

Завдання: Розробити класи для описаних нижче об’єктів. Передбачити в класах методи ***set***(), ***get***(), ***show***() та визначити інші методи для виконання індивідуального завдання.

Варіант 8

Створити клас АВТОМОБІЛЬ, який містить об’єкт класу ДВИГУН. У класі ДВИГУН визначено дані про об’єм двигуна, потужність, заводський номер. Клас АВТОМОБІЛЬ додатково містить марку, колір та номер державної реєстрації. Визначити конструктори без параметрів і з різним числом параметрів, деструктор, методи для роботи з полями даних. Визначити методи перепризначення номера і кольору.

Код програми:

OOP lab2.cpp

#include "CarManager.h"

int stateMenu;

void Menu()

{

cout << "Зробiть вибiр" << endl

<< "(1) заповнити машини" << endl

<< "(2) вивести машини" << endl

<< "(3) додати нову машину" << endl

<< "(4) видалити машину" << endl

<< "(0) для того, щоб вийти з програми"

<< "Ваш вибiр: ";

cin >> stateMenu;

}

int main()

{

setlocale(LC\_ALL, "Ukrainian");

cout << "Введiть кiлькiсть машин: ";

carManager allCars;

cout << endl;

Menu();

while (stateMenu != 0)

{

switch (stateMenu)

{

case 1:

allCars.fillCars();

cout << endl;

Menu();

break;

case 2:

allCars.Show();

cout << endl;

Menu();

break;

case 3:

allCars.AddCars1();

cout << endl;

Menu();

break;

case 4:

allCars.deleteCars1();

cout << endl;

Menu();

break;

default:

cout << "Ви ввелт невiрне значення. Повторiть спробу";

break;

}

}

}

CarManager.h

#pragma once

#include <Windows.h>

#include "CarIO.h"

class carManager

{

int carCount = ConsoleIO::GetValue();

Car\* cars = new Car[carCount];

CarIO\* IOWorker = new CarIO();

public:

carManager()

{

}

~carManager(void)

{

delete[] cars;

delete IOWorker;

}

void fillCars()

{

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

{

IOWorker->Input(cars[i]);

}

}

void Show()

{

cout << endl;

cout << "Список усiх машин" << endl;

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

{

IOWorker->Output(cars[i]);

}

}

void AddCars1()

{

AddCars(cars, carCount);

}

void AddCars(Car \*&cars, int &carCount)

{

Car\* newCars = new Car[carCount + 1];

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

{

newCars[i] = cars[i];

}

IOWorker->Input(newCars[carCount]);

carCount++;

delete[] cars;

cars = newCars;

}

void deleteCars1()

{

deleteCars(cars, carCount);

}

void deleteCars(Car\*& cars, int& carCount)

{

Car\* newCars = new Car[carCount - 1];

for (int i = 0; i < carCount - 1; i++)

{

newCars[i] = cars[i];

}

carCount--;

delete[] cars;

cars = newCars;

}

};

CarIO.h

#pragma once

#include "Car.h"

#include "ConsoleIO.h"

using namespace std;

class CarIO

{

public:

void Input(Car& car)

{

cout << endl;

cout << "Машина " << endl;

cout << "Введiть марку ";

car.setCarMark(ConsoleIO::GetText());

cout << "Введiть колiр ";

car.setCarColor(ConsoleIO::GetText());

cout << "Введiть номер державної регiстрацiї ";

car.setCarRegistration(ConsoleIO::GetValue());

cout << "Двигун " << endl;

cout << "Введiть об'м ";

car.setEngineCapacity(ConsoleIO::GetValue());

cout << "Введiть потужнiсть ";

car.setEnginePover(ConsoleIO::GetValue());

cout << "Введiть заводський номер ";

car.setEngineSerialNumber(ConsoleIO::GetValue());

}

void Output(Car& car)

{

cout << endl;

cout << "Машина "<< endl;

cout << "Марка " << car.getCarMark() << endl;

cout << "Колiр " << car.getCarColor() << endl;

cout << "Номер державної регiстрацiї " << car.getCarRegistration() << endl;

cout << "Двигун" << endl;

cout << "Об'єм " << car.getEngineCapacity() << endl;

cout << "Потужнiсть " << car.getEnginePover() << endl;

cout << "Заводський номер " << car.getEngineSerialNumber() << endl;

cout << endl;

}

};

Car.h

#pragma once

#include <iostream>

#include <string>

using namespace std;

class Engine

{

float engineCapacity;

float enginePover;

float engineSerialNumber;

public:

//Constructors

Engine()

{

engineCapacity = 0;

enginePover = 0;

engineSerialNumber = 0;

}

//Get and set for field of Engine

float getEngineCapacity()

{

return engineCapacity;

}

void setEngineCapacity(float engineCapacity)

{

this->engineCapacity = engineCapacity;

}

float getEnginePover()

{

return enginePover;

}

void setEnginePover(float enginePover)

{

this->enginePover = enginePover;

}

float getEngineSerialNumber()

{

return engineSerialNumber;

