

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»

«Перевантаження операторів»

Варіант 28

Виконав студент ІП-13, Петров Ігор Ярославович
(шифр, прізвище, ім'я, по батькові)

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

Київ 2022

Лабораторна робота № 4

Варіант 28

28. Визначити клас "Numeral_16", членом якого є шістнадцяткове число. Реалізувати для даного класу декілька конструкторів, геттери, методи перетворення числа у двійкове, у тому числі і скороченим способом. Перевантажити оператори: префіксний "++" – 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 двійковий формат двома способами (звичайним і скороченим).

Код C++

main.cpp

```
#include "func.h"

int main() {
    int n1;
    string n2;
    cout << "Enter N1 in decimal format: "; cin >> n1;
    cout << "Enter N2 in hex format: "; cin >> n2;
    Numeral_16 N1(n1), N2(n2), N3;
    cout << "N1 in hex format: " << N1.get_num_16() << endl;
    cout << "N2 in hex format: " << N2.get_num_16() << endl;
    ++N1;
    cout << "Incremented N1 in hex format: " << N1.get_num_16() << endl;
    string num_to_add;
    string mode = choose_mode();
    cout << "Enter a number to increase N2: "; cin >> num_to_add;
    if (mode == "h") N2 += num_to_add;
    else N2 += stoi(num_to_add);
    cout << "Increased N2 in hex format: " << N2.get_num_16() << "\n";
    N3 = N1 + N2;
    cout << "N3 in hex format(sum of N1 & N2): " << N3.get_num_16() << endl;
    cout << "N3 in bin format(using long method): " << N3.bin_long() << endl;
    cout << "N3 in bin format(using short method): " << N3.bin_short() << endl;
}
```

func.cpp

```
#include "func.h"

Numeral_16::Numeral_16(int dec_num) {
    string temp;
    int temp_int;
    while (dec_num) {
        temp_int = dec_num % 16;
        if (temp_int < 10) temp += (char)(temp_int + 48);
        else temp += (char)(temp_int + 55);
        dec_num /= 16;
    }
    for (int i = temp.length() - 1; i >= 0; i--) {
        num_16 += temp[i];
    }
}

Numeral_16::Numeral_16(const string& hex_num) {
```

```

    this->num_16 = hex_num;
}

string Numeral_16::get_num_16() {
    return num_16;
}

Numeral_16 Numeral_16::operator ++() {
    int dec_num = stoi(num_16, 0, 16);
    this->num_16 = Numeral_16(++dec_num).num_16;
    return *this;
}

const Numeral_16 Numeral_16::operator +(const Numeral_16& obj) {
    return Numeral_16(stoi(num_16, 0, 16) + stoi(obj.num_16, 0, 16)).num_16;
}

const Numeral_16 Numeral_16::operator +=(const string& hex_num) {
    num_16 = Numeral_16(stoi(num_16, 0, 16) + stoi(hex_num, 0, 16)).num_16;
    return *this;
}

const Numeral_16 Numeral_16::operator +=(int dec_num) {
    num_16 = Numeral_16(stoi(num_16, 0, 16) + dec_num).num_16;
    return *this;
}

string Numeral_16::bin_short() {
    string bin_num;
    for (char c : num_16) {
        switch (c) {
            case '0':
                bin_num += "0000";
                break;
            case '1':
                bin_num += "0001";
                break;
            case '2':
                bin_num += "0010";
                break;
            case '3':
                bin_num += "0011";
                break;
            case '4':
                bin_num += "0100";
                break;
            case '5':
                bin_num += "0101";
                break;
            case '6':
                bin_num += "0110";
                break;
            case '7':
                bin_num += "0111";
                break;
            case '8':
                bin_num += "1000";
                break;
            case '9':
                bin_num += "1001";
                break;
            case 'A':
            case 'a':
                bin_num += "1010";
                break;
            case 'B':
            case 'b':
                bin_num += "1011";
                break;
        }
    }
}

```

```

        break;
    case 'C':
    case 'c':
        bin_num += "1100";
        break;
    case 'D':
    case 'd':
        bin_num += "1101";
        break;
    case 'E':
    case 'e':
        bin_num += "1110";
        break;
    case 'F':
    case 'f':
        bin_num += "1111";
        break;
    }
}
bin_num.erase(0, bin_num.find_first_not_of('0'));
return bin_num;
}

string Numeral_16::bin_long() {
    int dec_num = stoi(num_16, nullptr, 16);
    string bin_num;
    string temp;
    int temp_int;
    while (dec_num) {
        temp_int = dec_num % 2;
        temp += (char)(temp_int + 48);
        dec_num /= 2;
    }
    for (int i = temp.length() - 1; i >= 0; i--) {
        bin_num += temp[i];
    }
    return bin_num;
}

string choose_mode() {
    string mode;
    cout << "Enter number format to add to N2.\n'h' for hex format or 'd' for decimal format: " ; cin >>
mode;
    while (mode != "h" and mode != "d") {
        cout << "Incorrect input. Enter 'h' or 'd': "; cin >> mode;
    }
    return mode;
}
}

```

Header.h

```

#pragma once
#include <iostream>
#include <string>
#include <cmath>

using namespace std;

class Numeral_16 {
private:
    string num_16;
public:
    Numeral_16() = default;
    Numeral_16(int);
    Numeral_16(const string&);
}

```

```
string get_num_16();
Numeral_16 operator ++();
const Numeral_16 operator +(const Numeral_16&);
const Numeral_16 operator += (const string&);
const Numeral_16 operator += (int);
string bin_short();
string bin_long();
};

string choose_mode();
```

Результат роботи

```
Консоль отладки Microsoft Visual Studio
Enter N1 in decimal format: 10
Enter N2 in hex format: A
N1 in hex format: A
N2 in hex format: A
Incremented N1 in hex format: B
Enter number format to add to N2.
'h' for hex format or 'd' for decimal format: d
Enter a number to increase N2: 10
Increased N2 in hex format: 14
N3 in hex format(sum of N1 & N2): 1F
N3 in bin format(using long method): 11111
N3 in bin format(using short method): 11111
```