

Міністерство освіти і науки України
Національний університет «Львівська політехніка»
Кафедра ЕОМ



Звіт
до лабораторної роботи № 4
з дисципліни: «Програмування, частина 2 (ООП)»
на тему: «Класи Та Об'єкти»
Варіант № 14

Підготував: Мишак М.А.
студент групи КІ-103
Перевірив:
Ст. викладач
Каф. ЕОМ
Гузинець Н.В

Мета: познайомитися із класами та об'єктами.

Завдання: Клас CGasStation (Бензозаправка). Клас зберігає об'єм кожного з 4-ох сховищ пального, об'єм запасів кожного з 4-ох видів пального, пропускну здатність заправочного пістолета. Конструктор має ініціалізувати ці властивості. Методи дозволяють модифікувати і читати властивості, збільшувати запаси палива для кожного виду, заправляти паливом зменшуючи при цьому об'єм палива і повертаючи час заправки, виводити на екран поточний стан об'єкта.

Теоретичний матеріал

Основна відмінність будь-якої об'єктно-орієнтованої мови програмування від інших не об'єктно-орієнтованих мов програмування є можливість створення нових визначених користувачем типів, що називаються класами. Клас – це визначений користувачем тип з даними-елементами (властивостями) та функціями-елементами (методами), що являються 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.

Члени класу (властивості і методи), оголошені після ключового слова `public` стають відкритими членами. Це означає, що вони доступні:

- усім іншим членам класу;
- дружнім конструкціям (класам, методам, функціям);
- членам похідних класів;
- з-під об'єктів класу після створення об'єктів; після створення об'єкту класу до його членів можна звертатися лише якщо вони є загальнодоступними.

Якщо в класі не оголошено секцій, але визначено методи чи властивості, то вони вважатимуться такими, що оголошені у секції `private`. Для модифікації властивостей, що оголошені в секції `private` створюються спеціальні методи, які здійснюють цю модифікацію. Назви таких методів прийнято починати зі слова `set`. На жаргоні такі методи інколи називають сетерами. Для читання властивостей, що оголошені в секції `private` створюються спеціальні методи, які безпечно повертають значення цих властивостей. Назви таких методів прийнято починати зі слова `get`. На жаргоні такі методи інколи називають гетерами.

При визначенні методу класу за межами класу необхідно зв'язати ім'я відповідного методу з ім'ям класу. Зв'язок встановлюється шляхом написання імені класу, оператора розширення області видимості (`::`) та імені методу.

Оскільки різні класи можуть мати елементи з однаковими іменами, то оператор розширення області видимості дозволяє однозначно ідентифікувати методи конкретного класу.

Статичні дані-елементи (властивості) оголошуються в класі, а ініціалізуються за межами класу (не у конструкторах). На відміну від інших властивостей вони належать класу, а не об'єкту. Тобто вони єдині для всіх об'єктів певного класу і зміни одним об'єктом вплинуть на всі об'єкти даного 50 класу. Якщо статичні властивості оголошені в секції `public`, то вони будуть завжди доступні незалежно від того чи створено хоч один об'єкт класу. За допомогою статичних властивостей зручно реалізовувати лічильники, наприклад, лічильники створених 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. У класі може бути визначено кілька перевантажених конструкторів, що відрізняються списком параметрів. При створенні об'єкта викликатися буде тільки один з них. Який саме – визначається зі способу створення об'єкта. Якщо конструктор має один параметр, то він може бути використаним для операції приведення типів. Якщо треба унеможливити використання конструктора для операції приведення типів, то його слід оголосити з використанням ключового слова `explicit`.

Деструктор – це спеціальна функція-елемент класу, яка викликається при знищенні 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.

Кожен об'єкт у мові C++ містить спеціальний вказівник `this`, який містить адресу об'єкту якому він належить. Крім цього цей вказівник неявно автоматично передається будь-якому нестатичному методу класу при його виклику, вказуючи цим самим на об'єкт з-під якого метод був викликаний. Як і будь-який вказівник у 32-ох бітній системі він займає 4 байти в пам'яті.

Угорська нотація - метод найменування змінних в програмуванні, який отримав найбільшого поширення у 90-х роках у програмістів компанії 64 Microsoft у яких вона була внутрішнім стандартом. Угорська нотація полягає в тому, що до ідентифікатора змінної або функції додається префікс, що вказує на його тип.

