ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Санкт – Петербургский государственный университет телекоммуникаций

им. проф. М.А. Бонч-Бруевича»

Санкт-Петербургский колледж телекоммуникаций

##### Лабораторно-практическая работа № 18

по профессиональному модулю

**ПМ.01 «ПРИКЛАДНОЕ ПРОГРАММИРОВАНИЕ»**

Студент гр. 590\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. Ю. Яскунова

(подпись)

Проверил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. В. Параничев

(оценка и подпись)

Санкт-Петербург

2021 год

ФОРМАТНЫЙ ВЫВОД ТЕКСТОВЫХ ДАННЫХ В ПРИЛОЖЕНИИ WINDOWS FORMS

Код консольного приложения представлен с помощью онлайн-компилятора OnlineGDB (<https://onlinegdb.com/B1VwKsw7O> ):

// calculator\_59028.cs

using System;

namespace ClassLib\_Calc\_59028

{

public class Calculator

{

public static string RShift(int first\_num, int second\_num)

{

string str\_ret = "Результат вычисления: ";

str\_ret += String.Format("\n1й параметр: {0,31}", first\_num);

str\_ret += String.Format("\n2й параметр: {0,31}", second\_num);

int numb = first\_num >> second\_num;

if (Math.Abs(numb) > 150)

str\_ret += "\nОшибка #322! Число больше 150 по модулю.";

else

str\_ret += String.Format("\nТочный результат: {0,31} \nРезультат в 16-ричной системе счисления: {1, 10}", numb.ToString(), Convert.ToString(numb,16));

return str\_ret;

}

public static string RShift(ref int first\_num, int second\_num)

{

string str\_ret = "Результат вычисления: ";

str\_ret += String.Format("\n1й параметр: {0,31}", first\_num);

str\_ret += String.Format("\n2й параметр: {0,31}", second\_num);

int numb = first\_num >>= second\_num;

if (Math.Abs(numb) > 150)

str\_ret += "\nОшибка #322! Число больше 150 по модулю.";

else

str\_ret += String.Format("\nТочный результат: {0,31} \nРезультат в 16-ричной системе счисления: {1, 10}", numb.ToString(), Convert.ToString(numb,16));

return str\_ret;

}

public static string Xor(int first\_num, int second\_num)

{

string str\_ret = "Результат вычисления: ";

str\_ret += String.Format("\n1й параметр: {0,31}", first\_num);

str\_ret += String.Format("\n2й параметр: {0,31}", second\_num);

int numb = first\_num ^ second\_num;

if (Math.Abs(numb) > 150)

str\_ret += "\nОшибка #322! Число больше 150 по модулю.";

else

str\_ret += String.Format("\nТочный результат: {0,31} \nРезультат в 16-ричной системе счисления: {1, 10}", numb.ToString(), Convert.ToString(numb,16));

return str\_ret;

}

public static string Xor(ref int first\_num, int second\_num)

{

string str\_ret = "\nРезультат вычисления: ";

str\_ret += String.Format("\n1й параметр: {0,31}", first\_num);

str\_ret += String.Format("\n2й параметр: {0,31}", second\_num);

int numb = first\_num ^= second\_num;

if (Math.Abs(numb) > 150)

str\_ret += "\nОшибка #322! Число больше 150 по модулю.";

else

str\_ret += String.Format("\nТочный результат: {0,31} \nРезультат в 16-ричной системе счисления: {1, 10}", numb.ToString(), Convert.ToString(numb,16));

return str\_ret;

}

}

}

// program.cs

//using calc = ClassLib\_Calc\_59028.Calculator;

namespace Program\_59028

{

class Program

{

static void Main()

{

int num = 59028;

