ФЕДЕРАЛЬНОЕ АГЕНСТВО СВЯЗИ   
ОРДЕНА ТРУДОВОГО КРАСНОГО ЗНАМЕНИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ

«МОСКОВСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ СВЯЗИ И ИНФОРМАТИКИ»

Кафедра информатики

**Лабораторная работа №5**

**по дисциплине: «Языки программирования»**

**Тема: Разработка простой циклической программы «Расчет значения ехр(х)»**

Выполнил студент

группы БИБ1701

Кузьмин К.С.

Проверил

Волков А.И.

Москва

2018 г.

Оглавление

[1. Цель работы 3](#_Toc512651810)

[2. Схемы алгоритмов 4](#_Toc512651811)

[3. Текст программы 11](#_Toc512651812)

[4. Тестирование программы 15](#_Toc512651813)

[5. Скриншоты 16](#_Toc512651814)

# Цель работы

Разработать простую циклическую программу «Расчет значения ехр(х)», которая будет принимать на вход значения от -13 до 20 включительно и считать значение exp(x) с заданной точностью путём разложения функции в ряд. Цикл должен быть с постусловием

Функции:

- Возможность выбирать точность вычисления: 0.1, 0.01, 0.001, 0.0001, 0.00001, 0.000001

- Автоматическое переключение на наименьшую возможную точность

- Защита от дурака в TextBox для значения X

# Схемы алгоритмов



Рис.1 – Блок-схема функции clear



Рис.2 - Схема алгоритма нажатия кнопки " Расчёт exp(x)" часть 1



Рис.3 - Схема алгоритма нажатия кнопки " Расчёт exp(x)" часть 2



Рис.4 – Схема процедуры обработки кнопки «Закрыть»



Рис. 5 – Схема алгоритма обработки открытия формы



Рис. 6 - Первая часть блок-схемы алгоритма "Защиты от дурака" для события txtX\_PressKey



Рис. 7 - Вторая часть блок-схемы алгоритма "Защиты от дурака" для события txtX\_PressKey

# Текст программы

// Пространства имён

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

using namespace System::Globalization;

// Вспомогательные переменные

int x1, n, t;

char ds;

double a, x, y;

Decimal Sum;

//Событие Form1\_Load

private: System::Void Form1\_Load(System::Object^ sender, System::EventArgs^ e) {

NumberFormatInfo^ nfi = NumberFormatInfo::CurrentInfo;

ds = (char)nfi->NumberDecimalSeparator[0];

rad0->Checked = true;

//Функция clear, служащая для очистки Label'ов

int clear (int numL)

{

switch (numL)

{

case 123456:

{

lbl1->Text="";

lbl2->Text="";

lbl3->Text="";

lbl4->Text="";

lbl5->Text="";

lbl6->Text="";

}

case 3456:

{

lbl3->Text="";

lbl4->Text="";

lbl5->Text="";

lbl6->Text="";

}

}

return 0;

}

//Кнопка “Расчёт exp(x)”

int n = 0;

String^ f;

String^ s;

if (this->txtX->Text=="" || this->txtX->Text=="-" || this->txtX->Text=="0,")

{

clear (123456);

x=1;

txtX->Text="1";

}

else

{

x=Convert::ToDouble(this->txtX->Text);

if (x>=-13 && x<=20) // Диапазон Х

{

if (this->rad0->Checked)

f = String::Format("{0:F2}", exp(x));

if (this->rad1->Checked)

f = String::Format("{0:F3}", exp(x));

if (this->rad2->Checked)

f = String::Format("{0:F4}", exp(x));

if (this->rad3->Checked)

f = String::Format("{0:F5}", exp(x));

if (this->rad4->Checked)

f = String::Format("{0:F6}", exp(x));

if (this->rad5->Checked)

f = String::Format("{0:F7}", exp(x));

if (f == "0,00") {

this->rad1->Checked = true;

f = String::Format("{0:F3}", exp(x));

n = 1;