Лістинги (тексти) програм:

GasStation.h:

```
#include <iostream>
```

```
#include <string>
```

```
using namespace std;
```

```
class CGasStation {
```

```
private:
```

```
    double dieselVolume;
```

```
    double gasolineVolume;
```

```
    double lpgVolume;
```

```
    double electricVolume;
```

```
    double dieselReserve;
```

```
    double gasolineReserve;
```

```
    double lpgReserve;
```

```
    double electricReserve;
```

```
    double nozzleCapacity;
```

```
public:
```

```
    CGasStation(double dieselVolume = 1000, double gasolineVolume = 1000, double lpgVolume = 1000, double electricVolume = 1000,
```

```
                double dieselReserve = 500, double gasolineReserve = 500, double lpgReserve = 500, double electricReserve = 500,
```

```
                double nozzleCapacity = 50);
```

```
    void get_info() const;
```

```
    void setDieselVolume(double volume);
```

```
    double getDieselVolume() const;
```

```
    void setGasolineVolume(double volume);
```

```
    double getGasolineVolume() const;
```

```
    void setLpgVolume(double volume);
```

```
    double getLpgVolume() const;
```

```
    void setElectricVolume(double volume);
```

```
    double getElectricVolume() const;
```

```
    void setDieselReserve(double reserve);
```

```
    double getDieselReserve() const;
```

```
void setGasolineReserve(double reserve);  
double getGasolineReserve() const;
```

```
void setLpgReserve(double reserve);  
double getLpgReserve() const;
```

```
void setElectricReserve(double reserve);  
double getElectricReserve() const;
```

```
void setNozzleCapacity(double capacity);  
double getNozzleCapacity() const;
```

```
void increaseDieselReserve(double amount);  
void increaseGasolineReserve(double amount);  
void increaseLpgReserve(double amount);  
void increaseElectricReserve(double amount);
```

```
double refuelDiesel(double amount);  
double refuelGasoline(double amount);  
double refuelLpg(double amount);  
double refuelElectric(double amount);
```

```
};
```

GasStation.cpp:

```
#include "GasStation.h"
```

```
CGasStation::CGasStation(double dieselVolume, double gasolineVolume, double lpgVolume, double  
electricVolume,
```

```
    double dieselReserve, double gasolineReserve, double lpgReserve, double electricReserve,  
    double nozzleCapacity)
```

```
    : dieselVolume(dieselVolume), gasolineVolume(gasolineVolume), lpgVolume(lpgVolume),  
    electricVolume(electricVolume),
```

```
    dieselReserve(dieselReserve), gasolineReserve(gasolineReserve), lpgReserve(lpgReserve),  
    electricReserve(electricReserve),
```

```
    nozzleCapacity(nozzleCapacity) {
```

```
}
```

```
void CGasStation::get_info() const {
```

```
    cout << "Об'єм дизельного палива: " << dieselVolume << " л" << endl
```

```

    << "Об'єм бензину: " << gasolineVolume << " л" << endl
    << "Об'єм LPG: " << lpgVolume << " л" << endl
    << "Об'єм електроенергії: " << electricVolume << " кВт-год" << endl
    << "Запас дизельного палива: " << dieselReserve << " л" << endl
    << "Запас бензину: " << gasolineReserve << " л" << endl
    << "Запас LPG: " << lpgReserve << " л" << endl
    << "Запас електроенергії: " << electricReserve << " кВт-год" << endl
    << "Пропускна здатність заправочного пістолета: " << nozzleCapacity << " л/сек" << endl;
}

```

```

void CGasStation::setDieselVolume(double volume) { dieselVolume = volume; }
double CGasStation::getDieselVolume() const { return dieselVolume; }

```

```

void CGasStation::setGasolineVolume(double volume) { gasolineVolume = volume; }
double CGasStation::getGasolineVolume() const { return gasolineVolume; }

```

```

void CGasStation::setLpgVolume(double volume) { lpgVolume = volume; }
double CGasStation::getLpgVolume() const { return lpgVolume; }

```

```

void CGasStation::setElectricVolume(double volume) { electricVolume = volume; }
double CGasStation::getElectricVolume() const { return electricVolume; }

