Differential Equations Computational practicum

Danila Korneenko B20-04

Variant №21:

$$y' = y/x - y - x$$

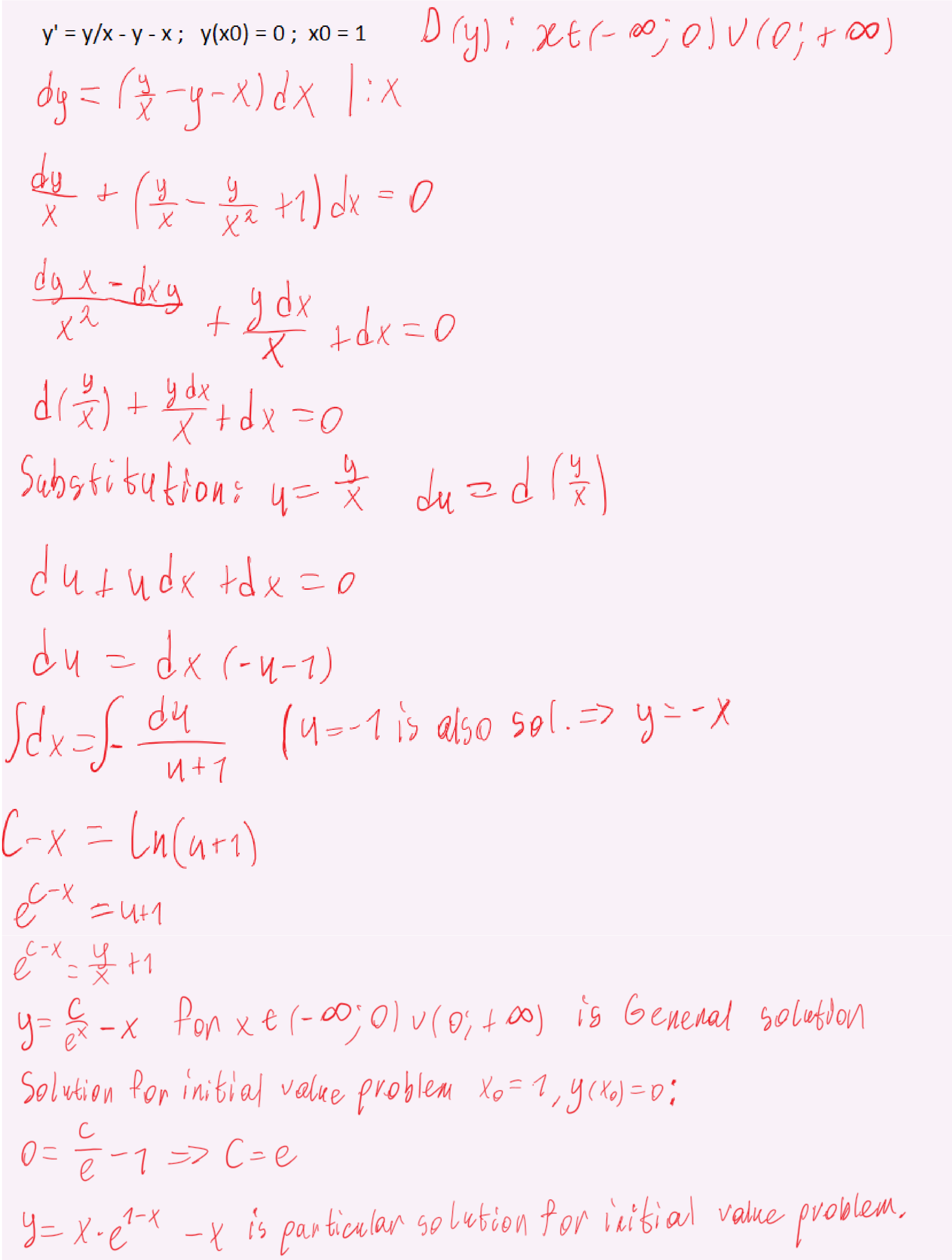
$$y(x_0) = 0$$

$$x_0 = 1$$

$$X = 10$$

Innopolis University 2021

Exact solution of the ODE:



y' = y/x - y - x ; y(x0) = 0 ; x0 = 1 $D(y): x \in (-\infty; 0) \cup (0; +\infty)$

