Міністерство освіти і науки України

Національний університет „Львівська політехніка”

Кафедра “Електронних обчислювальних машин”



**Звіт з лабораторної роботи №5**

на тему:

“ПЕРЕВАНТАЖЕННЯ ОПЕРАТОРІВ”

**Виконав:**

Ст. гр. КІ-15

Стецик О.І

**Перевірив:**

Викладач

Козак Н.Б.

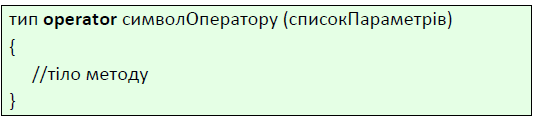
Львів – 2020

**Мета роботи:** познайомитися із перевантаженням операторів.

**Теоретичні відомості:**

**Перевантаження операторів**

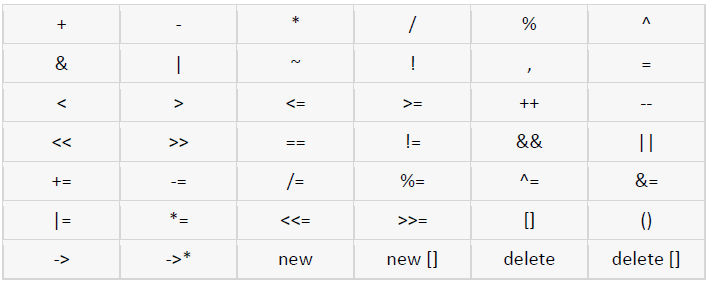
Кожному оператору мова С++ ставить у відповідність ім'я функції, що складається з ключового слова operator, власне оператору та аргументів відповідних типів:



Щоб використовувати операцію над об'єктами класів, ця операція повинна бути перевантажена, але є два виключення. Операції присвоювання (=) і взяття адреси (&) створюються в класі автоматично за замовчуванням, тому їх можна використовувати без явного перевантаження. За замовчуванням операція присвоювання зводиться до побітового копіювання даних-елементів класу. Проте таке побітове копіювання небезпечне для класів з елементами, що вказують на динамічно виділені області пам'яті, масиви, рядки, оскільки в цьому випадку відбувається копіювання не даних (глибоке копіювання), а лише вказівників на дані (поверхневе копіювання). Для таких класів слід явно перевантажувати операцію присвоювання і здійснювати у ній глибоке копіювання. Операція адресації також може бути використана з об'єктами будь-яких класів без перевантаження. Вона просто повертає адресу об'єкта в пам'яті. Але операцію адресації можна також і перевантажувати.

При перевантаженні операцій ( ), [], -> та = функція перевантаження операції може бути оголошена лише як метод класу. Для інших операцій функції перевантаження операцій можуть не бути методами класу.

Оператори, які можна перевантажити:



Оператори, які не можна перевантажити:

- sizeof

- . (селектор елемента структури або класу)

- \* (оператор доступу до елементу за вказівником)

- :: (оператор дозволу видимості)

- ?: (тернарний оператор)

- typeid

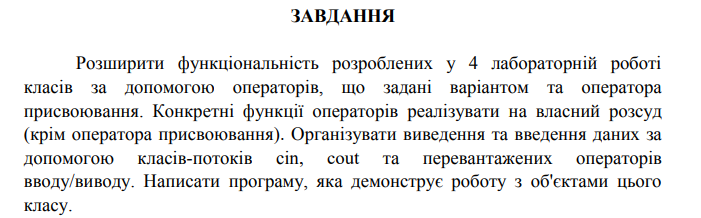
- const\_cast

- dynamic\_cast

- reinterpret\_cast

- static\_cast

- # і ## (символи препроцесору)

******

*Код програми:*

#include "stdafx.h"

#include <iostream>

#include <string>

#include <iomanip>

#include <fstream>

using namespace std;

class CTranslator

{

string name = "newTranslator";

int speed = 1;// мс/символ

bool check;

int t[];

public:

int translate()

{

string language1, language2;

cout << "Available languages:" << endl;

getLanguage();

cout << "Enter first language" << endl;

cin >> language1;

cout << "Enter second language" << endl;

cin >> language2;

if (check = language1 == language2) { cout << "The languages are same" << endl; system("pause"); }

system("cls");

string word1, word2;

cout << "Enter a word on " << language1 << endl;

cin >> word1;

system("cls");

cout << language1 << setw(30) << language2 << endl;

cout << endl;

cout << word1 << setw(28) << "\_Word in " << language2 << "\_" << endl;

return word1.length();

}

void setLanguage()

{

string lang;

fstream outfile("File.txt", ios::app);

cout << "Enter a language." << endl;

cin >> lang;

outfile.setf(ios::left);

outfile << lang << endl;

outfile.close();

system("cls");