}

if (f == "0,000") {

this->rad2->Checked = true;

f = String::Format("{0:F4}", exp(x));

n = 2;

}

if (f == "0,0000") {

this->rad3->Checked = true;

f = String::Format("{0:F5}", exp(x));

n = 3;

}

if (f == "0,00000") {

this->rad4->Checked = true;

f = String::Format("{0:F6}", exp(x));

n = 4;

}

if (n == 1)

MessageBox::Show("0,01 - наименьшая возможная точность", "Точность",

MessageBoxButtons::OK, MessageBoxIcon::Information);

if (n == 2)

MessageBox::Show("0,001 - наименьшая возможная точность", "Точность",

MessageBoxButtons::OK, MessageBoxIcon::Information);

if (n == 3)

MessageBox::Show("0,0001 - наименьшая возможная точность",

"Точность", MessageBoxButtons::OK, MessageBoxIcon::Information);

if (n == 4)

MessageBox::Show("0,00001 - наименьшая возможная точность",

"Точность", MessageBoxButtons::OK, MessageBoxIcon::Information);

Sum=1+Convert::ToDecimal(x);

y=x;

n=2;

clear (123456);

do

{

// Тело ряда

y=y\*x/n;

Sum=Sum+Convert::ToDecimal(y);

n++;

}

while (abs(y)>a);

this->lbl1->Text = Convert::ToString(x);

this->lbl2->Text = String::Format("{0:#,0.#######}",

a);

this->lbl4->Text = Convert::ToString(n-1);

this->lbl6->Text = String::Format("{0:#,0.#######}",

abs(Math::Exp(x)-Convert::ToDouble(Sum)));

if(rad0->Checked)

{

this->lbl3->Text = String::Format("{0:F2}", exp(x));

this->lbl5->Text = String::Format("{0:F2}", Sum);

}

else if (rad1->Checked)

{

this->lbl3->Text = String::Format("{0:F3}", exp(x));

this->lbl5->Text = String::Format("{0:F3}", Sum);

}

else if (rad2->Checked)

{

this->lbl3->Text = String::Format("{0:F4}", exp(x));

this->lbl5->Text = String::Format("{0:F4}", Sum);

}

else if (rad3->Checked)

{

this->lbl3->Text = String::Format("{0:F5}", exp(x));

this->lbl5->Text = String::Format("{0:F5}", Sum);

}

else if (rad4->Checked)

{

this->lbl3->Text = String::Format("{0:F6}", exp(x));

this->lbl5->Text = String::Format("{0:F6}", Sum);

}

else if (rad5->Checked)

{

this->lbl3->Text = String::Format("{0:F7}", exp(x));

this->lbl5->Text = String::Format("{0:F7}", Sum);

}

}

else

{

clear (123456);

MessageBox::Show(this,"Введите допустимое

значение","Ошибка",MessageBoxButtons::OK,MessageBo

Icon::Warning,MessageBoxDefaultButton::Button1;

}

}

// Защита от дурака для txtX

private: System::Void txtX\_KeyPress(System::Object^ sender,

System::Windows::Forms::KeyPressEventArgs^ e)

{

if (!Char::IsDigit(e->KeyChar))

if (e->KeyChar != '-' || txtX->Text->IndexOf('-') != -1)

if (e->KeyChar != ds || txtX->Text->IndexOf(ds) != -1)

if (e->KeyChar != (char)Keys::Back)

e->Handled = true;

if (e->KeyChar == (char)Keys::D0 ||

e->KeyChar == (char)Keys::NumPad0)

if (txtX->Text->Length >= 1)

if (txtX->Text[0] == '0' && txtX->SelectionStart < 2)

e->Handled = true;

if (e->KeyChar == ds) {

x1 = 0;

if (txtX->Text != "" && txtX->Text[0] == '-') { x1 = 1; }

if (txtX->Text->IndexOf(ds) == -1 &&

txtX->SelectionStart == x1) {

txtX->Text = txtX->Text->Insert(x1, "0");

txtX->SelectionStart = x1 + 1;

