Лабораторные работы по курсу

"Объектно-ориентированный анализ и проектирование".

Лабораторная работа №6

**Вариант 3**

Выполнили студенты  
гр. КТбо2-6

Бояринов В.В.

Ландин Ю.В.

Таганрог 2019

Оглавление

[Задание 1](#_Toc528784612)

[Используемые математические зависимости и алгоритмы 1](#_Toc528784613)

[Диаграмма классов 2](#_Toc528784614)

[Перечень сигнатур методов с комментариями 3](#_Toc528784615)

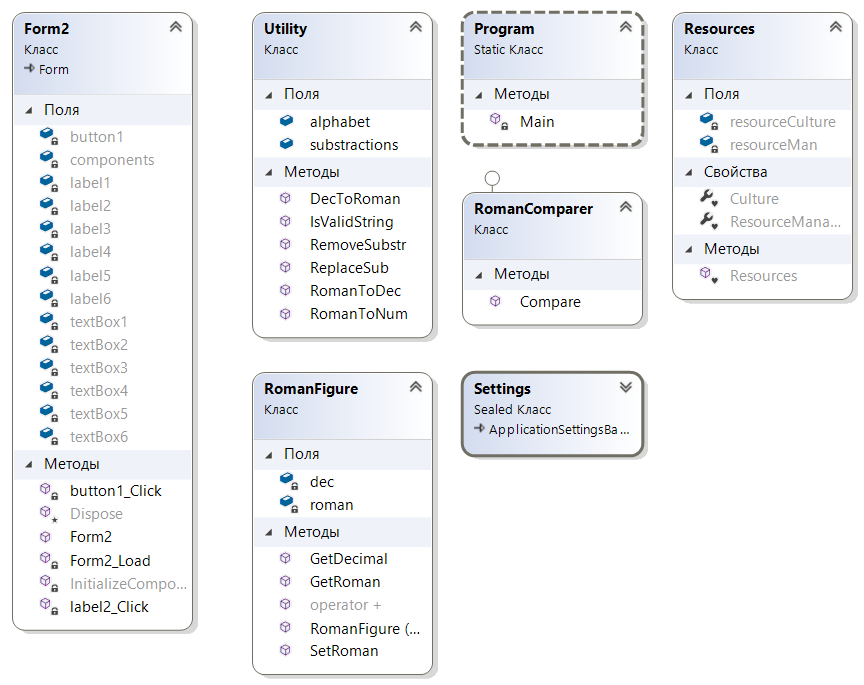
[Листинг: классы, методы, main 4](#_Toc528784616)

[Контрольные вопросы 12](#_Toc528784617)

# Задание

Разработайте графический интерфейс для проекта, описанного ниже. Язык и шаблон приложения выберите сами.  
 Определить класс римских чисел RomanFigure. Вычисления над входными римскими числами должны выполняться в той же системе. Одновременно для контроля входные и выходные числа переводятся в 10-ю систему.

# Диаграмма классов



# Перечень сигнатур методов с комментариями

* Методы класса RomanFigure
  + - public RomanFigure() //конструктор с заполнением поля «roman»
    - public string GetRoman() // Возвращает римское число
    - public int GetDecimal() // Возвращает десятичное число
    - public void SetRoman(string str)//Изменение значения
    - public static RomanFigure operator +(RomanFigure temp1, RomanFigure temp2)//сложение двух римских чисел с перегрузкой оператора
* Методы класса Utility
  + - public static int RomanToNum(char str) // перевод римского числа в числовое представление
    - public static string ReplaceSub(string str) // перевод римского числа к виду без вычитания
    - public static string DecToRoman(int a) // перевод десятичного числа в римское
    - public static int RomanToDec(string str) // Перевод из римского числа в десятичное
    - public static string RemoveSubstr(string a) // убирает все вычитания из римской записи числа
    - public static bool IsValidString(string str) // проверка, все ли символы строки есть в алфавите
* Методы класса RomanComparer
  + - public int Compare(char c1, char c2) // сравнение двух римских символов

Листинг: классы, методы, main

//Utility.cs

using System.Collections.Generic;

using System.Linq;

namespace lab\_6\_oop

{

class Utility

{

public static int RomanToNum(char str) // перевод римского символа в числовое представление

{

if (str == 'I')

return 1;

else if (str == 'V')

return 5;

else if (str == 'X')

return 10;

else if (str == 'L')

return 50;

else if (str == 'C')

return 100;

else if (str == 'D')

return 500;

else if (str == 'M')

return 1000;

else

return 0;

}

public static string ReplaceSub(string str) // перевод римского числа к виду без вычитания

{

if (str == "IV")

return "IIII";

else if (str == "IX")

return "VIIII";

else if (str == "XL")

return "XXXX";

else if (str == "XC")

return "LXXXX";

else if (str == "CD")

return "CCCC";

else return "DCCCC";

}

public static string DecToRoman(int a) // перевод десятичного числа в римское

{

if (a == 0)

return "I";

else if (a == 5)

return "V";

else if (a == 10)

return "X";

else if (a == 50)

return "L";

else if (a == 100)

return "C";

else if (a == 500)

return "D";

else

return "M";

}

public static int RomanToDec(string str) // Перевод из римского числа в десятичное

{

int buff = 0;

int num = 0;

for (int i = 0; i < str.Length; i++)