$$dy = \left(\frac{y}{x} - y - x\right) dx \quad \Big| :x$$

$$\frac{dy}{x} + \left(\frac{y}{x} - \frac{y}{x^2} + 1\right) dx = 0$$

$$\frac{dy\,x - dx\,y}{x^2} + \frac{y\,dx}{x} + dx = 0$$

$$d\left(\frac{y}{x}\right) + \frac{y\,dx}{x} + dx = 0$$

Substitution: $u = \frac{y}{x}$ $du = d\left(\frac{y}{x}\right)$

$$du + u\,dx + dx = 0$$

$$du = dx\,(-u-1)$$

$$\int dx = \int -\frac{du}{u+1} \quad (u = -1 \text{ is also sol.} \Rightarrow y = -x$$

$$C - x = \ln(u+1)$$

$$e^{C-x} = u+1$$

$$e^{C-x} = \frac{y}{x} + 1$$

$y = \frac{C}{e^x} - x$ for $x \in (-\infty; 0) \cup (0; +\infty)$ is General solution

Solution for initial value problem $x_0 = 1, y(x_0) = 0$:

$$0 = \frac{C}{e} - 1 \Rightarrow C = e$$

$y = x \cdot e^{1-x} - x$ is particular solution for initial value problem.

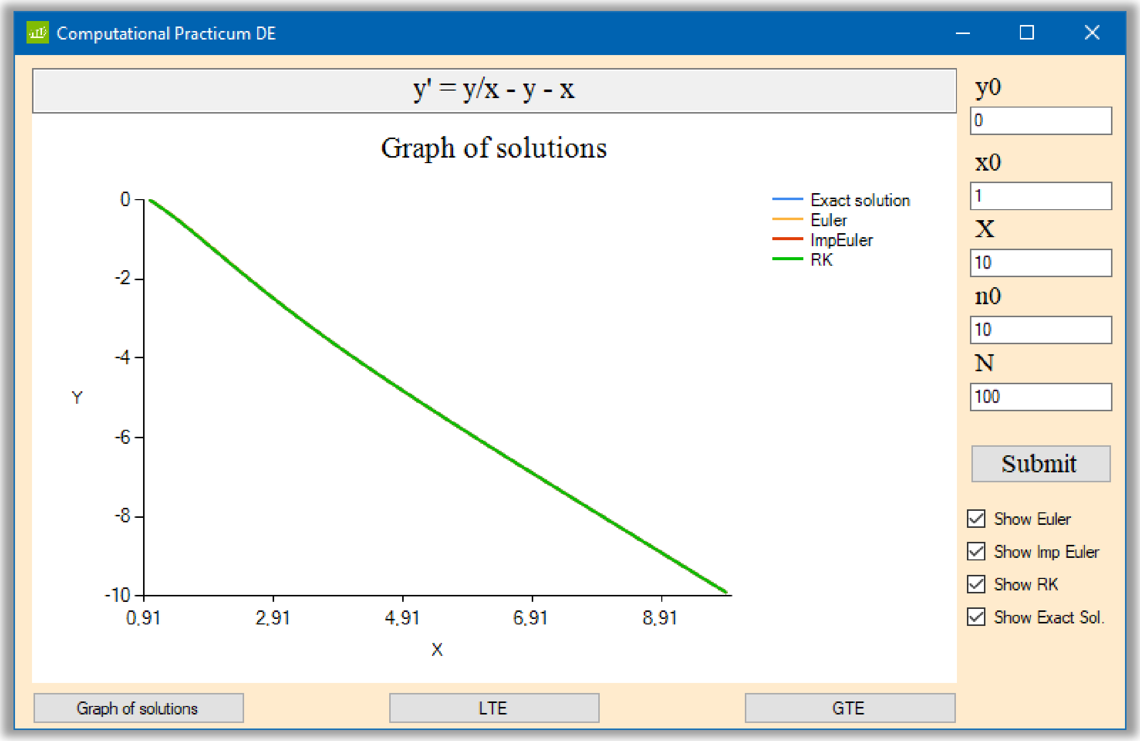
X = 0 is the point of the removable (first type) discontinuity

UML, GUI and the results of the computing:

This GUI was built by the Windows Forms (.NET Framework) on C# v7.3

The approximation of the solution of a given initial value problem: (x0 = 1, y0 = 0, X = 10, n_0 = 20, N = 100):

Graph of solutions:



