Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»



Звіт

З лабораторної роботи № 5, варіант 9

З дисципліни « ПРОГРАМУВАННЯ, ЧАСТИНА 2 (ОБ’ЄКТНООРІЄНТОВАНЕ ПРОГРАМУВАННЯ) »

На тему: «Перевантаження операторів»

Виконав: ст. гр. КІ-15

Горбачук В.Р.

Перевірив: викладач

Козак Н.Б.

Львів – 2020

**Завдання:**

9: +, - , (), >

**Код програми:**

#include "pch.h"

#include <iostream>

#include <string>

using namespace std;

class CPlane {

public:

CPlane() {

setDefault();

}

string name;

int fuel;

int passenger\_c;

int passenger;

void setDefault() {

name = "Unknown";

fuel = 0;

passenger\_c = 100;

}

void setParameters() {

cout << "Введiть iм'я лiтака: ";

cin >> name;

cout << "Введiть кiлькiсть палива: ";

cin >> fuel;

}

void getParameters() {

cout << "Iм'я - ";

cout << name << endl;

cout << "Кiлькiть палива (L) = " << fuel << endl;

cout << "Кiлькiсть пасажиромiткостi = " << passenger\_c << endl;

}

void fly()

{

if (passenger > passenger\_c) {

cout << "Лiтак не може летiти, через велику кiлькiсть пасажирiв!" << endl;

}

else {

if (fuel < 50) {

cout << "Лiтак розбився!";

passenger = 0;

}

else {

fuel = fuel - 50;

passenger = 0;

cout << "Лiтак успiшно приземлився, всi пасажири вийшли";

}

}

}

void addNewPassenger() {

int new\_passenger;

cout << "Введiть скiльки пасажирiв ви хочете добавити: ";

cin >> new\_passenger;

passenger = passenger + new\_passenger;

}

// Перевантаження Операторів

CPlane operator+(const CPlane &secondPlane) { // Перевантаження оператора '+'

//Перевантаження оператора '+' визначає наступну логіку , операнди типу int двох обєктів додаються між собою

// поля типу string обєднуються

CPlane planeForReturn; // Повертаємий обєкт

planeForReturn.fuel = this->fuel + secondPlane.fuel;

planeForReturn.name = this->name + ", " + secondPlane.name;

planeForReturn.passenger = this->passenger + secondPlane.passenger;

planeForReturn.passenger\_c = this->passenger\_c + secondPlane.passenger\_c;

return planeForReturn;

}

CPlane operator-(const CPlane &secondPlane) { // Перевантаження оператора '-'

//Перевантаження оператора '-' визначає наступну логіку , операнди типу int двох обєктів віднімаються між собою

// поле типу string стає undefined

CPlane planeForReturn; // Повертаємий обєкт

planeForReturn.fuel = this->fuel + secondPlane.fuel;

planeForReturn.name = "undefined"; // Невизначене імя

planeForReturn.passenger = this->passenger + secondPlane.passenger;

planeForReturn.passenger\_c = this->passenger\_c + secondPlane.passenger\_c;

return planeForReturn;

}

bool operator>(CPlane secondPlane) { // Перевантаження оператора '>' , 'більше'

// Перевантаження оператора '>' , визначає наступну логіку -

// якщо поля типу int першого операнду більші за поля типу int 2 операнду то повертаємо true інакше flase

if (this->fuel > secondPlane.fuel && this->passenger > secondPlane.passenger && this->passenger\_c > secondPlane.passenger\_c) {

return true;

}

else return false;

}

CPlane& operator= (const CPlane &secondPlane) { // Перевантаження оператора присвоювання

if (&secondPlane == this) { // якщо операнди рівні повернути операнд без змін

return \*this;

}

else { // якщо операнди різні поміняти значення повертаємого операнду

this->fuel = secondPlane.fuel;

this->name = secondPlane.name;

this->passenger = this->passenger;

this->passenger\_c = this->passenger\_c;

return \*this;

}

}

operator int() // Перевантаження оператора '()' , оператора приведення до типу int

{

return fuel;

}

operator string() // Перевантаження оператора '()' , оператора приведення до типу string

{

return name;

}

};

ostream& operator <<(ostream& output, const CPlane& obj) // Перевантаження оператору '<<' Потокового виводу

{

cout << "Iм'я - ";

cout << obj.name << endl;

cout << "Кiлькiть палива (L) = " << obj.fuel << endl;

cout << "Кiлькiсть пасажиромiткостi = " << obj.passenger\_c << endl;

return output;

}

istream& operator >> (istream &s, CPlane &obj) // Перевантаження оператору '>>' Потокового вводу

{

cout << "Input Iм'я - ";

cin >> obj.name;

cout << "Input Кiлькiть палива (L) = ";

cin >> obj.fuel;

cout << "Input Кiлькiсть пасажиромiткостi = ";

cin >> obj.passenger\_c;

return s;

}

//main.cpp

int main()

{

setlocale(LC\_CTYPE, "ukr");

CPlane Myplane;

CPlane Myplane1;

