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 № 17

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

РАЗРАБОТКА ПОЛЬЗОВАТЕЛЬСКИХ БИБЛИОТЕЧНЫХ ФУНКЦИЙ НА ПРИМЕРЕ ФУНКЦИОНАЛА КАЛЬКУЛЯТОРА

В данной работе (вариант 59028) выполнено следующее:

* создана библиотека функций RShift(), Xor(), RShift\_equal(), Xor\_equal().
* реализована программа-калькулятор (рис. 1-4), в которой выполнено подключение (встраивание) перечисленных функций.

Код консольного приложения представлен с помощью онлайн-компилятора OnlineGDB (<https://onlinegdb.com/B1zC73LQ_> ):

// 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: ";

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

if (Math.Abs(numb) > 150)

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

else

str\_ret += numb.ToString();

return str\_ret;

}

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

{

string str\_ret = "Результат вычисления: ";

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

if (Math.Abs(numb) > 150)

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

else

str\_ret += numb.ToString();

return str\_ret;

}

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

{

string str\_ret = "Результат вычисления: ";

int numb = first\_num ^ second\_num;

if (Math.Abs(numb) > 100)

str\_ret += "\nОшибка #322! Число больше 100 по модулю.";

else

str\_ret += numb.ToString();

return str\_ret;

}

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

{

string str\_ret = "Результат вычисления: ";

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

if (Math.Abs(numb) > 100)

str\_ret += "\nОшибка #322! Число больше 100 по модулю.";

else

str\_ret += numb.ToString();

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\_equal(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, 3);");

str\_out = ClassLib\_Calc\_59028.Calculator.Xor\_equal(ref first , 3);

Console.WriteLine(str\_out);