// исходные данные для вычислений

string [] op\_sign\_simple = {"^", "\*", "|", "&", "-", "/", "<<", "~", ">>", "!"};

string [] op\_sign\_overload = {"^=", "\*=", "|=", "&=", "-=", "/=", "<<=", "~=", ">>=", "!="};

string [] op\_description = {"передаваемых значений - не более 100",

"передаваемых значений - не менее 50",

"первого параметра значением - не более 90",

"первого параметра значением - не менее 40",

"второго параметра значением - не более 80",

"второго параметра значением - не менее 30",

"результата вычисления - не более 300",

"результата вычисления - не менее -4000",

"результата вычисления - не более 150 по модулю",

"результата вычисления - не менее 50 по модулю"};

// операции, которые нужно реализовать для своего варианта

for(int i = 0; i < 4; i+=2)

{

int n\_simple = ((num / 100) + num % 100 + i)%10;

int n\_overload = (n\_simple + 10)%10;

Console.WriteLine("Операция " + i + ": {0} (разрешить числа {1})",

op\_sign\_simple[n\_simple], op\_description[n\_simple]);

Console.WriteLine("Операция " + i+1 + ": {0} (разрешить числа {1})",

op\_sign\_overload[n\_overload], op\_description[n\_overload]);

}

string str\_out = "";

int first = 1000;

Console.WriteLine("\n###Выполнение функции >> с ошибкой: RShift(1000, 2);");

str\_out = ClassLib\_Calc\_59028.Calculator.RShift(first, 2);

Console.WriteLine(str\_out);

Console.WriteLine("\n###Выполнение функции >>= без ошибок: RShift\_equal(1000, 100);");

str\_out = ClassLib\_Calc\_59028.Calculator.RShift(ref first , 100);

Console.WriteLine(str\_out);

first = 100;

Console.WriteLine("\n###Выполнение функции ^ без ошибок: Xor(100, 50);");

str\_out = ClassLib\_Calc\_59028.Calculator.Xor(first , 50);

Console.WriteLine(str\_out);

Console.WriteLine("\n###Выполнение функции ^= с ошибкой: Xor\_equal(100, 200);");

str\_out = ClassLib\_Calc\_59028.Calculator.Xor(ref first , 200);

Console.WriteLine(str\_out);

}

}

}

В результате получим вывод данных на консоль, представленный на рис.1.

