Міністерство освіти і науки України

Національний технічний університет України «Київський політехнічний

інститут імені Ігоря Сікорського"

Факультет інформатики та обчислювальної техніки

Катедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 3 з дисципліни

«Основи програмування 2.

Модульне програмування»

«Перевантаження операторів»

Варіант 18

Виконав студент ІП-11 Лесів Владислав Ігорович

Перевірив Вітковська Ірина Іванівна

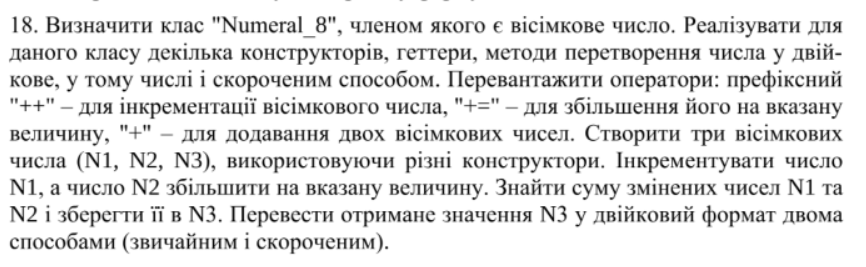
Київ 2022

**Лабораторна робота 3**

**Перевантаження операторів**

**Мета** – вивчити механізми створення класів з використанням перевантажених операторів (операцій).

**Варіант №18.**

****

**Постановка задачі.**

Результатом розв’язку є створені три об’єкти класу вісімкових чисел, з якими проведені відповідні операції, а значення одного з чисел переведене двома способами у двійковий формат. Для визначення результату повинно бути задане вісімкове число для одного з об’єктів, що буде створене конструктором з параметром, а також величина – також вісімкове число – на яку збільшуватимемо число N2. Інших початкових даних для розв’язку не потрібно.

Математичне формулювання задачі зводиться до знаходження суми вісімкових чисел: якщо маємо інкремент, зауважимо, що 7 – остання цифра перед новим десятком, а при сумі чисел переводимо їх у десяткові, додаємо, а потім знову у вісімкові; а також переведення вісімкового числа у бінарний формат: 1 спосіб – кожну цифру числа замінюємо відповідною бінарною тріадою, перші нулі забираємо, 2 спосіб – переводимо у десяткове, а потім діленням у двійкове.

**Виконання мовою C++.**

**Код програми:**

**classes.h**

#pragma once

#include <iostream>

#include <string>

using namespace std;

class Numeral\_8 {

int number;

int toBin();

public:

Numeral\_8();

Numeral\_8(int);

Numeral\_8(const Numeral\_8&);

int getNum();

int getBin();

Numeral\_8 operator++();

Numeral\_8 operator+=(const Numeral\_8);

Numeral\_8 operator+(const Numeral\_8);

};

int toNum(int, int, int);

**Lab3\_cpp.cpp**

#include "classes.h"

int main()

{

setlocale(LC\_ALL, "ukr");

Numeral\_8 n1;

int v;

cout << "Введiть вiсiмкове число №2: "; cin >> v;

Numeral\_8 n2(v);

Numeral\_8 n3(n2);

cout << "Створенi об'єкти класу вiсiмкових чисел: ";

cout << "\nN1 (за замовчуванням) = "<<n1.getNum();

cout << "\nN2 (введене) = "<<n2.getNum();

cout << "\nN3 (копiя N2) = "<<n3.getNum();

++n1;

int x;

cout << "\nВведiть число, на скiльки збiльшити N2: "; cin >> x;

Numeral\_8 n4(x);

n2 += n4;

cout << "\nЗмiненi числа:\nN1 (iнкрементовано) = "<<n1.getNum();

cout << "\nN2 (додане введене число) = "<<n2.getNum();

n3 = n1 + n2;

cout << "\nЗмiнене N3 (сума змiнених N1 i N2) = "<<n3.getNum();

n3.getBin();

}

**classes.cpp**

#include "classes.h"

Numeral\_8::Numeral\_8() { number = 7; } //Конструктор за замовчуванням

Numeral\_8::Numeral\_8(int n) { //Конструктор з параметром

if (to\_string(n).find('8')!=string::npos || to\_string(n).find('9')!=string::npos) {

cout << "\nНе вiсiмкове число. Задання за замовчуванням 16.\n";

number = 16;

}

else

number = n;

