PUNTO FIJO

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using BaseMetodos;

namespace tr

{

public partial class PuntoFijo : Form

{

public PuntoFijo()

{

InitializeComponent();

}

double M, t, V, g, ES, C1=0, C, fu, C1\_Ant= 0, EA=0;

private void txtM\_KeyPress(object sender, KeyPressEventArgs e)

{

if (Char.IsLetter(e.KeyChar) || Char.IsSymbol(e.KeyChar) || Char.IsWhiteSpace(e.KeyChar))

{

e.Handled = true;

return;

}

}

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

{

Menu frmMenu = new Menu();

frmMenu.Show();

this.Hide();

}

private void PuntoFijo\_FormClosing(object sender, FormClosingEventArgs e)

{

Application.Exit();

}

private void PuntoFijo\_Load(object sender, EventArgs e)

{

}

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

{

int i = 0;

if (txtM.Text == " " || txt\_t.Text == " " || txtV.Text == "" || txtES.Text == "" || txtC1.Text == "")

{

MessageBox.Show("Llene todos los campos", "Advertencia", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

else

{

try

{

M = Convert.ToDouble(txtM.Text);

t = Convert.ToDouble(txt\_t.Text);

V = Convert.ToDouble(txtV.Text);

g = Convert.ToDouble(9.8);

ES = Convert.ToDouble(txtES.Text);

C1 = Convert.ToDouble(txtC1.Text);

}

catch (Exception exc)

{

MessageBox.Show("Ingrese los valores correctamente");

txtM.Clear();

txt\_t.Clear();

txtV.Clear();

txtES.Clear();

txtC1.Clear();

return;

}

if (ES <= 0)

{

MessageBox.Show("ES no puede ser igual o menor a cero");

return;

}

do

{

C1\_Ant = C1;

i = i + 1;

C1 = f1(C1, t, g, M, V);

if (i == 1)

{

EA = 100;

}

EA = Math.Abs((C1 - C1\_Ant) / C1) \* 100;

txtC.Text = C1.ToString();

txti.Text = i.ToString();

txtEA.Text = EA.ToString();

C1\_Ant = 0;

} while (EA > ES);

}

}

public static double f1(double C1, double t, double g, double M, double V)

{

double des = (t / M);

return (g) \* (M) / (C1) \* (1 - Math.Exp(-des \* (C1))) + C1 - V;

}

}

}

using System;

SECANTE

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using BaseMetodos;

namespace tr

{

public partial class Secante : Form

{

Base\_Datos objDataAccess = new Base\_Datos();

public Secante()

{

InitializeComponent();

}

double M, t, V, g, ES, C1 = 0, C, fu, C1\_Ant = 0, EA = 0,FC1, C3\_Ant,C3,metodo2, Xi\_1,Xi,a,b,Fun1,Fun2, FXi,FXi\_1,C2,metodo,XiMas1,FXimas1;

private void txtM\_KeyPress(object sender, KeyPressEventArgs e)

{

if (Char.IsLetter(e.KeyChar) || Char.IsSymbol(e.KeyChar) || Char.IsWhiteSpace(e.KeyChar))

{

e.Handled = true;

return;

}

}

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

{

Menu frmMenu = new Menu();

frmMenu.Show();

this.Hide();

}

private void Secante\_FormClosing(object sender, FormClosingEventArgs e)

{

Application.Exit();

}

private void Secante\_Load(object sender, EventArgs e)

{

if (!objDataAccess.CreateConnection())

MessageBox.Show("Error al intentar conectar con la base de datos" + objDataAccess.Error);

}

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

{

int i = 0;

if (txtM.Text == " " || txt\_t.Text == " " || txtV.Text == "" || txtES.Text == "" || txtC1.Text == "" || txtC2.Text == "")

{

MessageBox.Show("Llene todos los campos", "Advertencia", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

else{

try

{

M = Convert.ToDouble(txtM.Text);

t = Convert.ToDouble(txt\_t.Text);

V = Convert.ToDouble(txtV.Text);

g = Convert.ToDouble(9.8);

ES = Convert.ToDouble(txtES.Text);

Xi = Convert.ToDouble(txtC1.Text);

Xi\_1 = Convert.ToDouble(txtC2.Text);

XiMas1 = 0;

}

catch (Exception exc)

{

MessageBox.Show("Ingrese los valores correctamente");

txtM.Clear();

txt\_t.Clear();

txtV.Clear();

txtES.Clear();

txtC1.Clear();

txtC2.Clear();

return;

}

if (ES <= 0)

{

MessageBox.Show("ES no puede ser igual o menor a cero");

return;

}

if (Xi == Xi\_1)

{

MessageBox.Show("Los valores son iguales, ingrese otros");

return;

}

do

{

i++;

FXi = f1(Xi, g, M, t, V);

FXi\_1 = f1(Xi\_1, g, M, t, V);

XiMas1 = Fmetodo(Xi, Xi\_1, FXi, FXi\_1);

EA = Math.Abs((XiMas1 - Xi) / XiMas1) \* 100;

if (i == 1)

{

EA = 100;

}

Xi\_1 = Xi;

Xi = XiMas1;

txtC.Text = XiMas1.ToString();

txti.Text = i.ToString();

txtEA.Text = EA.ToString();

} while (EA > ES);

}

C = XiMas1;

C1 = f1(C, g, M, t, V);

//txt.Text = C1.ToString();