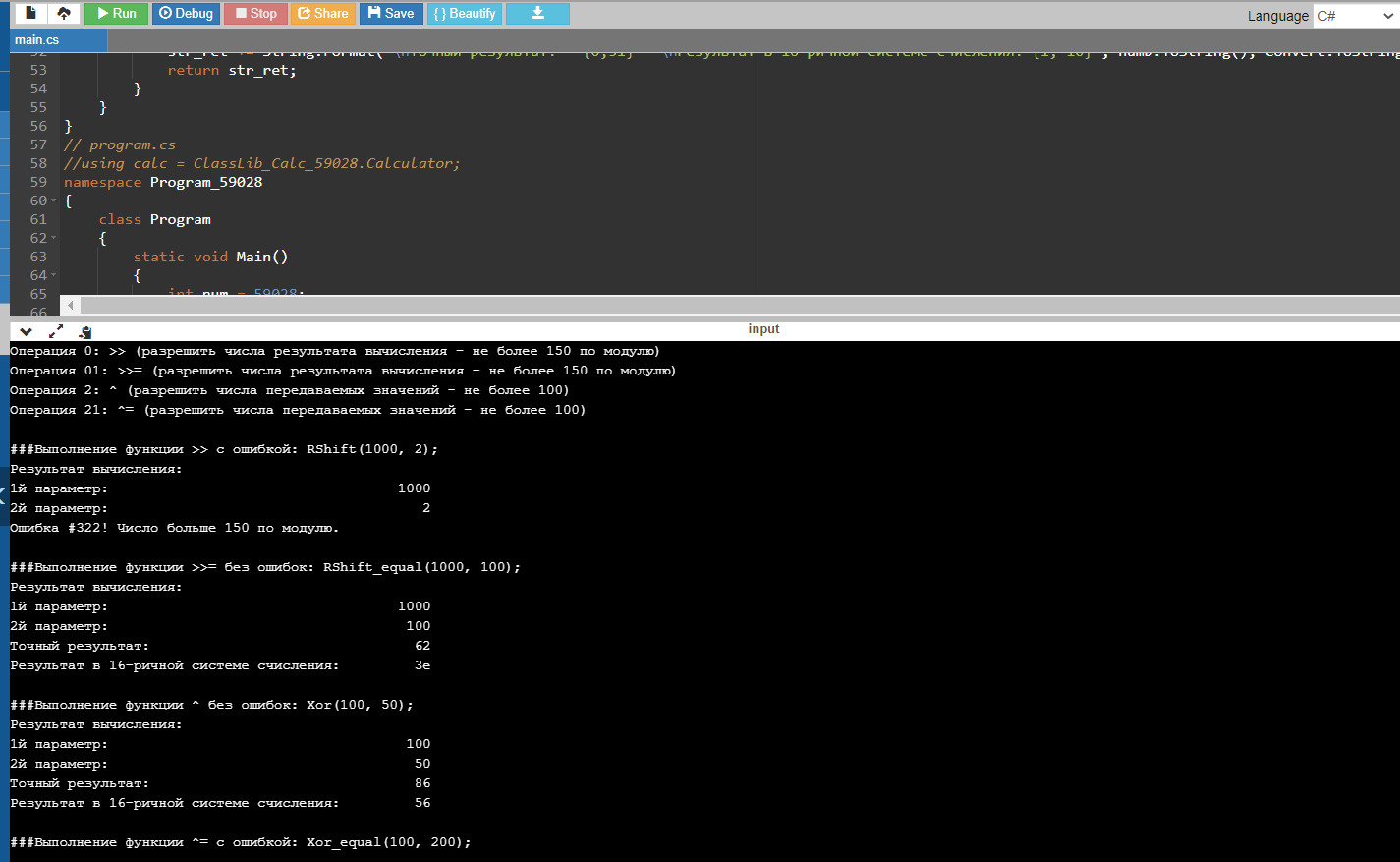


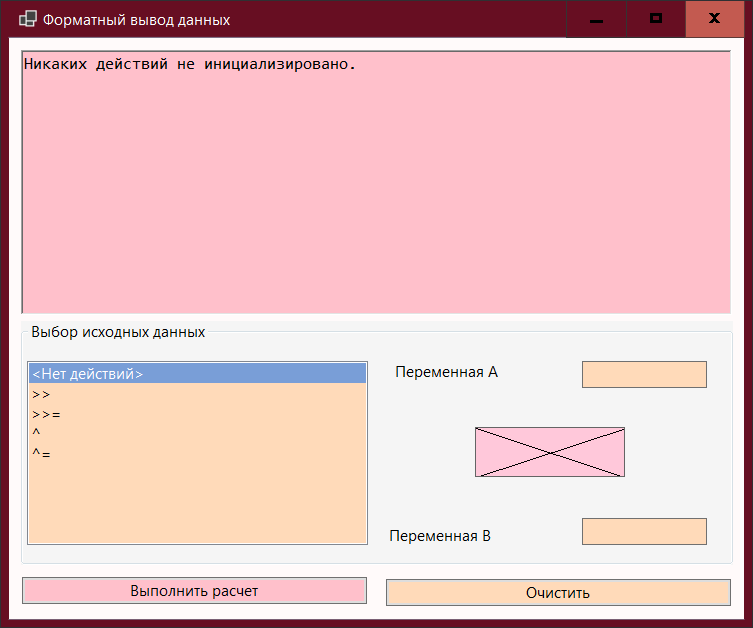
Рис. 1. Результат вывода данных на консоль

Реализуем выполнение этих функций для «виртуальной консоли», используя компоненты RichTextBox и PictureBox в технологии WindowsForms.