Computational Practicum DE

y' = y/x - y - x

y0: 0

x0: 1

X: 10

n0: 10

N: 100

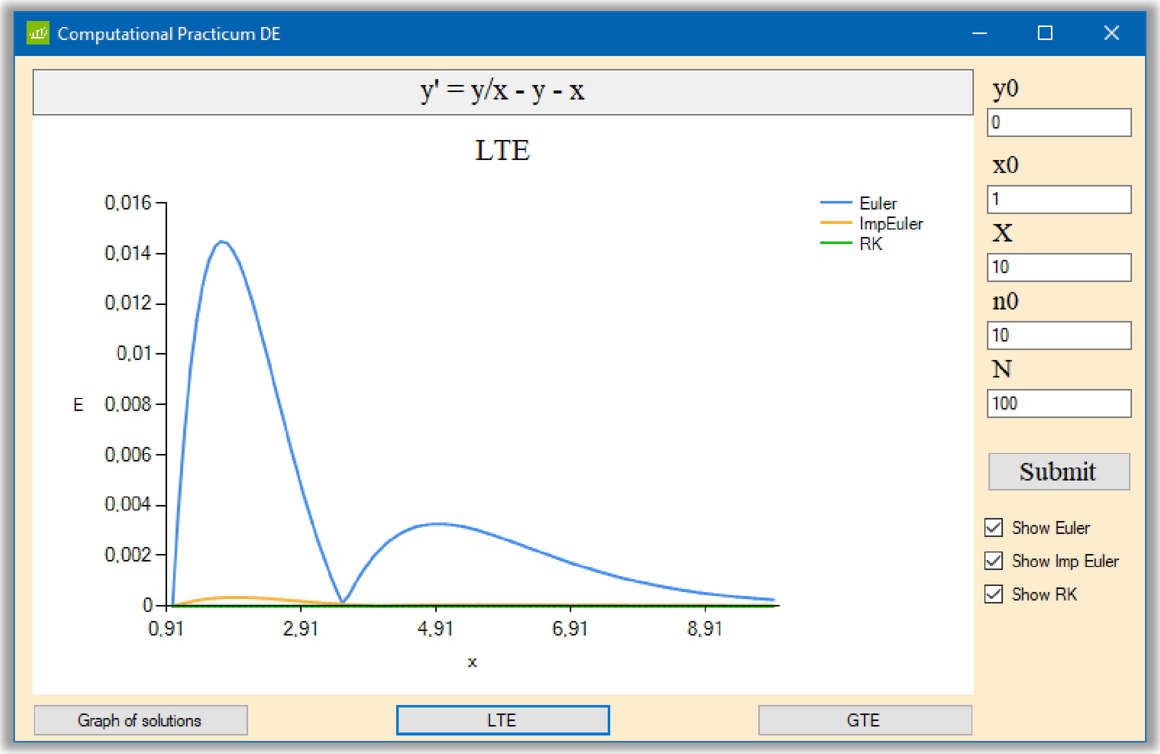
☑ Show Euler

☑ Show Imp Euler

☑ Show RK

☑ Show Exact Sol.

Graph of LTE (Local errors):