```

```

void CGasStation::setDieselReserve(double reserve) { dieselReserve = reserve; }
double CGasStation::getDieselReserve() const { return dieselReserve; }

```

```

void CGasStation::setGasolineReserve(double reserve) { gasolineReserve = reserve; }
double CGasStation::getGasolineReserve() const { return gasolineReserve; }

```

```

void CGasStation::setLpgReserve(double reserve) { lpgReserve = reserve; }
double CGasStation::getLpgReserve() const { return lpgReserve; }

```

```

void CGasStation::setElectricReserve(double reserve) { electricReserve = reserve; }
double CGasStation::getElectricReserve() const { return electricReserve; }

```

```

void CGasStation::setNozzleCapacity(double capacity) { nozzleCapacity = capacity; }
double CGasStation::getNozzleCapacity() const { return nozzleCapacity; }

```

```

void CGasStation::increaseDieselReserve(double amount) {

```

```

    if (dieselReserve + amount <= dieselVolume) {
        dieselReserve += amount;
    }
}

void CGasStation::increaseGasolineReserve(double amount) {
    if (gasolineReserve + amount <= gasolineVolume) {
        gasolineReserve += amount;
    }
}

void CGasStation::increaseLpgReserve(double amount) {
    if (lpgReserve + amount <= lpgVolume) {
        lpgReserve += amount;
    }
}

void CGasStation::increaseElectricReserve(double amount) {
    if (electricReserve + amount <= electricVolume) {
        electricReserve += amount;
    }
}

double CGasStation::refuelDiesel(double amount) {
    if (amount <= dieselReserve) {
        dieselReserve -= amount;
        return amount / nozzleCapacity;
    }
    return -1;
}

double CGasStation::refuelGasoline(double amount) {
    if (amount <= gasolineReserve) {
        gasolineReserve -= amount;
        return amount / nozzleCapacity;
    }
    return -1;
}

```

```
double CGasStation::refuelLpg(double amount) {
    if (amount <= lpgReserve) {
        lpgReserve -= amount;
        return amount / nozzleCapacity;
    }
    return -1;
}
```

```
double CGasStation::refuelElectric(double amount) {
    if (amount <= electricReserve) {
        electricReserve -= amount;
        return amount / nozzleCapacity;
    }
    return -1;
}
```

main.cpp:

```
#include <iostream>
```

```
#include <string>
```

```
#include "GasStation.h"
```

```
using namespace std;
```

```
int main() {
    setlocale(LC_ALL, "ukr");
```

```
    CGasStation station(2000, 2000, 1500, 1000, 1000, 1000, 750, 500, 60);
```

```
    while (true) {
        cout << endl;
        cout << "        Gas Station" << endl;
        cout << "1 - Переглянути загальну інформацію про заправку" << endl;
        cout << "2 - Встановити об'єм дизельного палива" << endl;
        cout << "3 - Переглянути об'єм дизельного палива" << endl;
        cout << "4 - Встановити об'єм бензину" << endl;
        cout << "5 - Переглянути об'єм бензину" << endl;
        cout << "6 - Встановити об'єм LPG" << endl;
        cout << "7 - Переглянути об'єм LPG" << endl;
        cout << "8 - Встановити об'єм електроенергії" << endl;
        cout << "9 - Переглянути об'єм електроенергії" << endl;
        cout << "10 - Встановити запас дизельного палива" << endl;
```



```

cout << "11 - Переглянути запас дизельного палива" << endl;
cout << "12 - Встановити запас бензину" << endl;
cout << "13 - Переглянути запас бензину" << endl;
cout << "14 - Встановити запас LPG" << endl;
cout << "15 - Переглянути запас LPG" << endl;
cout << "16 - Встановити запас електроенергії" << endl;
cout << "17 - Переглянути запас електроенергії" << endl;
cout << "18 - Встановити пропускну здатність заправочного пістолета" << endl;
cout << "19 - Переглянути пропускну здатність заправочного пістолета" << endl;
cout << "20 - Збільшити запас дизельного палива" << endl;
cout << "21 - Збільшити запас бензину" << endl;
cout << "22 - Збільшити запас LPG" << endl;
cout << "23 - Збільшити запас електроенергії" << endl;
cout << "24 - Заправка дизельним паливом" << endl;
cout << "25 - Заправка бензином" << endl;
cout << "26 - Заправка LPG" << endl;
cout << "27 - Заправка електроенергією" << endl;
cout << "0 - Вихід" << endl;
cout << "Введіть вибір: ";

int choice;
cin >> choice;
switch (choice) {
case 1: station.get_info(); break;
case 2: {
    double volume;
    cout << "Введіть об'єм дизельного палива: ";
    cin >> volume;
    station.setDieselVolume(volume);
    break;
}
case 3: cout << "Об'єм дизельного палива: " << station.getDieselVolume() << " л" << endl;
break;
case 4: {
    double volume;
    cout << "Введіть об'єм бензину: ";
    cin >> volume;
    station.setGasolineVolume(volume);
    break;
}
}