{

if ((i > 0) && (RomanToNum(str[i]) > RomanToNum(str[i - 1])))

{

num += RomanToNum(str[i]) - buff;

buff = 0;

}

else

{

num += buff;

buff = 0;

buff += RomanToNum(str[i]);

}

}

num += buff;

return num;

}

public static string[] substractions = { "IV", "IX", "XL", "XC", "CD", "CM" };

public static string alphabet = "IVXLCDM";

public static string RemoveSubstr(string a) // убирает все вычитания из римской записи числа

{

string new\_a = a;

for (int i = 0; i < 6; i++)

{

new\_a = new\_a.Replace(substractions[i], ReplaceSub(substractions[i]));

}

return new\_a;

}

public static bool IsValidString(string str) // проверка, все ли символы строки есть в алфавите

{

foreach (char c in str)

{

if (!alphabet.Contains(c))

return false;

}

return true;

}

}

class RomanComparer: IComparer<char>

{

public int Compare(char c1, char c2) // сравнение двух римских символов

{

if (Utility.RomanToNum(c1) > Utility.RomanToNum(c2))

return 1;

else if (Utility.RomanToNum(c1) < Utility.RomanToNum(c2))

return -1;

else

return 0;

}

}

}

// RomanFigure.cs

using System;

namespace lab\_6\_oop

{

class RomanFigure

{

string roman;

int dec;

public RomanFigure() {}

public RomanFigure(string str)

{

roman = str;

dec = Utility.RomanToDec(str);//в конструкторе сразу же происходит перевод из римского в десятичное

}

public string GetRoman() // Возвращает римское число

{

return this.roman;

}

public int GetDecimal() // Возвращает десятичное число

{

return this.dec;

}

public void SetRoman(string str)//Изменение значения

{

this.roman = str;

this.dec = Utility.RomanToDec(str);

}

//RomanFigure add(RomanFigure &temp);//сложение двух римских чисел

public static RomanFigure operator +(RomanFigure temp1, RomanFigure temp2)//сложение двух римских чисел с перегрузкой оператора

{

string temp\_roman = Utility.RemoveSubstr(temp2.GetRoman());

string str\_temp = Utility.RemoveSubstr(temp1.GetRoman());

string new\_roman = "";

temp\_roman += str\_temp;

char[] tmpch = temp\_roman.ToCharArray();

Array.Sort(tmpch, new RomanComparer());

temp\_roman = new string(tmpch);

int ones, tens, hundreds, thousands;

ones = tens = hundreds = thousands = 0;

for (int i = 0; i < temp\_roman.Length; i++)

{

switch (temp\_roman[i])

{

case 'I': ones += 1; break;

case 'V': ones += 5; break;

case 'X': tens += 1; break;

case 'L': tens += 5; break;

case 'C': hundreds += 1; break;

case 'D': hundreds += 5; break;

case 'M': thousands += 1; break;

}

}

tens += ones / 10;

ones %= 10;

hundreds += tens / 10;

tens %= 10;

thousands += hundreds / 10;

hundreds %= 10;

new\_roman += new string('M', thousands);

if (hundreds == 9)

{

new\_roman += "CM";

}

else if (hundreds == 4)

{

new\_roman += "CD";

}

else

{

new\_roman += new string('D', hundreds / 5) + new string('C' ,hundreds % 5);

}

if (tens == 9)

{

new\_roman += "XC";

}

else if (tens == 4)

{

new\_roman += "XL";

}

else

{

new\_roman += new string('L', tens / 5) + new string('X', tens % 5);

}

if (ones == 9)

{

new\_roman += "IX";

}

else if (ones == 4)

{

new\_roman += "IV";

}

else

{

new\_roman += new string('V', ones / 5) + new string('I', ones % 5);

}

return new RomanFigure(new\_roman);

}

}

}

// Form2.cs

using System;

using System.Windows.Forms;

namespace lab\_6\_oop

{

public partial class Form2 : Form

{

public Form2()

{

InitializeComponent();

}

private void Form2\_Load(object sender, EventArgs e) { }

private void button1\_Click(object sender, EventArgs e)

{

string s1 = (textBox1.Text);

if (!Utility.IsValidString(s1))

s1 = "0";

string s2 = (textBox2.Text);

if (!Utility.IsValidString(s2))

s2 = "0";

RomanFigure rom1 = new RomanFigure(s1), rom2 = new RomanFigure(s2);

RomanFigure sum;

sum = rom1 + rom2;

textBox3.Text = rom1.GetDecimal().ToString();

if (textBox3.Text == "0")

textBox3.Text = "error";

textBox4.Text = rom2.GetDecimal().ToString();

if (textBox4.Text == "0")

textBox4.Text = "error";

textBox5.Text = sum.GetRoman();

textBox6.Text = sum.GetDecimal().ToString();

}

private void label2\_Click(object sender, EventArgs e)

{

}

}

}

//Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace lab\_6\_oop

{

static class Program

{

/// <summary>

/// Главная точка входа для приложения.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form2());

}

}

}

# Контрольные вопросы

Вопрос, собственно говоря, один и формулируется он следующим образом: покажите, как вы все это сделали.

Автор Преподаватель

Бояринов В.В. \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Оценка\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ландин Ю.В. \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Оценка\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Подпись\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 29.03.2019