

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

Перевірів Вітковська Ірина Іванівна

Київ 2022

Лабораторна робота 3

Перевантаження операторів

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

Варіант №18.

18. Визначити клас "Numeral_8", членом якого є вісімкове число. Реалізувати для даного класу декілька конструкторів, геттери, методи перетворення числа у двійкове, у тому числі і скороченим способом. Перевантажити оператори: префіксний "++" – для інкрементації вісімкового числа, "+=" – для збільшення його на вказану величину, "+" – для додавання двох вісімкових чисел. Створити три вісімкових числа (N1, N2, N3), використовуючи різні конструктори. Інкрементувати число N1, а число N2 збільшити на вказану величину. Знайти суму змінених чисел N1 та N2 і зберегти її в N3. Перевести отримане значення N3 у двійковий формат двома способами (звичайним і скороченим).

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 – кожен 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 << "Введіть вісімкове число №2: "; cin >> v;
    Numeral_8 n2(v);
    Numeral_8 n3(n2);
    cout << "Створені об'єкти класу вісімкових чисел: ";
    cout << "\nN1 (за замовчуванням) = "<<n1.getNum();
    cout << "\nN2 (введене) = "<<n2.getNum();
    cout << "\nN3 (копія N2) = "<<n3.getNum();
    ++n1;
    int x;
    cout << "\nВведіть число, на скільки збільшити N2: "; cin >> x;
    Numeral_8 n4(x);
    n2 += n4;
    cout << "\nЗмінені числа:\nN1 (інкрементовано) = "<<n1.getNum();
    cout << "\nN2 (додане введене число) = "<<n2.getNum();
    n3 = n1 + n2;
    cout << "\nЗмінене N3 (сума змінених N1 і 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 вісімкове 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 спочи6:"; int n=(*this).toBin();
    cout << "\n2 спочи6:"; 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\ty двійковій системі: " << 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;    //Перевизначенням 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\ty десятковій системі: " << 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\ty двійковій системі: " << stoll(bintemp);
    return stoll(bintemp);
}

```

Випробування алгоритму.

```

Numeral_8 n1;
int v;
cout << "Введіть вісімкове число №2: ";
Numeral_8 n2(v);
Numeral_8 n3(n2);
cout << "Створені об'єкти класу вісімкових чисел: N1 (за замовчуванням) = 7\n";
cout << "N1 (введене) = 56\n";
cout << "N2 (копія N2) = 56\n";
cout << "N3 (введене) = 34\n";
v++;
n1++;
int x;
cout << "Змінені числа: N1 (інкрементовано) = 10\n";
cout << "N2 (додане введене число) = 112\n";
Numeral_8 n4(n1 + n2);
cout << "Змінене N3 (сума змінених N1 і N2) = 122\n";
n2 += n4;
cout << "1 спосіб: N3 у двійковій системі: 1010010\n";
cout << "2 спосіб: N2 у десятковій системі: 82\n";
n3 = n1 + n2;
cout << "N3 у двійковій системі: 1010010\n";
n3.getBin();

```

```

cout << "Введіть вісімкове  
Numeral_8 n2(v);  
Numeral_8 n3(n2);  
cout << "Створені об'єкти  
cout << "\nN1 (за замовчу  
cout << "\nN2 (введене) =  
cout << "\nN3 (копія N2) =  
++n1;  
int x;  
cout << "\nВведіть число,  
Numeral_8 n4(x);  
n2 += n4;  
cout << "\nЗмінені числа:  
cout << "\nN2 (додане введ  
n3 = n1 + n2;  
cout << "\nЗмінене N3 (сум  
n3.getBin();

```

```

Microsoft Visual Studio Debug Console
Введіть вісімкове число №2: 19
Не вісімкове число. Задання за замовчуванням 16.
Створені об'єкти класу вісімкових чисел:
N1 (за замовчуванням) = 7
N2 (введене) = 16
N3 (копія N2) = 16
Введіть число, на скільки збільшити N2: 9
Не вісімкове число. Задання за замовчуванням 16.
Змінені числа:
N1 (інкрементовано) = 10
N2 (додане введене число) = 34
Змінене N3 (сума змінених N1 і N2) = 44
1 спосіб:
у двійковій системі: 100100
2 спосіб:
у десятичній системі: 36
у двійковій системі: 100100

```

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.