}

}

}

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

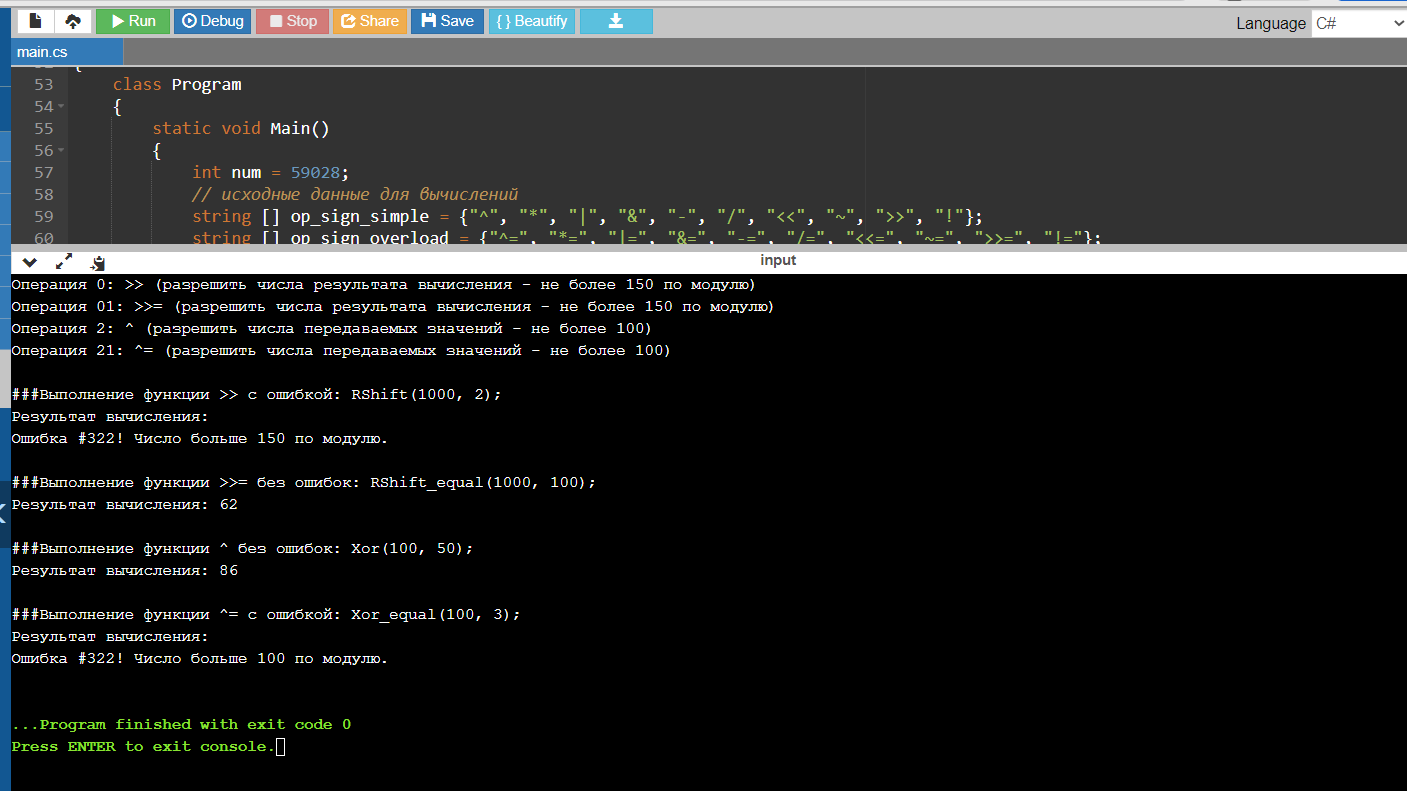


Рис. 1. Результат вывода данных на консоль

Реализуем выполнение этих функций в технологии WindowsForms.

Код управления калькулятора form\_Main.cs:

using System;

using System.Windows.Forms;

using Calc = ClassLib\_Calc\_59028.Calculator;

namespace WinForms\_Calc\_59028

{

public partial class form\_Main : Form

{

public form\_Main()

{

InitializeComponent();

}

private void button\_rightdraf\_Click(object sender, EventArgs e)

{

short \_first\_value = Convert.ToInt16(textBox\_num1.Text);

short \_second\_value = Convert.ToInt16(textBox\_num2.Text);

string str\_message = Calc.RShift(\_first\_value, \_second\_value);

label\_result.Text = str\_message;

}

private void button\_rightdraf\_equal\_Click(object sender, EventArgs e)

{

short \_first\_value = Convert.ToInt16(textBox\_num1.Text);

short \_second\_value = Convert.ToInt16(textBox\_num2.Text);

string str\_message = Calc.RShift\_equal(\_first\_value, \_second\_value);

label\_result.Text = str\_message;

}

private void button\_galka\_equal\_Click(object sender, EventArgs e)

{

short \_first\_value = Convert.ToInt16(textBox\_num1.Text);

short \_second\_value = Convert.ToInt16(textBox\_num2.Text);

string str\_message = Calc.Xor\_equal(\_first\_value, \_second\_value);

label\_result.Text = str\_message;

}

private void button\_galka\_Click(object sender, EventArgs e)

{

short \_first\_value = Convert.ToInt16(textBox\_num1.Text);

short \_second\_value = Convert.ToInt16(textBox\_num2.Text);

string str\_message = Calc.Xor(\_first\_value, \_second\_value);

label\_result.Text = str\_message;

}

private void textBox\_num1\_KeyPress(object sender, KeyPressEventArgs e)

{

if (textBox\_num1.Text == "")

{

if ((e.KeyChar >= '0') && (e.KeyChar <= '9') || (e.KeyChar == '-'))

return;

}

else

{

if ((e.KeyChar >= '1') && (e.KeyChar <= '9') && (textBox\_num1.Text == "-"))

return;

if ((e.KeyChar >= '0') && (e.KeyChar <= '9') && (textBox\_num1.Text != "0") && (textBox\_num1.Text != "-"))

return;

}

if (Char.IsControl(e.KeyChar)) return;

e.HXorled = true;

}

private void textBox\_num2\_KeyPress(object sender, KeyPressEventArgs e)

{

if (textBox\_num2.Text == "")

