

Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний інститут
імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи №4 з дисципліни
«Основи програмування 4. Модульне програмування»

«Перевантаження операторів»

Варіант 18

Виконав студент

Лазьов Кирило Владиславович
(шифр, прізвище, ім'я, по батькові)

Перевірила

Вечерковська Анастасія Сергіївна
(прізвище, ім'я, по батькові)

Лабораторна робота №4
Перевантаження операторів
Варіант 18
Задача

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).

C++

main.cpp:

```
1  #include "Classes.h"
2
3  using namespace std;
4
5  int main() {
6      int a;
7      cout << "Input N1: ";
8      cin >> a;
9      string b;
10     cout << "Input N2: ";
11     cin >> b;
12
13     Numeral_8 N1(a);
14     Numeral_8 N2(b);
15     Numeral_8 N3(N2);
16
17     cout << "N1.a = " << N1.getA() << endl;
18     cout << "N2.a = " << N2.getA() << endl;
19     cout << "N3.a = " << N3.getA() << endl;
20
21     ++N1;
22     cout << "++N1.a = " << N1.getA() << endl;
23
24     int num;
25     cout << "Input a number to add to N2: ";
26     cin >> num;
27     N2 += num;
28     cout << "N2.a += " << num << " = " << N2.getA() << endl;
29
30     N3 = N1 + N2;
31     cout << "N3.a = N1 + N2 = " << N3.getA() << endl;
32
33     cout << "Binary first method: " << N3.octal2binary() << endl;
34     cout << "Binary second method: " << N3.octal2binary2() << endl;
35
36 }
```

Classes.h

```
1  #pragma once
2  #include<iostream>
3  #include<string>
4  using namespace std;
5
6  class Numeral_8 {
7      int a;
8  public:
9      Numeral_8(int);
10     Numeral_8(string);
11     Numeral_8(const Numeral_8& obj);
12     Numeral_8& operator++();
13     Numeral_8& operator+= (int);
14     friend Numeral_8 operator+(Numeral_8, Numeral_8);
15
16     int getA();
17     int octal2binary();
18     string octal2binary2();
19 };
20
```

Classes.cpp

```
1  #include "Classes.h"
2  using namespace std;
3
4  Numeral_8::Numeral_8(int b) {
5      int i = 1;
6      int octal = 0;
7      while (b != 0) {
8          octal += (b % 8) * i;
9          b /= 8;
10         i *= 10;
11     }
12     a = octal;
13 };
14
15 Numeral_8::Numeral_8(string line) {
16     int b = stoi(line);
17     int octal = 0;
18     int i = 1;
19     while (b != 0) {
20         octal += (b % 8) * i;
21         b /= 8;
22         i *= 10;
23     }
24     a = octal;
25 }
26
27 Numeral_8::Numeral_8(const Numeral_8& obj) {
28     a = obj.a;
29 }
30
```

```
31 Numeral_8& Numeral_8::operator++() {
32     a++;
33     if (a % 10 > 7) {
34         a += 2;
35     }
36     if (a / 10 > 7 and a / 10 < 10) {
37         a += 20;
38     }
39     return *this;
40 }
41
42 Numeral_8& Numeral_8::operator+=(int b) {
43     a += b;
44     if (a % 10 > 7) {
45         a += 2;
46     }
47     if (a / 10 > 7 and a / 10 < 10) {
48         a += 20;
49     }
50     return *this;
51 }
52
```

```

53 Numeral_8 operator+(Numeral_8 x, Numeral_8 y) {
54     Numeral_8 tmp(0);
55     int a = x.getA();
56     int b = y.getA();
57     int carry = 0, digit = 0, mult = 1;
58     while (a != 0 or b != 0 or carry != 0) {
59         digit = a % 10 + b % 10 + carry;
60
61         if (digit > 7)
62         {
63             carry = 1;
64             digit %= 8;
65         }
66         else
67             carry = 0;
68
69         tmp.a += digit * mult;
70         mult *= 10;
71
72         a /= 10;
73         b /= 10;
74     }
75     return tmp;
76 }
77
78 int Numeral_8::getA() {
79     return a;
80 }
81

```

```

82 int Numeral_8::octal2binary() {
83     int octal = a;
84     int decimalNum = 0, binaryNum = 0, count = 0;
85
86     while (octal != 0) {
87         decimalNum += (octal % 10) * pow(_Left: 8, _Right: count);
88         ++count;
89         octal /= 10;
90     }
91
92     count = 1;
93     while (decimalNum != 0) {
94         binaryNum += (decimalNum % 2) * count;
95         decimalNum /= 2;
96         count *= 10;
97     }
98     return binaryNum;
99 }

```

```

101 string Numeral_8::octal2binary2() {
102     int octal = a;
103     int decimalNum = 0, binaryNum = 0, count = 0;
104     while (octal != 0) {
105         decimalNum += (octal % 10) * pow(_Left:8, _Right:count);
106         ++count;
107         octal /= 10;
108     }
109
110     int num = 0, digit = decimalNum;
111     while (digit != 0) {
112         digit /= 2;
113         num++;
114     }
115
116     int* array = new int[num];
117     for (int i = 0; i < num; i++) {
118         array[i] = pow(_Left:2, _Right:i);
119     }
120
121     string binary;
122     for (int j = num - 1; j >= 0; j--) {
123         if (decimalNum >= array[j]) {
124             decimalNum -= array[j];
125             binary += "1";
126         }
127         else {
128             binary += "0";
129         }
130     }
131     return binary;
132 }

```

Результати виконання програми

C++:

Microsoft Visual Studio Debug Console

```
Function sin(ax + b)
Input amount of functions: 3
Input a :2
Input b :4
Input a :7
Input b :3
Input a :8
Input b :9
Input x: 3
The function number 2 has the biggest value in point x
C:\Users\User\Source\Repos\OP_Lab3.2\x64\Debug\OP_Lab3.2.exe (proces
To automatically close the console when debugging stops, enable Tool
le when debugging stops.
Press any key to close this window . . .
```