}

void getLanguage()

{

string name;

ifstream fin("File.txt");

while (!fin.eof()) {

getline(fin, name);

cout << name << endl;

}

};

void getLength()

{

}

CTranslator& operator += (CTranslator c2)

{

speed += c2.speed;

return \*this;

}

CTranslator& operator++ ()

{

speed += 1;

return \*this;

}

int& operator[] (int i)

{

return t[i];

}

};

bool operator ==(CTranslator l1, CTranslator l2) { return l1 == l2; }

int main()

{

CTranslator translator;

while (true)

{

cout << "Work with languages - press 1" << endl;

cout << "To translate - press 2" << endl;

cout << "Exit - press 3" << endl;

int choise;

cin >> choise;

system("cls");

if (choise == 1) {

int choise2;

cout << "Already added languages:" << endl;

translator.getLanguage();

cout << "If you want to add new language - press 1;\nIf no - press 2" << endl;

cin >> choise2;

if (choise2 == 1) { translator.setLanguage(); system("cls"); }

else(system("cls"));

}

if (choise == 2)

{

int len;

len = translator.translate();

int t[] = { 0 };

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

{

t[0] += 1;

}

cout << "Time: " << t[0] << " ms" << endl;

system("pause");

system("cls");

}

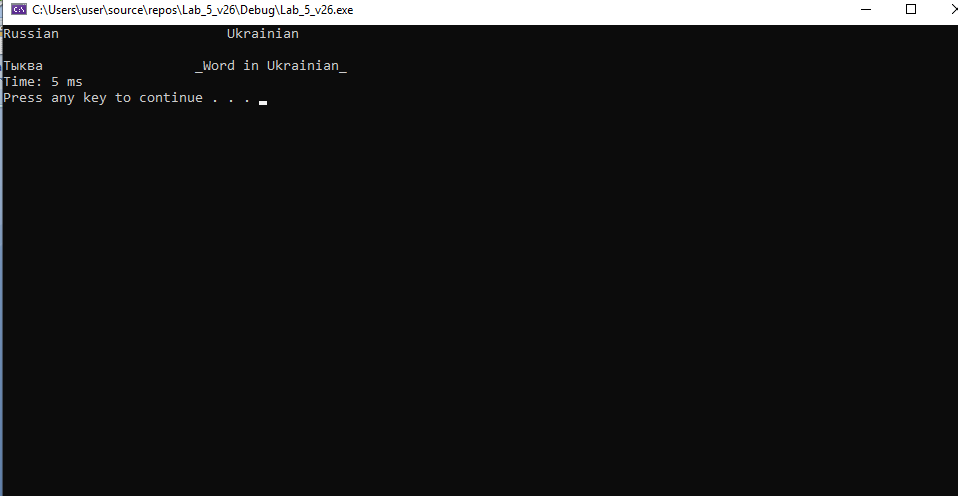
if (choise == 3) { break; }

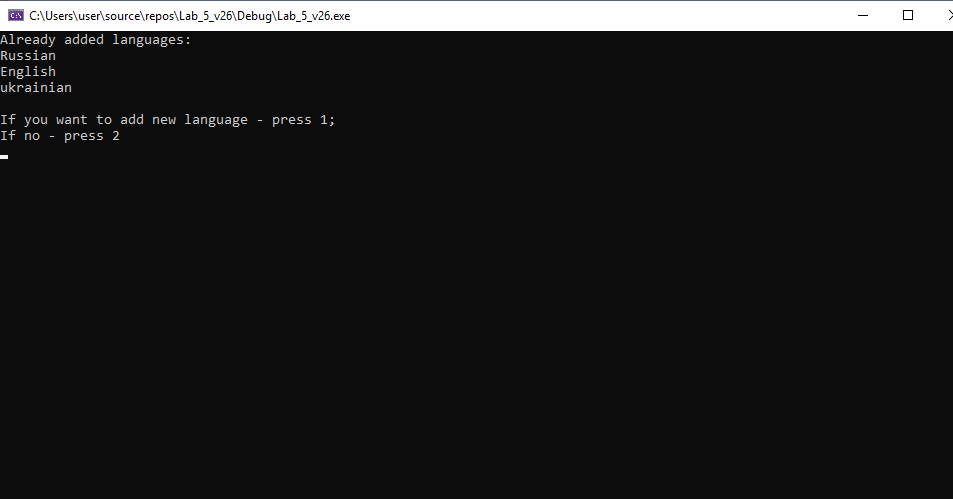
}

system("pause");

}

*Вікно результату:*

**

**

*Висновок:* я познайомився з таким «явищем», як перевантаження операторів. Також пропрактикувався в їх написанні.