{

if ((e.KeyChar >= '0') && (e.KeyChar <= '9') || (e.KeyChar == '-'))

return;

}

else

{

if ((e.KeyChar >= '1') && (e.KeyChar <= '9') && (textBox\_num2.Text == "-"))

return;

if ((e.KeyChar >= '0') && (e.KeyChar <= '9') && (textBox\_num2.Text != "0") && (textBox\_num2.Text != "-"))

return;

}

if (Char.IsControl(e.KeyChar)) return;

e.HXorled = true;

}

}

}

Фрагменты работы, соответствующие операциям, представленным на рис.1, показаны:

* на рис.2-3 для операции ‘>>’ (битовый сдвиг вправо)
* на рис. 4-5 для операции “>>=”
* на рис.6-7 для операции “^” (исключающее ИЛИ)
* на рис.8-9 для операции “^=”

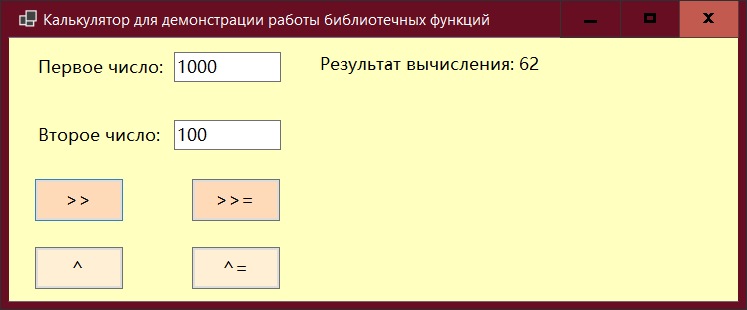
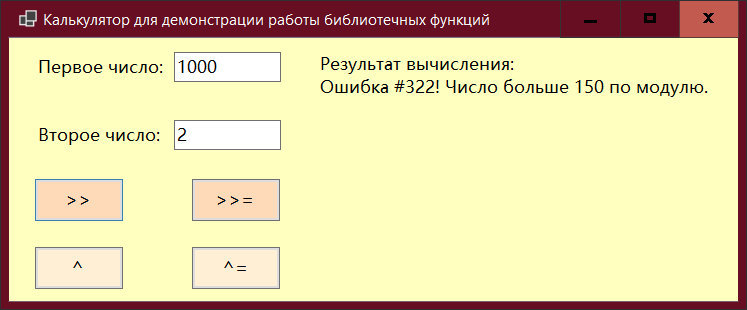