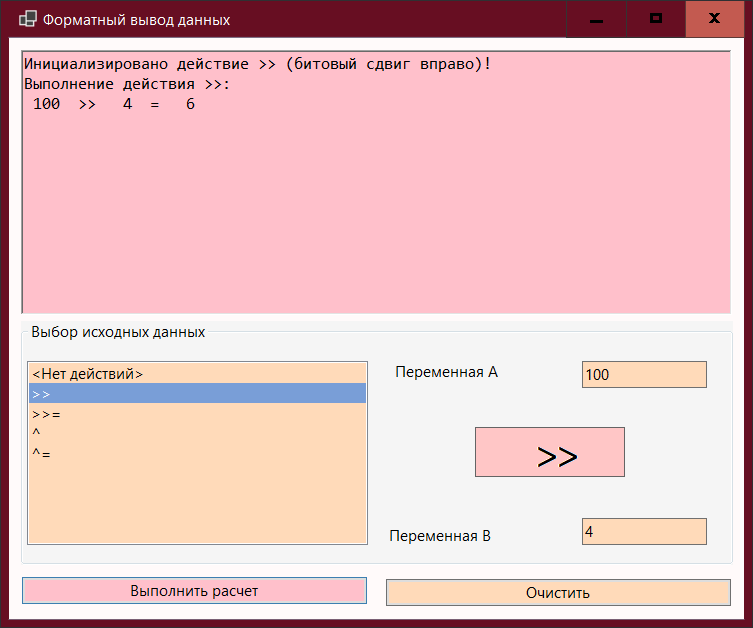
В данной лабораторной работе выполнено следующее:

* создана библиотека классов Operation\_None, Operation\_Rshift, Operation\_Rshift\_equal, Operation\_XOR, Operation\_XOR\_equal.
* подключены ресурсы 4 изображений;
* реализована программа (рис. 1-8), в которой выполнено подключение (встраивание) перечисленных функций, при этом:

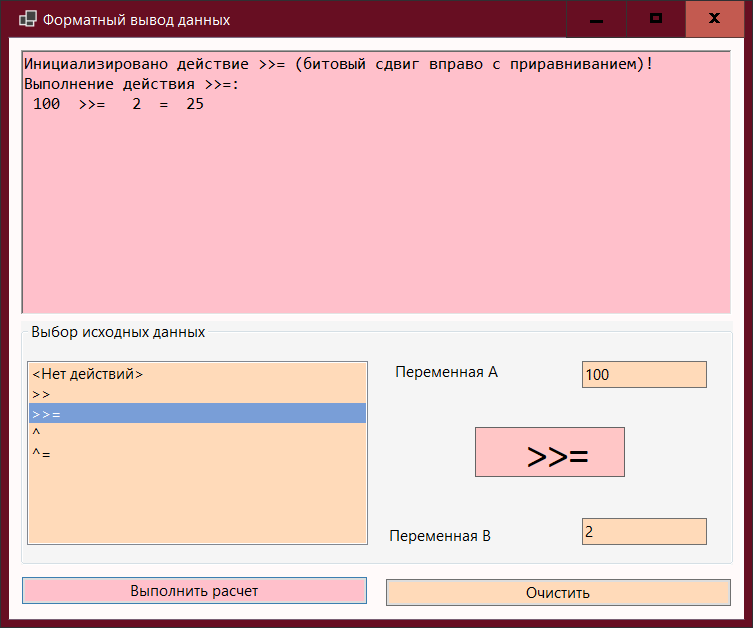
1. реализован функционал выполнения операций над числами (рис. 2-6);
2. подставлены числа, при которых выводится ошибка (рис. 7-8).