}

public static double Fmetodo(double Xi, double Xi\_1, double FXi, double FXi\_1)

{

return Xi - (FXi \* (Xi\_1 - Xi)/(FXi\_1-FXi));

}

public static double f1(double Xi, double g, double M, double t, double V)

{

double des = (t / M);

return (g) \* (M) / (Xi) \* (1 - Math.Exp(-des \* (Xi))) - V;

}

}

}

BISECCION

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using BaseMetodos;

namespace tr

{

public partial class Biseccion : Form

{

Base\_Datos objDataAccess = new Base\_Datos();

public Biseccion()

{

InitializeComponent();

}

double M, t, V, g, ES, C1, C2, C, C3 = 0, C3\_Ant=0, EA=0;

private void Biseccion\_FormClosing(object sender, FormClosingEventArgs e)

{

Application.Exit();

}

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

{

Menu frmMenu = new Menu();

frmMenu.Show();

this.Hide();

}

private void txtM\_KeyPress(object sender, KeyPressEventArgs e)

{

if (Char.IsLetter(e.KeyChar) || Char.IsSymbol(e.KeyChar) || Char.IsWhiteSpace(e.KeyChar))

{

e.Handled = true;

return;

}

}

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

{

string query = string.Format("TRUNCATE table biseccion");

if (objDataAccess.Execute(query) == -1)

MessageBox.Show("Ocurrio un error: " + objDataAccess.Error);

else

FillGrid();

}

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

{

dataGridView1.DataSource = objDataAccess.FillTable("SELECT \* FROM biseccion");

}

private void groupBox1\_Enter(object sender, EventArgs e)

{

}

int i = 0;

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

{

if (txtM.Text == " " || txt\_t.Text == " " || txtV.Text == "" || txtES.Text == "" || txtC1.Text == ""|| txtC2.Text=="")

{

MessageBox.Show("Llene todos los campos", "Advertencia", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

else

{

try

{

M = Convert.ToDouble(txtM.Text);

t = Convert.ToDouble(txt\_t.Text);

V = Convert.ToDouble(txtV.Text);

g = Convert.ToDouble(9.8);

ES = Convert.ToDouble(txtES.Text);

C1 = Convert.ToDouble(txtC1.Text);

C2 = Convert.ToDouble(txtC2.Text);

}

catch (Exception exc)

{

MessageBox.Show("Ingrese los valores correctamente");

txtM.Clear();

txt\_t.Clear();

txtV.Clear();

txtES.Clear();

txtC1.Clear();

txtC2.Clear();

return;

}

if (ES <= 0)

{

MessageBox.Show("ES no puede ser igual o menor a cero");

return;

}

if (f2(C2, g, M, t, V) \* f1(C1, g, M, t, V) > 0)

{

MessageBox.Show("Los valores ingresados dan positivo, ingresar otros numeros");

return;

}

else

{

i = 0;

do

{

C3\_Ant = C3;

i = i + 1;

C3 = (C1 + C2) / 2;

EA = Math.Abs((C3 - C3\_Ant) / C3) \* 100;

if (i == 1)

{

EA = 100;

}

if (f3(C3, g, M, t, V) \* f1(C1, g, M, t, V) < 0)

{

C2 = C3;

}

else if (f3(C3, g, M, t, V) \* f2(C2, g, M, t, V) < 0)

{

C1 = C3;

}

txti.Text = i.ToString();

txtEA.Text = EA.ToString();

txtC.Text = C3.ToString();

string query = String.Format("INSERT INTO biseccion VALUES( '{0}', '{1}','{2}' )", txti.Text, txtC.Text, txtEA.Text);

if (objDataAccess.Execute(query) == -1)

MessageBox.Show("Ocurrio un error: " + objDataAccess.Error);

} while (EA > ES);

}

}

}

private void Form1\_Load(object sender, EventArgs e)

{

if (!objDataAccess.CreateConnection())

MessageBox.Show("Error al intentar conectar con la base de datos" + objDataAccess.Error);

FillGrid();

}

public static double f1(double C1, double g, double M, double t, double V)

{

double des = (t / M);

return (g) \* (M) / (C1) \* (1 - Math.Exp(-des \* (C1))) - V;

}

public static double f2(double C2, double g, double M, double t, double V)

{

double des2 = (t / M);

return (g) \* (M) / (C2) \* (1 - Math.Exp(-des2 \* (C2))) - V;

}

public static double f3(double C3, double g, double M, double t, double V)

{

double des3 = (t / M);

return (g) \* (M) / (C3) \* (1 - Math.Exp(-des3 \* (C3))) - V;

}

private void FillGrid()

{

dataGridView1.DataSource = objDataAccess.FillTable("SELECT \* FROM biseccion");

}

}

}

Menu

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace tr

{

public partial class Menu : Form

{

public Menu()

{

InitializeComponent();

}

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

{

Biseccion frmBise = new Biseccion();

frmBise.Show();

this.Hide();

}

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

{

PuntoFijo frmPunto = new PuntoFijo();

frmPunto.Show();

this.Hide();

}

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

{

Secante frmSec = new Secante();

frmSec.Show();

this.Hide();

}

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

{

Application.Exit();

}

private void Menu\_FormClosing(object sender, FormClosingEventArgs e)

{

Application.Exit();

}

private void Menu\_Load(object sender, EventArgs e)

{

}

}

}