Міністерство освіти і науки України

Національний університет “Львівська Політехніка”

Кафедра ЕОМ



**ЗВІТ**

З лабораторної роботи № 5

З дисципліни: “Програмування, частина 2 (Об'єктно-орієнтоване програмування)”

На тему: “ПЕРЕВАНТАЖЕННЯ ОПЕРАТОРІВ”

Варіант 18

Виконав: ст. гр. КІ-15

Марков Л.Ю.

Прийняв:

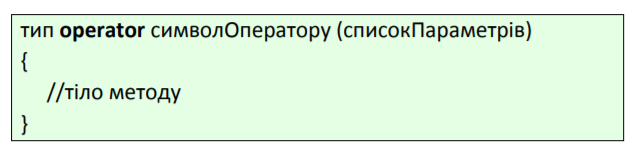
Козак Н.Б.

Львів – 2020

Мета: познайомитися із перевантаженням операторів.

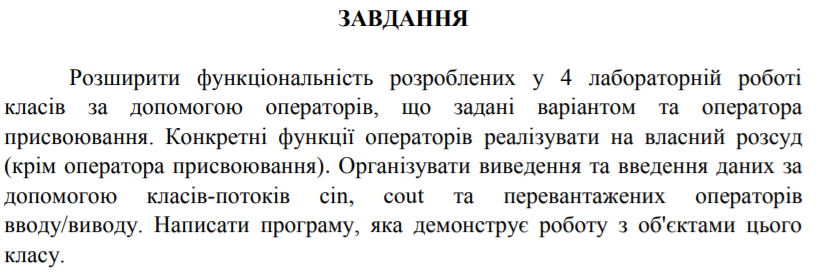
ТЕОРЕТИЧНІ ВІДОМОСТІ

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.

Варіант 18:





Виконання:

#include <iostream>

#include <string>

using namespace std;

class CHuman

{

private:

string currNm;

int age, countOfSkills, priceForSkill;

public:

CHuman()

{

currNm = "Unknown";

age = 0;

countOfSkills = 0;

priceForSkill = 0;

}

CHuman(string currNm, int age, int countOfSkills, int priceForSkill)

{

this->currNm = currNm;

this->age = age;

this->countOfSkills = countOfSkills;

this->priceForSkill = priceForSkill;

}

void Print()

{

cout << "Current name: " << currNm << endl <<

"Age: " << age << endl <<

"Count of skills: " << countOfSkills << endl <<

"Price for 1 skill (grn): " << priceForSkill << endl;

}

void Print2()

{

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

cout << "Enter current name of 2 human: ";

cin >> this->currNm;

cout << "Enter current age of 2 human: ";

cin >> this->age;

cout << "Enter count of skills: ";

cin >> this->countOfSkills;

cout << "Enter price for 1 skill (grn): ";

cin >> this->priceForSkill;

}

CHuman operator + (const CHuman& other)

{

CHuman x;

x.age = this->age + other.age;

x.countOfSkills = this->countOfSkills + other.countOfSkills;

x.priceForSkill = this->priceForSkill + other.priceForSkill;

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

cout << "Operator +" << endl;

cout << "Age: " << x.age << endl;

cout << "Count of skills: " << x.countOfSkills << endl;

cout << "Price for skill: " << x.priceForSkill << endl;

}

CHuman operator\*(const CHuman& other)

{

CHuman y;

y.age = this->age \* other.age;

y.countOfSkills = this->countOfSkills \* other.countOfSkills;

y.priceForSkill = this->priceForSkill \* other.priceForSkill;

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

cout << "Operator \*" << endl;

cout << "Age: " << y.age << endl;

cout << "Count of skills: " << y.countOfSkills << endl;

cout << "Price for skill: " << y.priceForSkill << endl;

}

bool operator < (const CHuman& other)

{

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

cout << "Operator < " << endl;

if (this->age < other.age && this->countOfSkills < other.countOfSkills && this->priceForSkill < other.priceForSkill)

{

cout << "True" << endl;

}

else

{

cout << "False" << endl;

}

}

bool operator> (const CHuman& other)

{

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

cout << "Operator > " << endl;

if (this->age > other.age && this->countOfSkills > other.countOfSkills && this->priceForSkill > other.priceForSkill)

{

cout << "True" << endl;

}

else

{

cout << "False" << endl;

}

}

};

int main()

{

setlocale(LC\_ALL, "UKR");

CHuman firstHuman("Alison", 18, 5, 100);

firstHuman.Print();

CHuman secondHuman;

secondHuman.Print2();

CHuman a = firstHuman + secondHuman;

CHuman b = firstHuman \* secondHuman;

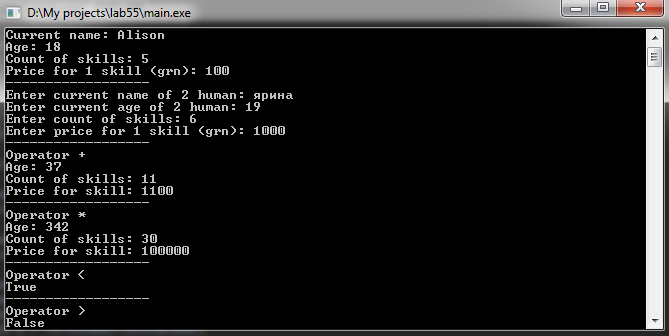
bool c = firstHuman < secondHuman;

bool d = firstHuman > secondHuman;

return 0;

}

**Результат:**

****

Висновок: процесі виконання даної лабораторної роботи я ознайомився із перевантаженням операторів.