```

```
case 5: cout << "Об'єм бензину: " << station.getGasolineVolume() << " л" << endl; break;
case 6: {
    double volume;
    cout << "Введіть об'єм LPG: ";
    cin >> volume;
    station.setLpgVolume(volume);
    break;
}
case 7: cout << "Об'єм LPG: " << station.getLpgVolume() << " л" << endl; break;
case 8: {
    double volume;
    cout << "Введіть об'єм електроенергії: ";
    cin >> volume;
    station.setElectricVolume(volume);
    break;
}
case 9: cout << "Об'єм електроенергії: " << station.getElectricVolume() << " кВт-год" <<
endl; break;
case 10: {
    double reserve;
    cout << "Введіть запас дизельного палива: ";
    cin >> reserve;
    station.setDieselReserve(reserve);
    break;
}
case 11: cout << "Запас дизельного палива: " << station.getDieselReserve() << " л" << endl;
break;
case 12: {
    double reserve;
    cout << "Введіть запас бензину: ";
    cin >> reserve;
    station.setGasolineReserve(reserve);
    break;
}
case 13: cout << "Запас бензину: " << station.getGasolineReserve() << " л" << endl; break;
case 14: {
    double reserve;
    cout << "Введіть запас LPG: ";
    cin >> reserve;
    station.setLpgReserve(reserve);
    break;
```

```

    }
    case 15: cout << "Запас LPG: " << station.getLpgReserve() << " л" << endl; break;
    case 16: {
        double reserve;
        cout << "Введіть запас електроенергії: ";
        cin >> reserve;
        station.setElectricReserve(reserve);
        break;
    }
    case 17: cout << "Запас електроенергії: " << station.getElectricReserve() << " кВт-год" <<
endl; break;
    case 18: {
        double capacity;
        cout << "Введіть пропускну здатність заправочного пістолета: ";
        cin >> capacity;
        station.setNozzleCapacity(capacity);
        break;
    }
    case 19: cout << "Пропускна здатність заправочного пістолета: " <<
station.getNozzleCapacity() << " л/сек" << endl; break;
    case 20: {
        double amount;
        cout << "Введіть кількість дизельного палива для збільшення: ";
        cin >> amount;
        station.increaseDieselReserve(amount);
        break;
    }
    case 21: {
        double amount;
        cout << "Введіть кількість бензину для збільшення: ";
        cin >> amount;
        station.increaseGasolineReserve(amount);
        break;
    }
    case 22: {
        double amount;
        cout << "Введіть кількість LPG для збільшення: ";
        cin >> amount;
        station.increaseLpgReserve(amount);
        break;
    }

```

```

case 23: {
    double amount;
    cout << "Введіть кількість електроенергії для збільшення: ";
    cin >> amount;
    station.increaseElectricReserve(amount);
    break;
}
case 24: {
    double amount;
    cout << "Введіть кількість дизельного палива для заправки: ";
    cin >> amount;
    double time = station.refuelDiesel(amount);
    if (time != -1) {
        cout << "Час заправки: " << time << " сек" << endl;
    }
    else {
        cout << "Невдалося заправити дизельним паливом!" << endl;
    }
    break;
}
case 25: {
    double amount;
    cout << "Введіть кількість бензину для заправки: ";
    cin >> amount;
    double time = station.refuelGasoline(amount);
    if (time != -1) {
        cout << "Час заправки: " << time << " сек" << endl;
    }
    else {
        cout << "Невдалося заправити бензином!" << endl;
    }
    break;
}
case 26: {
    double amount;
    cout << "Введіть кількість LPG для заправки: ";
    cin >> amount;
    double time = station.refuelLpg(amount);
    if (time != -1) {
        cout << "Час заправки: " << time << " сек" << endl;
    }
}