Computational Practicum DE

y' = y/x - y - x

y0: 0

x0: 1

X: 10

n0: 10

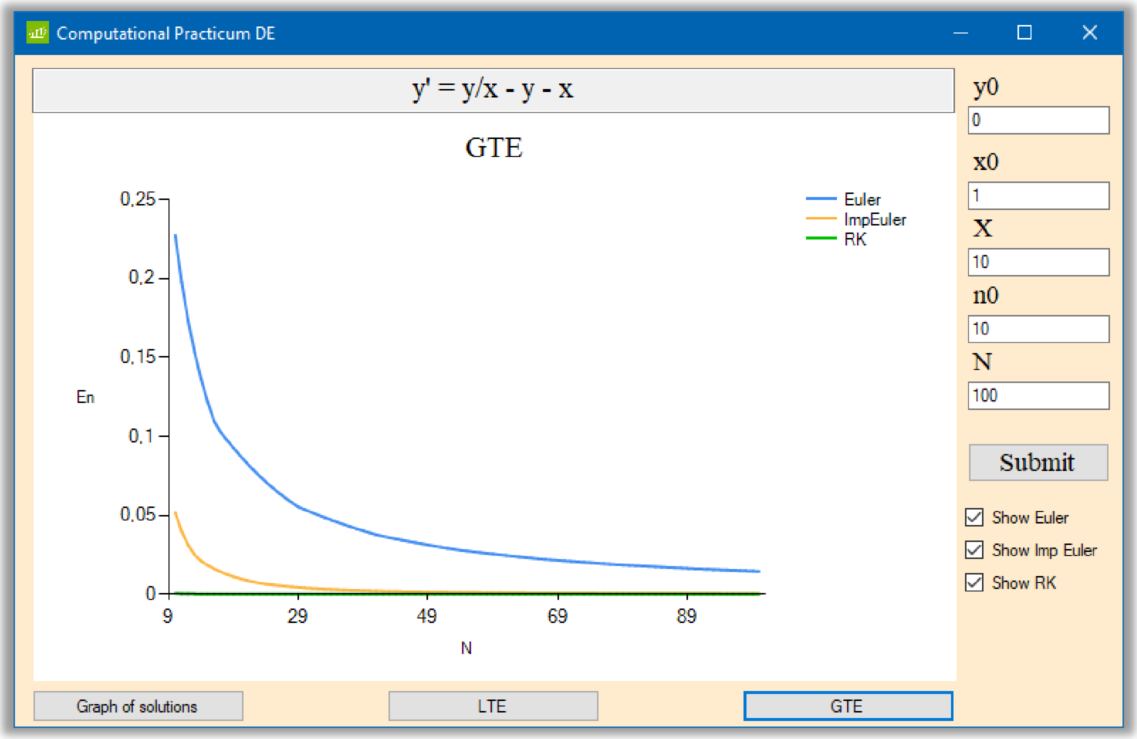
N: 100

☑ Show Euler

☑ Show Imp Euler

☑ Show RK

Graph of GTE (Global errors):



