

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

Кафедра ІІІ

Звіт

з лабораторної роботи № 4 з дисципліни
«Основи програмування 2. Модульне програмування»

„ ПЕРЕВАНТАЖЕННЯ ОПЕРАТОРІВ ”

Варіант 8

Виконав(ла)

Дацьо Іван Іванович

(шифр, прізвище, ім'я, по батькові)

Перевірів

Вєчерковська Анастасія Сергіївна

(прізвище, ім'я, по батькові)

Київ 2022

Лабораторна робота 4

ПЕРЕВАНТАЖЕННЯ ОПЕРАТОРІВ

Мета – вивчити механізм створення класів з використанням перевантажених операторів/операцій.

Варіант 8

Задача

8. Визначити клас "Numeral_16", членом якого є шістнадцяткове число. Реалізувати для даного класу декілька конструкторів, геттери, метод перетворення числа у десяткове. Перевантажити оператори: постфіксний "++" – для інкрементації шістнадцяткового числа, "+=" – для збільшення його на вказану величину, "+" – для додавання двох шістнадцяткових чисел. Створити три шістнадцяткових 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 десятковий формат.

Код:

C++

```
foo.cpp x  foo.h x  main.cpp x  hex_numb.cpp x  hex_numb.h x  CMakeLists.txt x
1  #include "foo.h"
2  int main (){
3      Numeral_16 N1 , N2 , N3;
4      inputHexNUmber( &N1 , &N2 , &N3);
5      AllPrint( &N1 , &N2 , &N3);
6      int number_to;
7      number_to = inputNum();
8      N1++;
9      N2 += number_to;
10     N3 = N1 + N2;
11     cout <<"-----After changing-----"<<endl;
12     AllPrint( &N1 , &N2 , &N3);
13     cout <<"\nHexadecimal to Decimal (N3) : "<<(N3.convertHex())<<endl;
14 }
15
```

```
foo.cpp x foo.h x main.cpp x hex_numb.cpp x hex_numb.h x CMakeLists.txt x
1  #pragma once
2  #include <iostream>
3  #include <string>
4  #include <hex_numb.h>
5  using namespace std;
6  void inputHexNUmber(Numeral_16 & obj1 , Numeral_16 & obj2 , Numeral_16 &obj3);
7  string verHex(string number);
8  bool valHexs(string number);
9  void AllPrint(Numeral_16 & obj1 , Numeral_16 & obj2 , Numeral_16 &obj3);
10 int inputNum();
11 bool isNumber(string s);
```

```
foo.cpp x foo.h x main.cpp x hex_numb.cpp x hex_numb.h x CMakeLists.txt x
1  #include "foo.h"
2  void inputHexNUmber(Numeral_16 & obj1 , Numeral_16 & obj2 , Numeral_16 &obj3){
3      string hex_n;
4      cout << "Enter hex number for 1 item :"<<endl;
5      cin >> hex_n;
6      hex_n = verHex( number: hex_n);
7      obj1 = Numeral_16( hex: hex_n);
8      obj2 = Numeral_16(obj1);
9      obj3 = Numeral_16();
10
11
12 }
13 string verHex(string number){
14
15     while (!valHexs(number)){
16         cout <<"Enter correct hex number :"<<endl;
17         cin >> number;
18     }
19     cin.ignore();
20     return number;
21 }
22 bool valHexs(string number){
23     bool flag = false;
24     for (int i=0; i<number.length(); i++)
25     {
26         if (!isxdigit( C: number[i]))
27         {
28             return flag;
29         }
30     }
```

```

28         return flag;
29     }
30 }
31 return true;
32 }
33 void AllPrint(Numeral_16 & obj1 , Numeral_16 & obj2 , Numeral_16 &obj3){
34     cout <<"First (1) :";obj1.print();
35     cout <<"Second (2) :";obj2.print();
36     cout <<"Third (3) :";obj3.print();
37 }
38 }
39 int inputNum(){
40     string n;
41     cout <<"How much do you want to increase N2? :"<<endl;
42     cin >> n;
43     while (!isNumber(s: n) || stoi(str: n) <=0){
44         cout <<"Enter correct number :"<<endl;
45         cin >> n;
46     }
47     return stoi(str: n);
48 }
49 bool isNumber(string s)
50 {
51     for (char ch : s) {
52         if (!isdigit(ch)) return false;
53     }
54
55     return true;
56 }
57 //

```

```

1  #include <iostream>
2  #include <string>
3  #include <cmath>
4  using namespace std;
5  class Numeral_16 {
6      string hex_;
7      int toAdd = 0;
8  public:
9      Numeral_16();
10     Numeral_16(string hex_);
11     Numeral_16(const Numeral_16 & obj);
12     void setHex(string hex_);
13     string getHex();
14     string convertHex();
15     void print(){cout << "Your number is :"<<hex_<<endl;}
16     string hexAdd(string nex1 , string next2);
17     Numeral_16 operator++(int);
18     Numeral_16 operator +(const Numeral_16 obj);
19     Numeral_16 operator +=(const int number);
20
21
22 };
23

