

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 № 3 з дисципліни
«Основи програмування 2. Модульне
програмування»

«ПЕРЕВАНТАЖЕННЯ ОПЕРАТОРІВ»

Варіант__30__

Виконав студент _____ ІП-15 Розін Олексій Іванович _____
(шифр, прізвище, ім'я, по батькові)

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

Київ 2021__

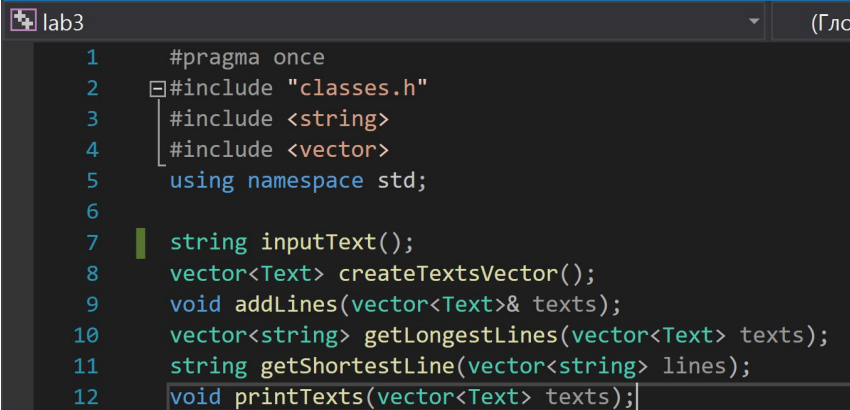
Завдання

30. Визначити клас "Roman_numerals", який задає число в римській формі запису (у вигляді рядка). Реалізувати для даного класу декілька конструкторів, геттери, методи перетворення числа у десяткове. Перевантажити оператори: префіксний "++" – для інкрементації даного числа, "+=" – для збільшення його на вказану величину (римське число), "+" – для додавання двох римських чисел. Створити три римських числа (R1, R2, R3), використовуючи різні конструктори. Інкрементувати число R1, а число R2 збільшити на вказану величину. Знайти суму змінених чисел R1 та R2 і зберегти її в R3. Перевести отримане значення N3 у десятковий формат.

Код C++

Lab4.cpp

Functions.h



```
lab3 (Гло
1  #pragma once
2  #include "classes.h"
3  #include <string>
4  #include <vector>
5  using namespace std;
6
7  string inputText();
8  vector<Text> createTextsVector();
9  void addLines(vector<Text>& texts);
10 vector<string> getLongestLines(vector<Text> texts);
11 string getShortestLine(vector<string> lines);
12 void printTexts(vector<Text> texts);
```

Functions.cpp

```
lab4 (Глобальная область)
1  #include "Functions.h"
2
3  int rtoi(char roman) {
4      switch (roman) {
5          case 'I': return 1;
6          case 'V': return 5;
7          case 'X': return 10;
8          case 'L': return 50;
9          case 'C': return 100;
10         case 'D': return 500;
11         case 'M': return 1000;
12     }
13 }
14
15 string convertToRoman(int arabic) {
16     string res = "";
17     int arabics[] = {1000, 900, 500, 400, 100, 90, 50, 40, 10, 9, 5, 4, 1};
18     string romans[] = { "M", "CM", "D", "CD", "C", "XC", "L", "XL", "X", "IX", "V", "IV", "I" };
19     for (size_t i = 0; i < 13; i++) {
20         while (arabic >= arabics[i]) {
21             arabic -= arabics[i];
22             res += romans[i];
23         }
24     }
25     return res;
26 }
27
28 int convertToArabic(string roman) {
29     int prev = 0;
30     int res = 0;
31     for (int i = roman.length() - 1; i >= 0; i--) {
32         int myNum = rtoi(roman[i]);
33         if (myNum >= prev) {
34             res += myNum;
35         }
36         else {
37             res -= myNum;
38         }
39         prev = myNum;
40     }
41     return res;
42 }
```

RomanNumber.h

```
lab4 (Глобальная область)
1  #pragma once
2  #include <string>
3  using namespace std;
4
5  class RomanNumber {
6  private:
7      string romanValue;
8      int arabicValue;
9  public:
10     RomanNumber(string roman);
11     RomanNumber(int arabic);
12     RomanNumber();
13
14     string getRomanValue();
15     int getArabicValue();
16
17     RomanNumber operator++();
18     RomanNumber operator+=(string roman2);
19     RomanNumber operator+(RomanNumber roman2);
20
21     void printNumber();
22 };
```

RomanNumber.cpp

```
lab4 → RomanNumber
1  #include <iostream>
2  #include <string>
3  #include "RomanNumber.h";
4  #include "Functions.h"
5  using namespace std;
6
7  RomanNumber::RomanNumber(string roman) {
8      this->romanValue = roman;
9      this->arabicValue = convertToArabic(roman);
10 }
11
12 RomanNumber::RomanNumber(int arabic) {
13     this->arabicValue = arabic;
14     this->romanValue = convertToRoman(arabic);
15 }
16
17 RomanNumber::RomanNumber() {
18     this->romanValue = "I";
19     this->arabicValue = 1;
20 }
21
22 string RomanNumber::getRomanValue() {
23     return this->romanValue;
24 }
25
26 int RomanNumber::getArabicValue() {
27     return this->arabicValue;
28 }
29
30 RomanNumber RomanNumber::operator++() {
31     ++this->arabicValue;
32     this->romanValue = convertToRoman(this->arabicValue);
33     return *this;
34 }
35
36 RomanNumber RomanNumber::operator+=(string roman2){
37     this->arabicValue += convertToArabic(roman2);
38     this->romanValue = convertToRoman(this->arabicValue);
39     return *this;
40 }
41
42 RomanNumber RomanNumber::operator+(RomanNumber roman2) {
43     return RomanNumber(this->arabicValue + roman2.arabicValue);
44 }
45
46 void RomanNumber::printNumber() {
47     cout << "Roman presentation of the number: " << this->romanValue;
48     cout << "\nArabic presentation of the number: " << this->arabicValue << "\n";
49 }
```

Консоль

C++

Консоль отладки Microsoft Visual Studio

```
R1:
Roman presentation of the number: XXIV
Arabic presentation of the number: 24

R2:
Roman presentation of the number: XV
Arabic presentation of the number: 15

R3:
Roman presentation of the number: I
Arabic presentation of the number: 1

R1 after incrementation:
Roman presentation of the number: XXV
Arabic presentation of the number: 25

R2 after incrementation:
Roman presentation of the number: XXII
Arabic presentation of the number: 22

R3 as a sum of R1 and R2:
Roman presentation of the number: XLVII
Arabic presentation of the number: 47
```

Висновок

Я вивчив механізм створення класів з використанням перевантажених операторів. В даній лабораторній роботі я створив клас, який задає число в римському форматі, перевантажив оператор префіксного інкрементування, оператор “+=” та оператор додавання. Створив 3 об’єкта класа різними конструкторами, змінив значення перших двох за допомогою перевантажених операторів, та зберіг значення їх суми у третій об’єкт.