y0: 0

x0: 1

X: 10

n0: 10

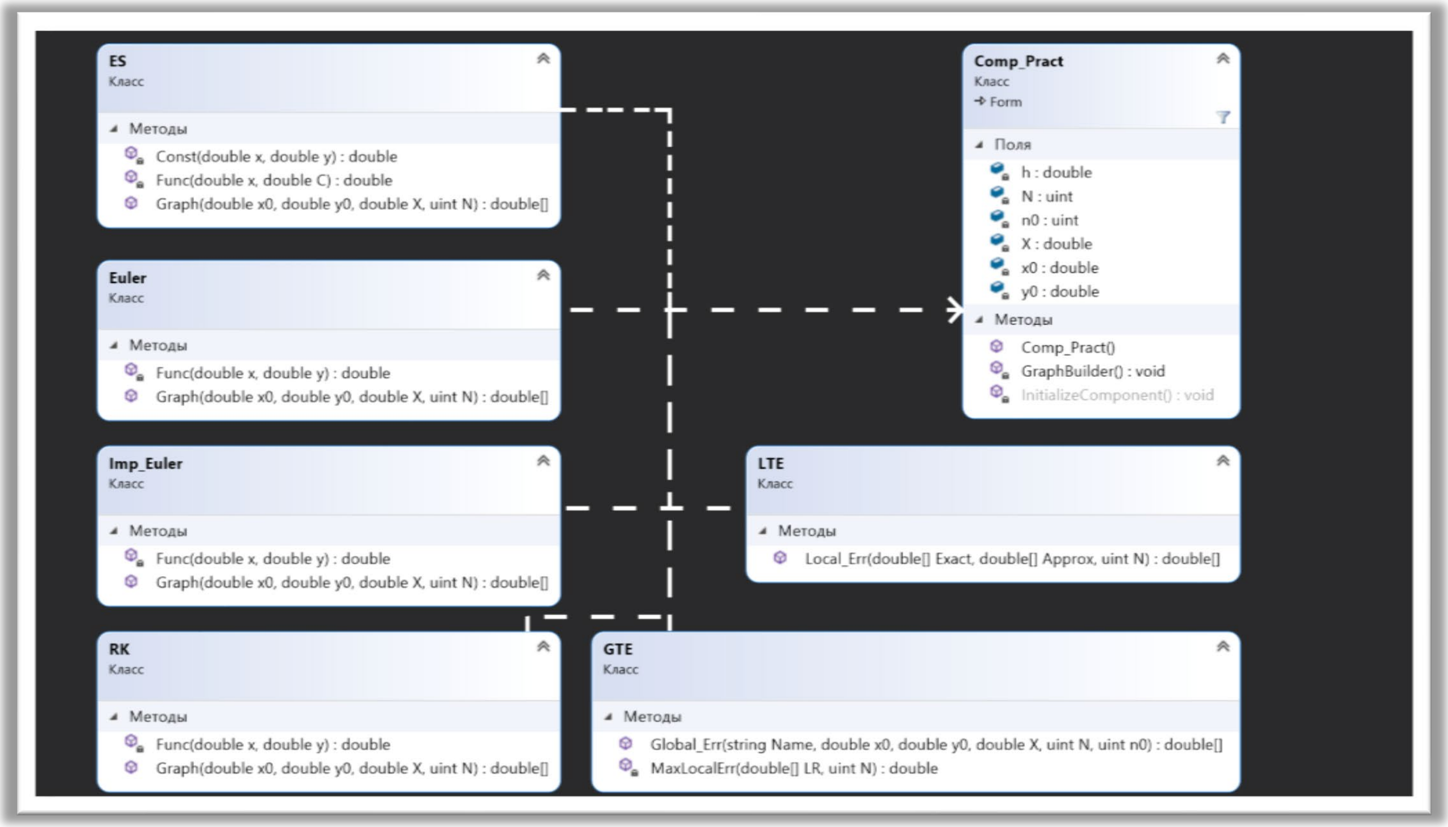
N: 100

☑ Show Euler

☑ Show Imp Euler

☑ Show RK

The UML diagram of the program:

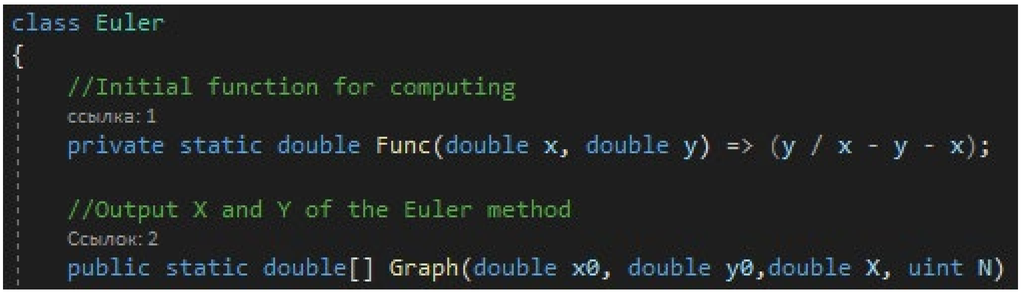


```
ES
Класс
Методы
  Const(double x, double y) : double
  Func(double x, double C) : double
  Graph(double x0, double y0, double X, uint N) : double[]

Euler
Класс
Методы
  Func(double x, double y) : double
  Graph(double x0, double y0, double X, uint N) : double[]

Imp_Euler
Класс
Методы
  Func(double x, double y) : double
  Graph(double x0, double y0, double X, uint N) : double[]

RK
Класс
Методы
  Func(double x, double y) : double
  Graph(double x0, double y0, double X, uint N) : double[]

Comp_Pract
Класс
→ Form
Поля
  h : double
  N : uint
  n0 : uint
  X : double
  x0 : double
  y0 : double
Методы
  Comp_Pract()
  GraphBuilder() : void
  InitializeComponent() : void

LTE
Класс
Методы
  Local_Err(double[] Exact, double[] Approx, uint N) : double[]

GTE
Класс
Методы
  Global_Err(string Name, double x0, double y0, double X, uint N, uint n0) : double[]
  MaxLocalErr(double[] LR, uint N) : double
```

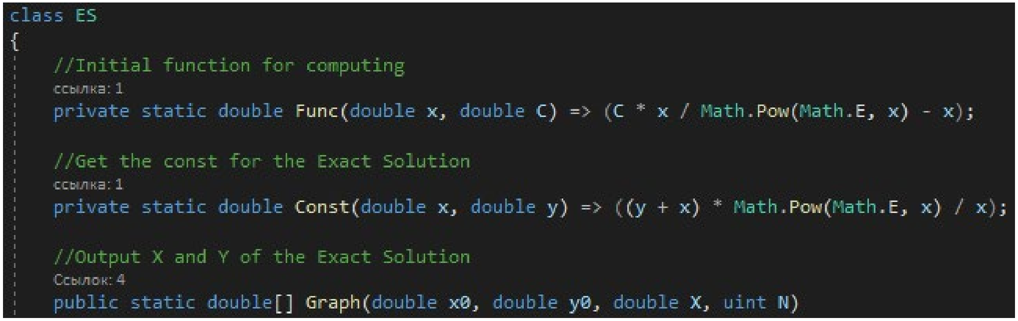
P.S. In the original version of the structure, there should have been an interface "Abs_method", however, since "the interface implementation component is not available by default in C# 7.3", any abstraction of method classes was a rewrite of the code, so the abstract class was not implemented in this application.

