Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кафедра ЕОМ



**Звіт**

З лабораторної роботи № 4

З дисципліни «Об’єктно-орієнтоване програмування»

**На тему**: «Класи та об’єкти»

**Виконав:** ст. гр. КІ-12

Кузьма І.В.

**Прийняла:** доцент

Пастернак І.І.

Львів 2020

**Мета:** познайомитися із класами та об‘єктами.

**Теоретичні відомості**

Клас – це визначений користувачем тип з даними-елементами (властивостями) та функціями-елементами (методами), що є членами класу. Він зазвичай описує певну абстракцію якоїсь сутності реального світу з її властивостями і можливими діями. Змінна типу клас називається об‘єктом.

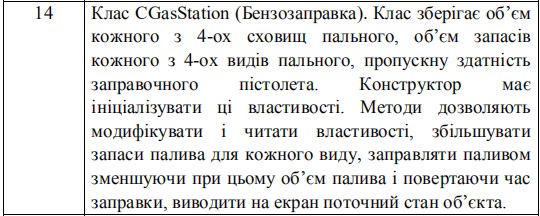
Об‘єкт – це вже не абстракція, а елемент реального світу, який може володіти певними характеристиками, які задаються властивостями в класі, та діяти згідно операцій заданих методами у класі. Рівні доступу членів классу. Рівні доступу членів класу є такими:

|  |  |  |  |
| --- | --- | --- | --- |
| **Доступ** | **private** | **protected** | **public** |
| Інші члени цього класу | **+** | **+** | **+** |
| Друзі класу | **+** | **+** | **+** |
| Члени похідних класів | **-** | **+** | **+** |
| З-під об‘єктів класу | **-** | **-** | **+** |

При створенні об'єкта класу автоматично викликається спеціальний метод, що зветься конструктор. Конструктор – це метод класу ім‘я якого співпадає з іменем класу та не повертає ніяких значень (навіть void). Основне призначення конструктора – початкова ініціалізація об‘єкту класу. Деструктор – це спеціальна функція-елемент класу, яка викликається при знищені об‘єкту і не приймає параметрів та не повертає значень. Знищення об'єкту може відбуватися, наприклад, коли виконувана програма залишає область дії, у якій був створений об'єкт цього класу, або явно здійснюється знищення динамічно створеного об‘єкту. Деструктор сам не знищує об'єкт, а лише виконує підготовку до знищення об‘єкту (вивільняє пам‘ять від динамічно створених даних-елементів об‘єкту, закриває потоки обміну даними,…) перед тим, як система звільняє область пам'яті, у якій зберігався об'єкт, щоб використовувати її для розміщення нових об'єктів. Ім'я деструктора складається з символ тильда (~) та імені класу. Клас може мати тільки один деструктор – перевантаження деструктора забороняється. Якщо деструктор не визначений явно, то він буде згенерований автоматично і при виклику нічого не робитиме. Деструктор доцільно оголошувати написавши перед ним ключове слово virtual, яке забезпечує коректність функціонування класу при спадкуванні.

**Завдання:** Спроектувати клас, що реалізує предметну область згідно варіанту, та розробити програму-драйвер (main-функцію), яка демонструє роботу класу, а

саме: можливі способи створення об‘єктів класів і коректність функціонування розроблених методів класу (запустити кожен метод класу хоча б 2 рази).



**Код програми:**

**Header**

#pragma once

#include <iostream>

#include <iomanip>

using namespace std;

class CGasStation

{

private:

struct TypeGas

{

string Gas\_\_Name;

int Amount, MaxAmount;

};

int MaxLitres, Amout\_\_Litres;

double Capacity;

string GasNames[4] = { "Diesel","Gasoline","Methanol","Propane" };

TypeGas Dgases[4];

void Initialize()

{

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

{

Dgases[i].Gas\_\_Name = GasNames[i];

Dgases[i].Amount = Amout\_\_Litres;

Dgases[i].MaxAmount = MaxLitres;

}

}

public:

CGasStation()

{

MaxLitres = 1500;

Amout\_\_Litres = 750;

Capacity = 0.5;

Initialize();

}

~CGasStation()

{

If()

}

CGasStation(int MaxLitres, int Amout\_\_Litres, double Capacity)

{

this->MaxLitres = MaxLitres;

this->Amout\_\_Litres = Amout\_\_Litres;

this->Capacity = Capacity;

Initialize();

}

void Information()

{

cout << setw(15) << "Gas name" << setw(15) << "Now litres" << setw(15) << "Max Litres" << endl;

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

{

cout << setw(15) << Dgases[i].Gas\_\_Name << setw(15) << Dgases[i].Amount << setw(15) << Dgases[i].MaxAmount << endl;

}

}

void AddAmount(int ZAmount, int Where)

{

if (ZAmount+Dgases[Where].Amount< Dgases[Where].MaxAmount)

{

Dgases[Where].Amount += ZAmount;

}

else

{

cout << "Maximum";

Dgases[Where].Amount = Dgases[Where].MaxAmount;

}

}

void Refuil(int Litres, int Where)

{

double time;

if (Dgases[Where].Amount- Litres >0)

{

time = Litres / Capacity;

Dgases[Where].Amount -= Litres;

cout << "It will took: " << time << " secunds"<<endl;

}

else

{

cout << "Not enough gas" << endl;

}

}

};

**Main**

#include <iostream>

#include "Gas.h"

using namespace std;

int main()

{

int choose, amount, where;

CGasStation GasStation;

do

{

cout << "Information(0)\nAdd amount fo gas(1)\nRefuil(3)\nExit(4)\n";

cin >> choose;

switch (choose)

{

case 0:

system("cls");

GasStation.Information();

break;

case 1:

system("cls");

cout << "Enter amount of gas: ";

cin >> amount;

cout << "Diesel(0)\nGasoline(1)\nMethanol(2)\nPropane(3)\nWhere you want to add: ";

cin >> where;

GasStation.AddAmount(amount, where);

break;

case 3:

system("cls");

cout << "Enter amount of gas you want to refuil: ";

cin >> amount;

cout << "Diesel(0)\nGasoline(1)\nMethanol(2)\nPropane(3)\nWhere you want to add: ";

cin >> where;

GasStation.Refuil(amount, where);

break;

case 4:

system("cls");

return 0;

default:

system("cls");

cout << "Enter corect!!!" << endl;

continue;

}

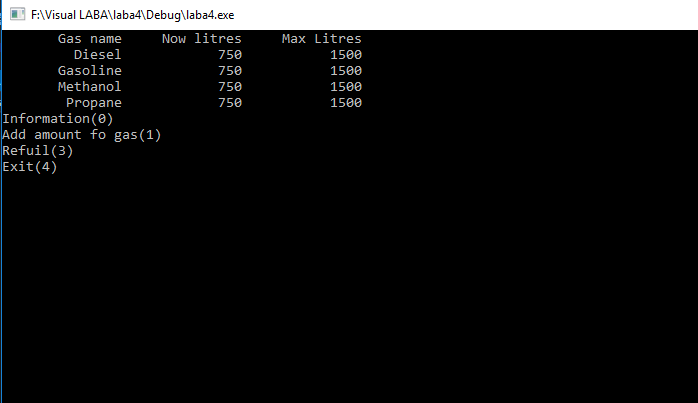
} while (true);

system("pause");

return 0;

}

**Результат програми:**



**Висновок:** Під час виконання цієї лабораторної роботи я познайомився і навчився застосовувати класи та об‘єкти.