```

```

    }
    else {
        cout << "Невдалося заправити LPG!" << endl;
    }
    break;
}
case 27: {
    double amount;
    cout << "Введіть кількість електроенергії для заправки: ";
    cin >> amount;
    double time = station.refuelElectric(amount);
    if (time != -1) {
        cout << "Час заправки: " << time << " сек" << endl;
    }
    else {
        cout << "Невдалося заправити електроенергією!" << endl;
    }
    break;
}
case 0: return 0;
default: cout << "Невірний вибір!" << endl; break;
}
}

return 0;
}

```

Результати виконання програм:

```
E:\V522\lab4c++\x64\Debug\  x  +  v  -  □  x
2 - Встановити об'єм дизельного палива
3 - Переглянути об'єм дизельного палива
4 - Встановити об'єм бензину
5 - Переглянути об'єм бензину
6 - Встановити об'єм LPG
7 - Переглянути об'єм LPG
8 - Встановити об'єм електроенергії
9 - Переглянути об'єм електроенергії
10 - Встановити запас дизельного палива
11 - Переглянути запас дизельного палива
12 - Встановити запас бензину
13 - Переглянути запас бензину
14 - Встановити запас LPG
15 - Переглянути запас LPG
16 - Встановити запас електроенергії
17 - Переглянути запас електроенергії
18 - Встановити пропускну здатність заправочного пістолета
19 - Переглянути пропускну здатність заправочного пістолета
20 - Збільшити запас дизельного палива
21 - Збільшити запас бензину
22 - Збільшити запас LPG
23 - Збільшити запас електроенергії
24 - Заправка дизельним паливом
25 - Заправка бензином
26 - Заправка LPG
27 - Заправка електроенергією
0 - Вихід
Введіть вибір: 2
Введіть об'єм дизельного палива: 1000
```

```
E:\VS22\lab4c++\x64\Debug\  X + v
2 - Встановити об'єм дизельного палива
3 - Переглянути об'єм дизельного палива
4 - Встановити об'єм бензину
5 - Переглянути об'єм бензину
6 - Встановити об'єм LPG
7 - Переглянути об'єм LPG
8 - Встановити об'єм електроенергії
9 - Переглянути об'єм електроенергії
10 - Встановити запас дизельного палива
11 - Переглянути запас дизельного палива
12 - Встановити запас бензину
13 - Переглянути запас бензину
14 - Встановити запас LPG
15 - Переглянути запас LPG
16 - Встановити запас електроенергії
17 - Переглянути запас електроенергії
18 - Встановити пропускну здатність заправочного пістолета
19 - Переглянути пропускну здатність заправочного пістолета
20 - Збільшити запас дизельного палива
21 - Збільшити запас бензину
22 - Збільшити запас LPG
23 - Збільшити запас електроенергії
24 - Заправка дизельним паливом
25 - Заправка бензином
26 - Заправка LPG
27 - Заправка електроенергією
0 - Вихід
Введіть вибір: 3
Об'єм дизельного палива: 1000 л
```

```
E:\VS22\lab4c++\x64\Debug\  X + v
1 - Переглянути загальну інформацію про заправку
2 - Встановити об'єм дизельного палива
3 - Переглянути об'єм дизельного палива
4 - Встановити об'єм бензину
5 - Переглянути об'єм бензину
6 - Встановити об'єм LPG
7 - Переглянути об'єм LPG
8 - Встановити об'єм електроенергії
9 - Переглянути об'єм електроенергії
10 - Встановити запас дизельного палива
11 - Переглянути запас дизельного палива
12 - Встановити запас бензину
13 - Переглянути запас бензину
14 - Встановити запас LPG
15 - Переглянути запас LPG
16 - Встановити запас електроенергії
17 - Переглянути запас електроенергії
18 - Встановити пропускну здатність заправочного пістолета
19 - Переглянути пропускну здатність заправочного пістолета
20 - Збільшити запас дизельного палива
21 - Збільшити запас бензину
22 - Збільшити запас LPG
23 - Збільшити запас електроенергії
24 - Заправка дизельним паливом