Interesting code parts

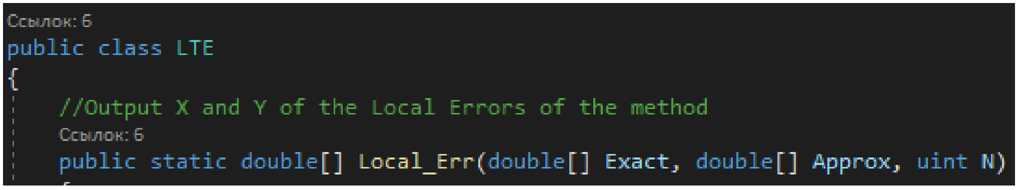
1) Each class of numerical methods is **static**, so the program does not use a lot of memory on displaying graphs, because the data destroys after calculations. This is done to optimize large-volume calculations.



```
class Euler
{
    //Initial function for computing
    ссылка: 1
    private static double Func(double x, double y) => (y / x - y - x);

    //Output X and Y of the Euler method
    Ссылок: 2
    public static double[] Graph(double x0, double y0,double X, uint N)
```



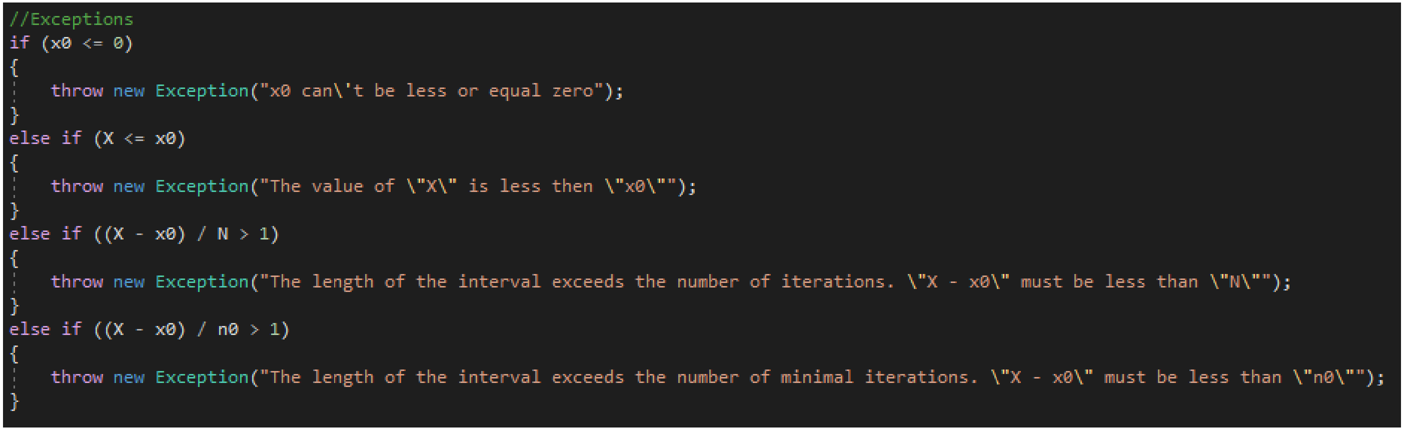
```
class ES
{
    //Initial function for computing
    ссылка: 1
    private static double Func(double x, double C) => (C * x / Math.Pow(Math.E, x) - x);

    //Get the const for the Exact Solution
    ссылка: 1
    private static double Const(double x, double y) => ((y + x) * Math.Pow(Math.E, x) / x);

    //Output X and Y of the Exact Solution
    Ссылок: 4
    public static double[] Graph(double x0, double y0, double X, uint N)
```



```
Ссылок: 6
public class LTE
{
    //Output X and Y of the Local Errors of the method
    Ссылок: 6
    public static double[] Local_Err(double[] Exact, double[] Approx, uint N)
```