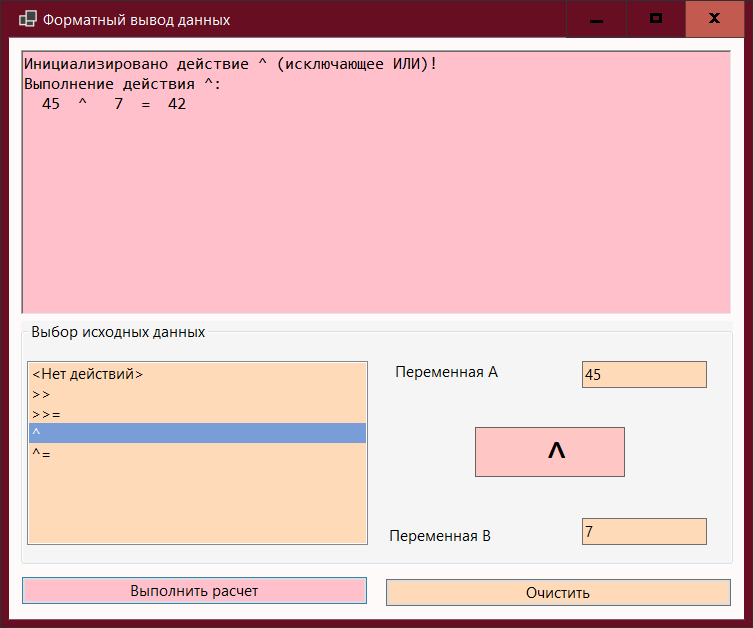
*Рис. 2. Приложение в работе: нет данных*



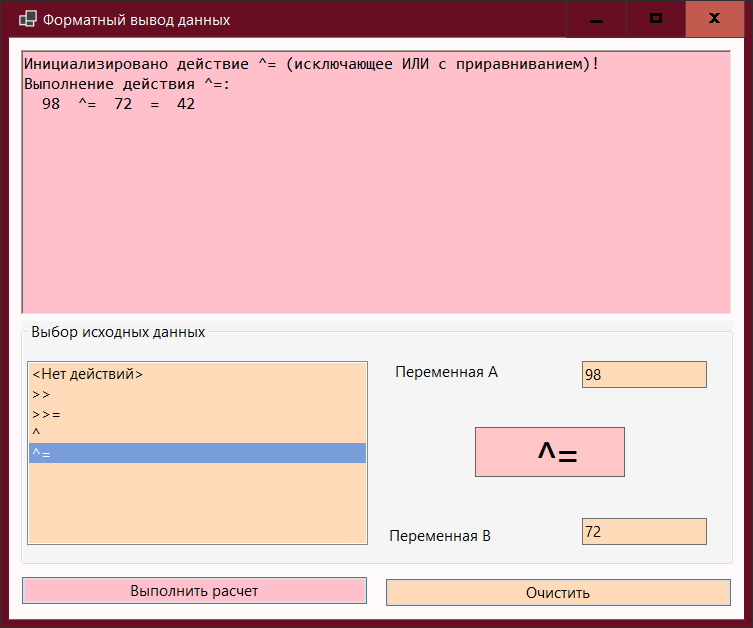
*Рис. 3. Приложение в работе: битоый сдвиг вправо*



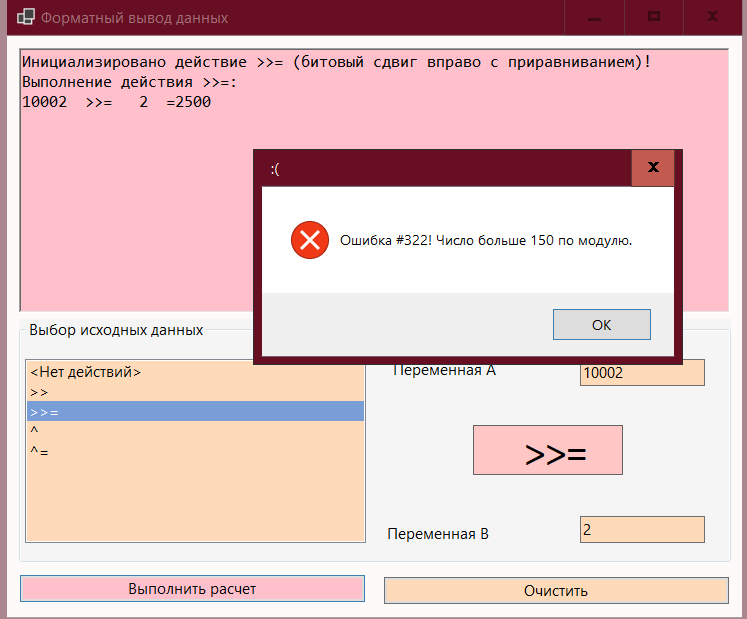
*Рис. 4. Приложение в работе битоый сдвиг вправо с приравниванием*