25 - Заправка бензином
26 - Заправка LPG
27 - Заправка електроенергією
0 - Вихід
Введіть вибір: 10
Введіть запас дизельного палива: 500
```

```
E:\VS22\lab4c++\x64\Debug\  X + v
2 - Встановити об'єм дизельного палива
3 - Переглянути об'єм дизельного палива
4 - Встановити об'єм бензину
5 - Переглянути об'єм бензину
6 - Встановити об'єм LPG
7 - Переглянути об'єм LPG
8 - Встановити об'єм електроенергії
9 - Переглянути об'єм електроенергії
10 - Встановити запас дизельного палива
11 - Переглянути запас дизельного палива
12 - Встановити запас бензину
13 - Переглянути запас бензину
14 - Встановити запас LPG
15 - Переглянути запас LPG
16 - Встановити запас електроенергії
17 - Переглянути запас електроенергії
18 - Встановити пропускну здатність заправочного пістолета
19 - Переглянути пропускну здатність заправочного пістолета
20 - Збільшити запас дизельного палива
21 - Збільшити запас бензину
22 - Збільшити запас LPG
23 - Збільшити запас електроенергії
24 - Заправка дизельним паливом
25 - Заправка бензином
26 - Заправка LPG
27 - Заправка електроенергією
0 - Вихід
Введіть вибір: 11
Запас дизельного палива: 500 л
```

```
E:\VS22\lab4c++\x64\Debug\ x + v
2 - Встановити об'єм дизельного палива
3 - Переглянути об'єм дизельного палива
4 - Встановити об'єм бензину
5 - Переглянути об'єм бензину
6 - Встановити об'єм LPG
7 - Переглянути об'єм LPG
8 - Встановити об'єм електроенергії
9 - Переглянути об'єм електроенергії
10 - Встановити запас дизельного палива
11 - Переглянути запас дизельного палива
12 - Встановити запас бензину
13 - Переглянути запас бензину
14 - Встановити запас LPG
15 - Переглянути запас LPG
16 - Встановити запас електроенергії
17 - Переглянути запас електроенергії
18 - Встановити пропускну здатність заправочного пістолета
19 - Переглянути пропускну здатність заправочного пістолета
20 - Збільшити запас дизельного палива
21 - Збільшити запас бензину
22 - Збільшити запас LPG
23 - Збільшити запас електроенергії
24 - Заправка дизельним паливом
25 - Заправка бензином
26 - Заправка LPG
27 - Заправка електроенергією
0 - Вихід
Введіть вибір: 19
Пропускна здатність заправочного пістолета: 4 л/сек
```

```
E:\VS22\lab4c++\x64\Debug\ x + v
3 - Переглянути об'єм дизельного палива
4 - Встановити об'єм бензину
5 - Переглянути об'єм бензину
6 - Встановити об'єм LPG
7 - Переглянути об'єм LPG
8 - Встановити об'єм електроенергії
9 - Переглянути об'єм електроенергії
10 - Встановити запас дизельного палива
11 - Переглянути запас дизельного палива
12 - Встановити запас бензину
13 - Переглянути запас бензину
14 - Встановити запас LPG
15 - Переглянути запас LPG
16 - Встановити запас електроенергії
17 - Переглянути запас електроенергії
18 - Встановити пропускну здатність заправочного пістолета
19 - Переглянути пропускну здатність заправочного пістолета
20 - Збільшити запас дизельного палива
21 - Збільшити запас бензину
22 - Збільшити запас LPG
23 - Збільшити запас електроенергії
24 - Заправка дизельним паливом
25 - Заправка бензином
26 - Заправка LPG
27 - Заправка електроенергією
0 - Вихід
Введіть вибір: 24
Введіть кількість дизельного палива для заправки: 40
Час заправки: 10 сек
```

Висновок: На даній лабораторній роботі я познайомився з класами та об'єктами в мові програмування C++. Створив програму яка зберігає об'єм кожного з 4-ох сховищ пального, об'єм запасів кожного з 4-ох видів пального, пропускну здатність заправочного пістолета, модифікує певні характеристики та обчислює час заправки.