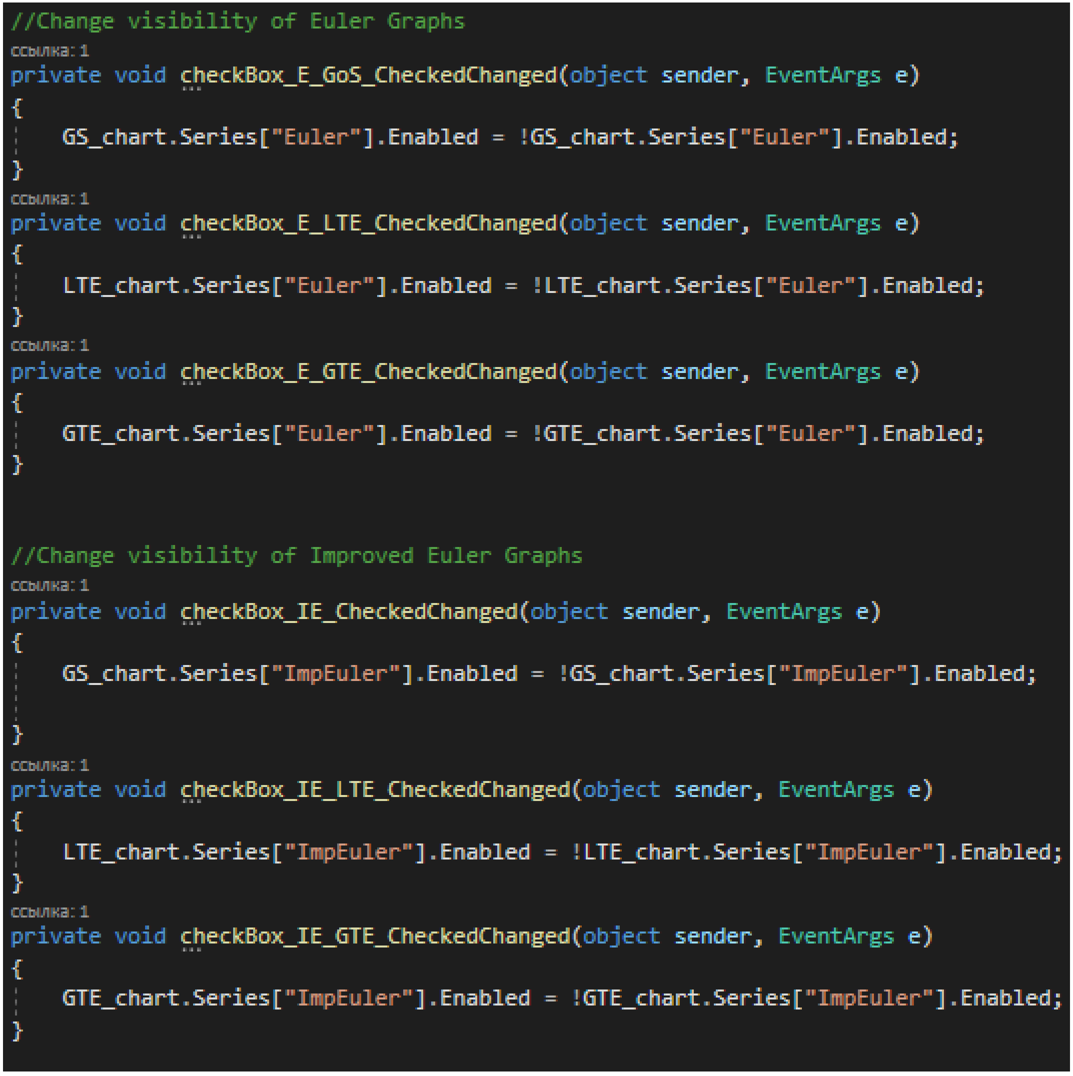
and etc.

2) When you click on the "send data" button, program process **exceptions** so the methods work correctly



```
//Exceptions
if (x0 <= 0)
{
    throw new Exception("x0 can\'t be less or equal zero");
}
else if (X <= x0)
{
    throw new Exception("The value of \"X\" is less then \"x0\"");
}
else if ((X - x0) / N > 1)
{
    throw new Exception("The length of the interval exceeds the number of iterations. \"X - x0\" must be less than \"N\"");
}
else if ((X - x0) / n0 > 1)
{
    throw new Exception("The length of the interval exceeds the number of minimal iterations. \"X - x0\" must be less than \"n0\"");
}
```