Рис. 2. Результат работы приложения без ошибки при операции >>

Рис. 3. Результат работы приложения с ошибкой #322 при операции >>

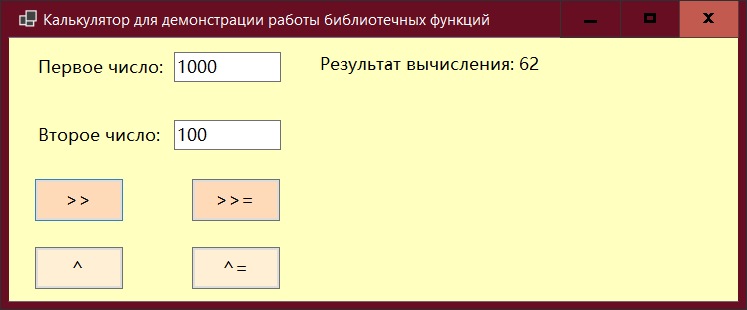


Рис. 4. Результат работы приложения без ошибки при операции >>=

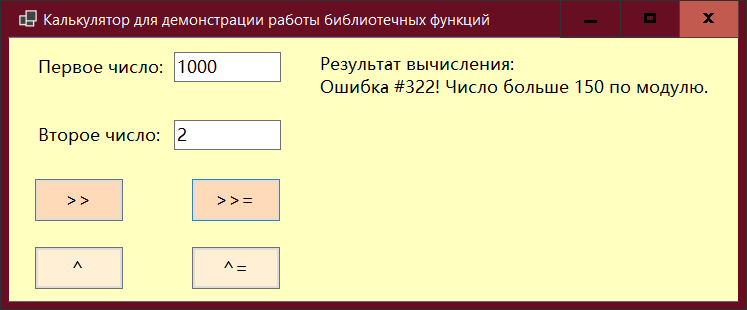
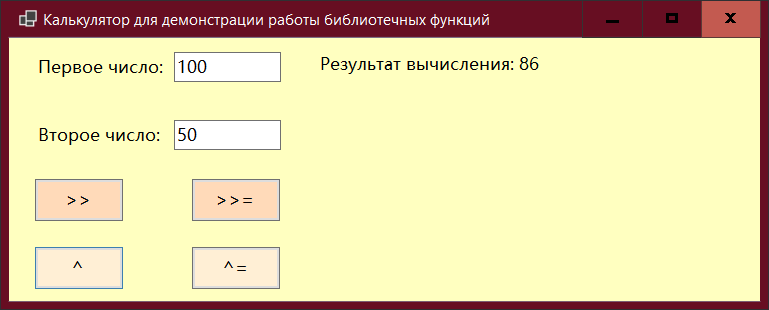


Рис. 5. Результат работы приложения с ошибкой #322 при операции >>=

Рис. 6. Результат работы приложения без ошибки при операции ^

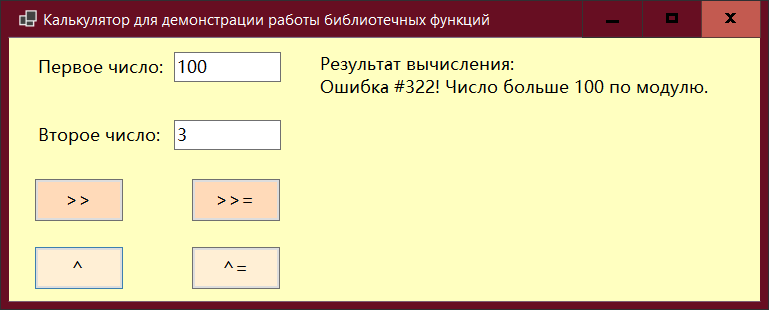


Рис. 7. Результат работы приложения с ошибкой #322 при операции ^

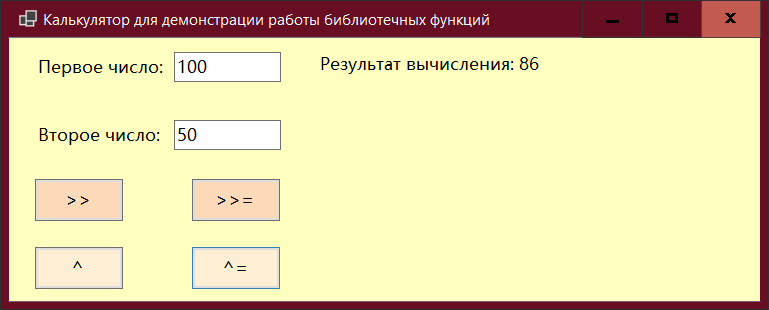
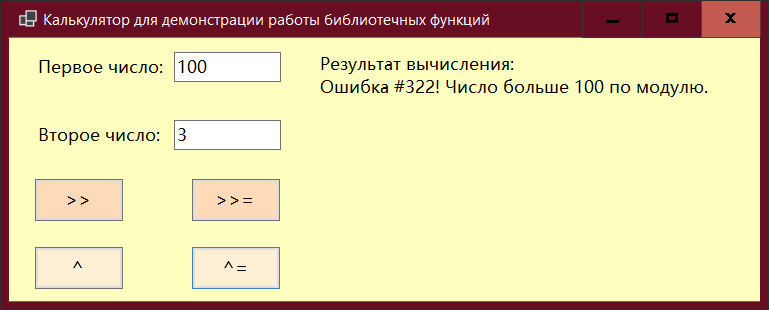


Рис. 8. Результат работы приложения без ошибки при операции ^=

Рис. 9. Результат работы приложения с ошибкой #322 при операции ^=