```

```

foo.cpp x foo.h x main.cpp x hex_numb.cpp x hex_numb.h x CMakeLists.txt x
Q 0 results
1 #include "hex_numb.h"
2 Numeral_16::Numeral_16() {}
3 Numeral_16::Numeral_16(string hex_) {
4     this->hex_ = hex_;
5 }
6 Numeral_16::Numeral_16(const Numeral_16 &obj) {
7     hex_ = obj.hex_;
8 }
9
10 void Numeral_16::setHex(string hex_) {
11     this->hex_ = hex_;
12 }
13
14 string Numeral_16::getHex() {
15     return this->hex_;
16 }
17
18 string Numeral_16::convertHex() { // convert from 16 to 10 (last task)
19     int base = 1;
20     int dec_val = 0;
21     for (int i = hex_.length() - 1; i >= 0; i--) {
22         if (hex_[i] >= '0' && hex_[i] <= '9') {
23             dec_val += (int(hex_[i]) - 48) * base; // convert from char '0' - '9' to number 0 - 9
24             base = base * 16;
25         }
26         else if (hex_[i] >= 'A' && hex_[i] <= 'F') { // convert from char 'A' - 'F' to number 10 - 15
27             dec_val += (int(hex_[i]) - 55) * base;
28             base = base * 16;
29         }
30     }
31 }

```

```

foo.cpp x foo.h x main.cpp x hex_numb.cpp x hex_numb.h x CMakeLists.txt x
Q 0 results
28     base = base * 16;
29 }
30 }
31 return to_string(val: dec_val);
32
33 }
34 string Numeral_16::hexAdd(string hex1, string hex2) // add to numbers
35 {
36     if (hex1.length() < hex2.length()) {
37         hex1.swap(&hex2); // check if hex 1 < hex2 to make loop
38     }
39
40     int length1, length2;
41     length1 = hex1.length();
42     length2 = hex2.length();
43     int flag = 0;
44     int get1, get2;
45     int sum;
46
47     while (length1 > 0)
48     {
49         // first
50         if (hex1[length1 - 1] >= 'A') {
51             get1 = hex1[length1 - 1] - 55;
52         }
53         else {
54             get1 = hex1[length1 - 1] - '0';
55         }
56         // second

```

```

foo.cpp x  foo.h x  main.cpp x  hex_numb.cpp x  hex_numb.h x  CMakeLists.txt x
Q  Cc W .* 0 results  ↑ ↓  + -  ☰ ☷ ☹
55     }
56     //second
57
58     if (length2 > 0)
59     {
60         if (hex2[length2 - 1] >= 'A') {
61             get2 = hex2[length2 - 1] - 55;
62         }
63         else {
64             get2 = hex2[length2 - 1] - '0';
65         }
66     }
67     else {
68         get2 = 0;
69     }
70
71     //sum of all ( 1 + 2 + ..)
72     sum = get1 + get2 + flag;
73
74     if (sum >= 16)
75     {
76         int left = sum % 16;
77         if (left >= 10)
78             hex1[length1 - 1] = 'A' + left % 10;
79         else
80             hex1[length1 - 1] = '0' + left;
81         flag = 1;
82     }
83     else

```

```

foo.cpp x  foo.h x  main.cpp x  hex_numb.cpp x  hex_numb.h x  CMakeLists.txt x
Q  Cc W .* 0 results  ↑ ↓  + -  ☰ ☷ ☹
79     else
80         hex1[length1 - 1] = '0' + left;
81     flag = 1;
82 }
83 else
84 {
85     if (sum >= 10)
86         hex1[length1 - 1] = 'A' + sum % 10;
87     else
88         hex1[length1 - 1] = '0' + sum;
89     flag = 0;
90 }
91
92 length1--;
93 length2--;
94 }
95
96 if (flag == 1) {
97     return "1" + hex1;
98 }
99 else {
100     return hex1;
101 }
102 }
103
104 Numeral_16 Numeral_16::operator++(int) {
105     toAdd++;
106     hex_ = hexAdd( hex1: hex_ , hex2: to_string( val: toAdd));
107     return *this;

```

```
foo.cpp x  foo.h x  main.cpp x  hex_numb.cpp x  hex_numb.h x  CMakeLists.txt x
Q:  Cc W *  0 results  ↑ ↓  + -  ☰ ☷
98     }
99     else {
100         return hex1;
101     }
102 }
103
104 Numeral_16 Numeral_16::operator++(int) {
105     toAdd++;
106     hex_ = hexAdd( hex1: hex_ , hex2: to_string( val: toAdd));
107     return *this;
108 }
109
110
111
112 Numeral_16 Numeral_16::operator+(const Numeral_16 obj) {
113     string d1 = this->hexAdd( hex1: hex_ , hex2: obj.hex_);
114     Numeral_16 test = Numeral_16( hex_: d1);
115     return test;
116 }
117 Numeral_16 Numeral_16::operator+=(int const number) {
118     hex_ = hexAdd( hex1: hex_ , hex2: to_string( val: number));
119     return *this;
120 }
121
```

Тестування :

C++

```
lab4 x
C:\Op-1.2\2-sem\lab\lab4\cmake-build-debug\lab4.exe
Enter hex number for 1 item :
15A
First (1) :Your number is :15A
Second (2) :Your number is :15A
Third (3) :Your number is :
How much do you want to increase N2? :
3
-----After changing-----
First (1) :Your number is :15B
Second (2) :Your number is :15D
Third (3) :Your number is :2B8

Hexadecimal to Decimal (N3) :696

Process finished with exit code 0
```

Висновок : При виконанні лабораторної роботи було вивчено особливості роботи з класами та об'єктами та використання перевантажених операторів . Також було створено та використано три різні конструктори для створення об'єктів класу .