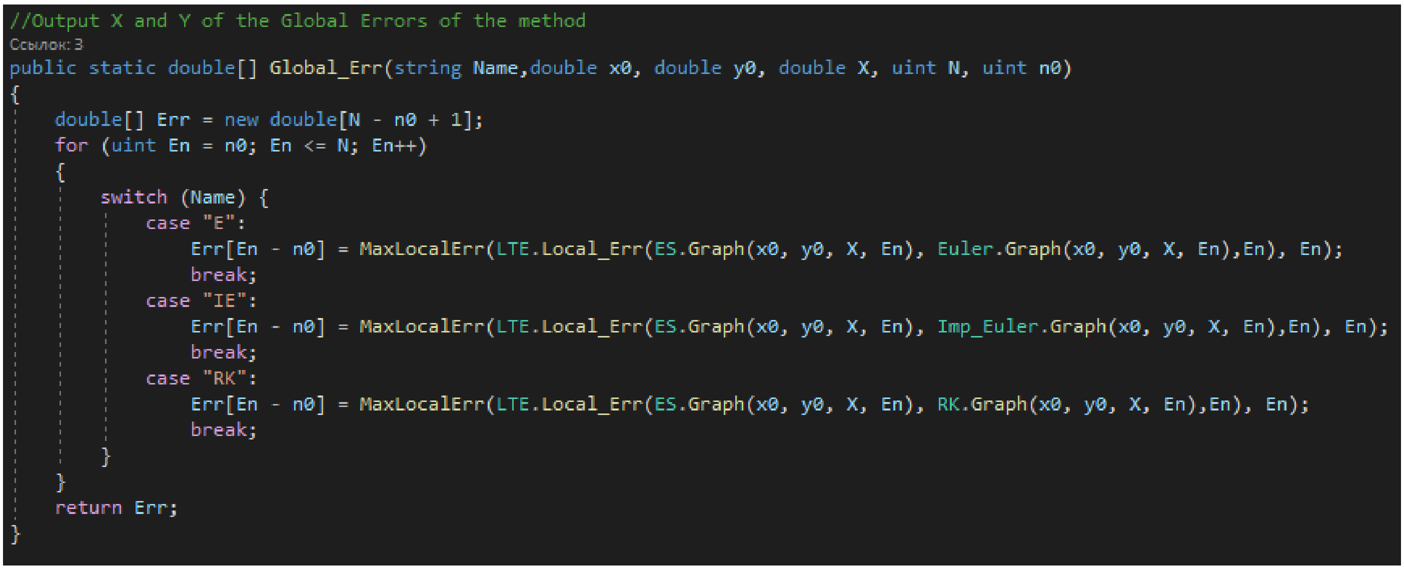
3) For each graph, there is an option to **enable/disable the visibility** of certain methods to make it easier to compare certain data.



```
//Change visibility of Euler Graphs
ссылка: 1
private void checkBox_E_GoS_CheckedChanged(object sender, EventArgs e)
{
    GS_chart.Series["Euler"].Enabled = !GS_chart.Series["Euler"].Enabled;
}
ссылка: 1
private void checkBox_E_LTE_CheckedChanged(object sender, EventArgs e)
{
    LTE_chart.Series["Euler"].Enabled = !LTE_chart.Series["Euler"].Enabled;
}
ссылка: 1
private void checkBox_E_GTE_CheckedChanged(object sender, EventArgs e)
{
    GTE_chart.Series["Euler"].Enabled = !GTE_chart.Series["Euler"].Enabled;
}


//Change visibility of Improved Euler Graphs
ссылка: 1
private void checkBox_IE_CheckedChanged(object sender, EventArgs e)
{
    GS_chart.Series["ImpEuler"].Enabled = !GS_chart.Series["ImpEuler"].Enabled;

}
ссылка: 1
private void checkBox_IE_LTE_CheckedChanged(object sender, EventArgs e)
{
    LTE_chart.Series["ImpEuler"].Enabled = !LTE_chart.Series["ImpEuler"].Enabled;
}
ссылка: 1
private void checkBox_IE_GTE_CheckedChanged(object sender, EventArgs e)
{
    GTE_chart.Series["ImpEuler"].Enabled = !GTE_chart.Series["ImpEuler"].Enabled;
}
```

and etc.

4) For GTE was used **switch-case** to find particular GTE graph.



```
//Output X and Y of the Global Errors of the method
Ссылок: 3
public static double[] Global_Err(string Name,double x0, double y0, double X, uint N, uint n0)
{
    double[] Err = new double[N - n0 + 1];
    for (uint En = n0; En <= N; En++)
    {
        switch (Name) {
            case "E":
                Err[En - n0] = MaxLocalErr(LTE.Local_Err(ES.Graph(x0, y0, X, En), Euler.Graph(x0, y0, X, En),En), En);
                break;
            case "IE":
                Err[En - n0] = MaxLocalErr(LTE.Local_Err(ES.Graph(x0, y0, X, En), Imp_Euler.Graph(x0, y0, X, En),En), En);
                break;
            case "RK":
                Err[En - n0] = MaxLocalErr(LTE.Local_Err(ES.Graph(x0, y0, X, En), RK.Graph(x0, y0, X, En),En), En);
                break;
        }
    }
    return Err;
}
```