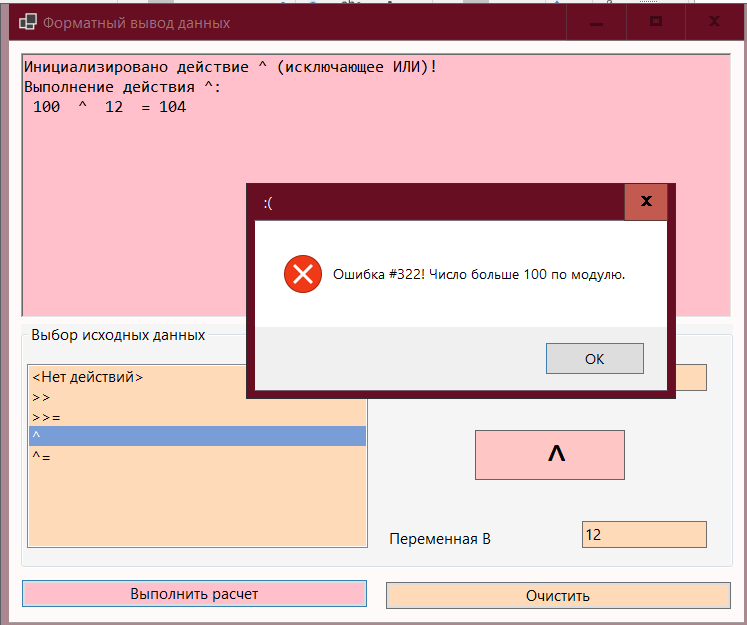
*Рис. 5. Приложение в работе: исключающее ИЛИ*



*Рис. 6. Приложение в работе: исключающее ИЛИ с приравниванием*

**

*Рис. 7. Приложение в работе: ошибка при операции >>=*

**

*Рис. 8. Приложение в работе ошибка при операции ^*

Как видно из рис. 2-8, обеспечена правильность работы программы, в частности:

* на рис. 2: когда нет данных, выводится рисунок крестика и никаких расчетов не производится;
* на рис. 3: когда выбираешь >> и нажимаешь на кнопку происходит битовый сдвиг вправо чисел и вывод картинки;
* на рис. 4: когда выбираешь >>= и нажимаешь на кнопку происходит битовый сдвиг вправо с приравниванием чисел и вывод картинки;
* на рис. 5: когда выбираешь ^ и нажимаешь на кнопку происходит операция «Исключающее ИЛИ» с числами и вывод картинки;
* на рис. 6: когда выбираешь ^= и нажимаешь на кнопку происходит операция «Исключающее ИЛИ»с приравниванием и вывод картинки;
* на рис. 7: когда выбираешь >>= и нажимаешь на кнопку происходит битовый сдвиг вправо с приравниванием чисел и вывод картинки, также выводится окно с ошибкой, т.к. результат >150.
* на рис. 8: когда выбираешь ^ и нажимаешь на кнопку происходит операция «Исключающее ИЛИ» и вывод картинки, также выводится окно с ошибкой, т.к. результат >100.

Таким образом, в данной работы выполнена реализация программы с встраиваемыми ресурсами и вычислениями.