//Демонстрація роботи перевантажень

cout << endl << "Демонстрацiя перевантаження оператора '>>' : " << endl;

cin >> Myplane;

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

cout << endl << "Демонстрацiя перевантаження оператора '<<' : " << endl;

cout << Myplane;

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

cout << endl << "Демонстрацiя перевантаження оператора '=' : " << endl;

Myplane1 = Myplane;

cout << "Myplane1 is the same as MyPlane\n" << Myplane1;

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

cout << endl << "Демонстрацiя перевантаження оператора '>' : " << endl;

cout << (Myplane > Myplane1);

cout << "" << endl;

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

cout << endl << "Демонстрацiя перевантаження оператора '+' i '-' : " << endl;

Myplane = Myplane + Myplane1;

cout << Myplane;

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

cout << endl << "Демонстрацiя перевантаження оператора '()' : " << endl;

int tempFuel = (int)Myplane;

cout << tempFuel << endl;

string tempName = (string)Myplane;

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

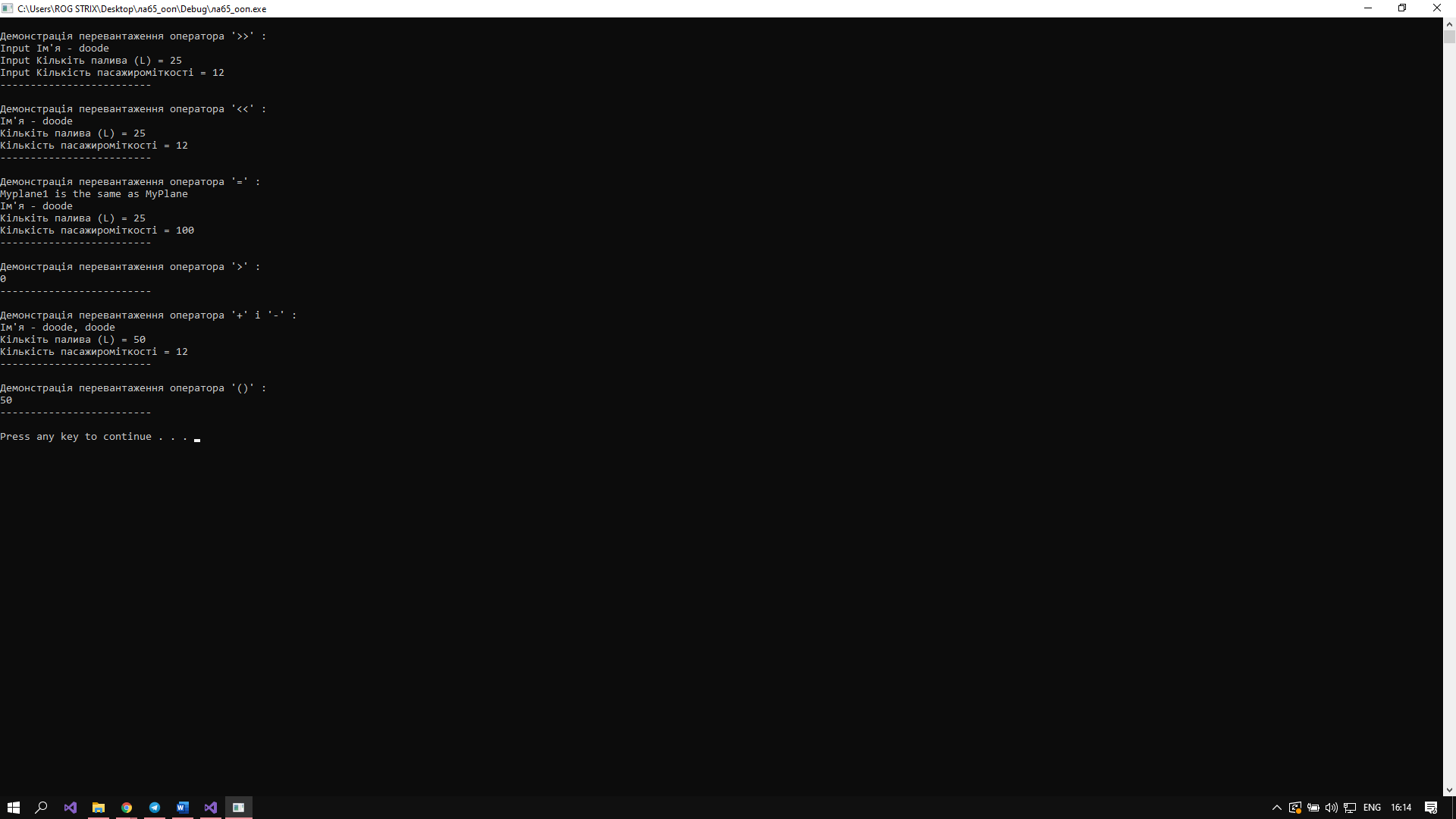
cout << " " <<endl;

system("pause");

return 0;

}

**Вікно результату:**



**Висновок.** На цій лабораторній роботі я познайомився з перенавантаженням операторів.