}

}

}

# Тестирование программы

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Значение Х | Точность | Точное значение exp(x) | Число слагаемых | Сумма ряда exp(x) | Разность | Вывод |
| 5 | 0.1 | 148.413159 | 15 | 148.37958 | 0.033579 |  |
| 8 | 0.01 | 2 980.957987 | 25 | 2 980.954493 | 0.003494 |  |
| 7.5 | 0.001 | 1 808.042414 | 26 | 1 808.042222 | 0.000193 |  |
| 16 | 0.0001 | 8 886 110.520508 | 51 | 8 886 110.520484 | 0.000024 |  |
| 18,25 | 0.00001 | 84 309 069.231265 | 59 | 84 309 069.231262 | 0.000003 |  |
| 10.111 | 0.000001 | 24 612.260695 | 39 | 24 612.260695 | 0 |  |
| -6 | 0.001 | 0.002479 | 22 | 0.002386 | 0.000093 |  |
| -11,987 | 0.00001 | 0.000006 | 42 | 0.000005 | 0.000001 |  |
| -13.01 | 0.1 |  |  |  |  | Введите допустимое значение. |
| 1702 | 0.000001 |  |  |  |  | Введите допустимое значение. |

# Скриншоты

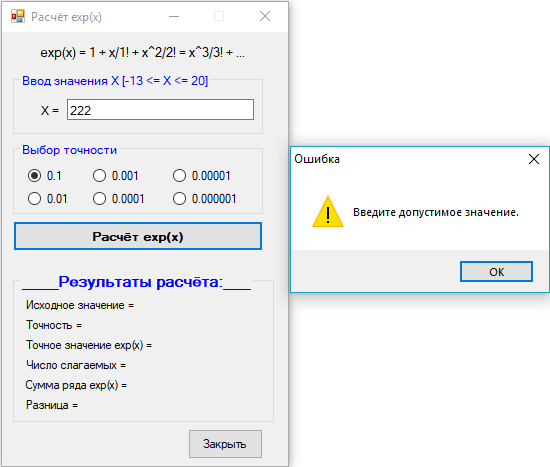


Рис. 8 – Защита от ввода чисел лежащих вне установленного диапазона

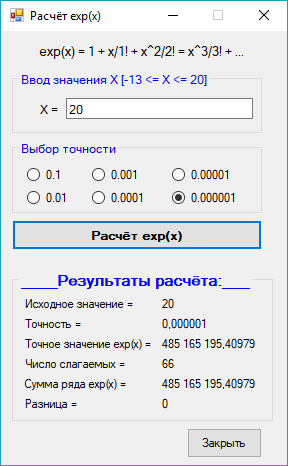
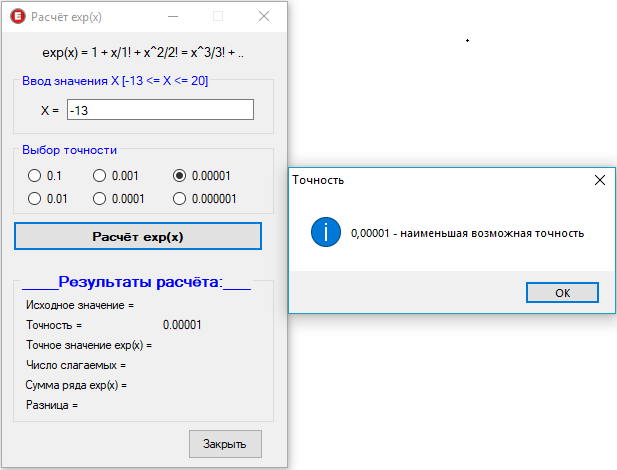


Рис. 9 – exp(20) с точностью 0,000001



На рисунке 10 показана работа автоматического выбора наименьшей возможной точности для данного числа.