}

Numeral\_8::Numeral\_8(const Numeral\_8& obj) { number = obj.number; } //Конструктор копіювання

int Numeral\_8::getNum() { return number; }

int Numeral\_8::getBin(){

cout << "\n1 спосiб:"; int n=(\*this).toBin();

cout << "\n2 спосiб:"; toNum(8, 2, (\*this).number);

return n;

}

int Numeral\_8::toBin() { //Спосіб переведення тріадами

string s = "", temp=to\_string(number);

for (int i = 0; i < temp.length(); i++) {

switch (temp[i]) {

case '0':

s += "000";

break;

case '1':

s += "001";

break;

case '2':

s += "010";

break;

case '3':

s += "011";

break;

case '4':

s += "100";

break;

case '5':

s += "101";

break;

case '6':

s += "110";

break;

case '7':

s += "111";

break;

}

}

cout << "\n\tу двiйковiй системi: " << stoll(s);

return stoll(s);

}

Numeral\_8 Numeral\_8::operator++() { //Перевизначаємо префіксний інкремент

int c = 0;

while (number % 10 == 7) { //Забираємо останні сімки

c++;

number /= 10;

}

number++;

number \*= pow(10, c);

return \*this;

}

Numeral\_8 Numeral\_8::operator+(Numeral\_8 obj) {

int s = toNum(8, 10, (\*this).number) + toNum(8, 10, obj.number); //Сума в десятковій

s = toNum(10, 8, s); //Перевід у вісімкову

return Numeral\_8(s);

}

Numeral\_8 Numeral\_8::operator+=(const Numeral\_8 obj) {

number = (\*this + obj).number; //Перевизначеним додаванням

return \*this;

}

int toNum(int f, int t, int number) { //Переведення між системами

int dec = 0; string temp = to\_string(number);

for (int i = 0; i < temp.length(); i++)

dec += (int)(temp[i] - '0') \* pow(f, temp.length() - i - 1); //Перевід у десяткову

if (t == 2)

cout << "\n\tу десятковiй системi: " << dec;

string bintemp = "";

while (dec != 0) { //Перевід діленням у вихідну

bintemp += to\_string(dec % t);

dec /= t;

}

reverse(bintemp.begin(), bintemp.end()); //Обертаємо остачі

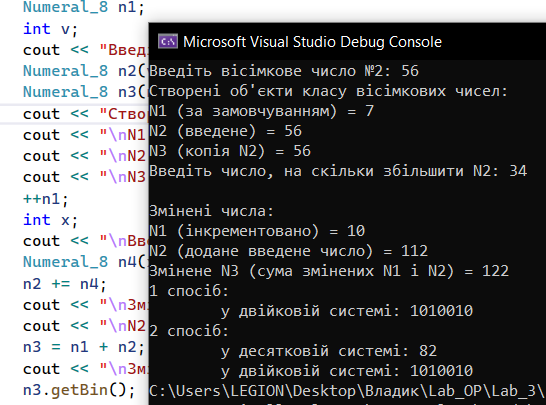
if (t == 2)

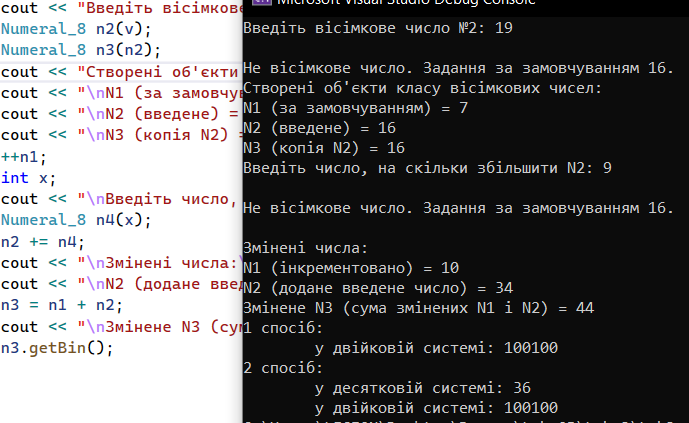
cout << "\n\tу двiйковiй системi: " << stoll(bintemp);

return stoll(bintemp);

}

**Випробування алгоритму.**

****

****

**Висновок.** Отже, у цій роботі я вивчив механізми створення класів з використанням перевантажених операторів (операцій). У результаті лабораторної роботи було розроблено програму, яка виконує задачу відповідно до постановки. Використовуючи розроблений клас вісімкового числа з атрибутом його значення, різними видами конструкторів, методу переведення числа в бінарне та перевантаження операторів ++(префіксний), + та +=, а також функцію для переведення числа між системами числення, отримуємо коректний результат.