}

void setEngineSerialNumber(float engineSerialNumber)

{

this->engineSerialNumber = engineSerialNumber;

}

};

class Car

{

string carMark;

string carColor;

int carStateRegistration;

Engine engine;

public:

//Constructors

Car()

{

carStateRegistration = 0;

}

//Get and set for fields of Car

string getCarMark()

{

return carMark;

}

void setCarMark(string carMark)

{

this->carMark = carMark;

}

string getCarColor()

{

return carColor;

}

void setCarColor(string carColor)

{

this->carColor = carColor;

}

int getCarRegistration()

{

return carStateRegistration;

}

void setCarRegistration(int carStateRegistration)

{

this->carStateRegistration = carStateRegistration;

}

//Get and set for field of Engine

float getEngineCapacity()

{

return engine.getEngineCapacity();

}

void setEngineCapacity(float engineCapacity)

{

engine.setEngineCapacity(engineCapacity);

}

float getEnginePover()

{

return engine.getEnginePover();

}

void setEnginePover(float enginePover)

{

engine.setEnginePover(enginePover);

}

float getEngineSerialNumber()

{

return engine.getEngineSerialNumber();

}

void setEngineSerialNumber(float engineSerialNumber)

{

engine.setEngineSerialNumber(engineSerialNumber);

}

};

ConsoleIO.h

#pragma once

#include <iostream>

#include <Windows.h>

using namespace std;

class ConsoleIO

{

public:

static string GetText()

{

while (true)

{

string value;

cin >> value;

if (value != " ")

{

return value;

break;

}

cout << "Некоректний ввiд. Повторіть спробу";

}

}

static int GetValue()

{

int value;

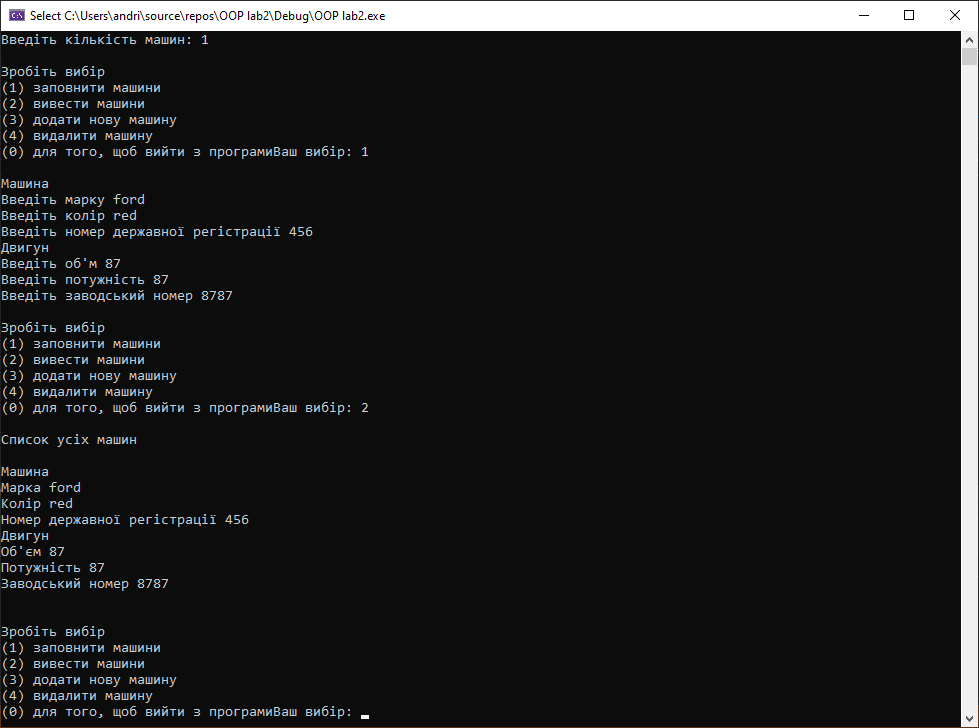
cin >> value;

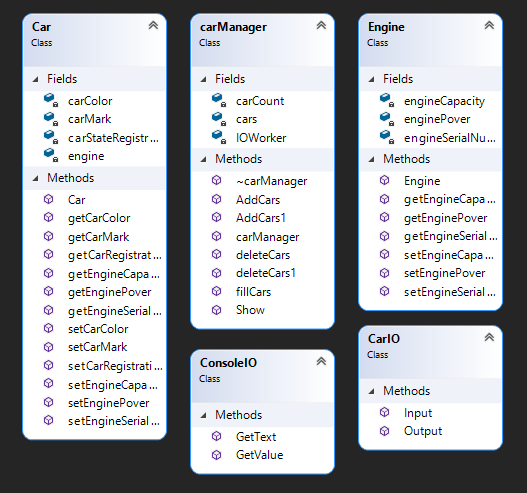
return value;

}

};

Результати воконання:





Висновок: освоїв технологію створення програм в консолі з використанням класів, розв’язав поставлені задачі з масивами об’єктів.