using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using Finisar.SQLite;

using ZedGraph;

using System.Drawing.Drawing2D;

namespace BEPoint

{

public partial class Form1 : Form

{

public double fixcost, vcost, qsales, part\_vcost, vipusk1, qsales\_money, sum\_cost1, sum\_cost2, BreakEvenPoint\_money, BreakEvenPoint\_count, shag, price;

public string s\_fixcost, s\_vcost, s\_qsales, s\_qsales\_money, s\_price;

public int id\_now;

SQLiteConnection sqlite\_conn;

SQLiteCommand sqlite\_cmd;

SQLiteDataReader sqlite\_datareader;

public Form1()

{

InitializeComponent();

zedGraphControl1.ContextMenuBuilder +=

new ZedGraphControl.ContextMenuBuilderEventHandler(zedGraph\_ContextMenuBuilder);

PreDrawGraph();

}

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

{

sqlite\_conn = new SQLiteConnection("Data Source=dbBEP.db;Version=3;New=True;Compress=True;");

sqlite\_conn.Open();

sqlite\_cmd = sqlite\_conn.CreateCommand();

sqlite\_cmd.CommandText = "CREATE TABLE data\_BEP(" +

"id INT(11) NOT NULL," +

"name CHAR(20) NOT NULL," +

"fixcost FLOAT(7, 3)," +

"vcost FLOAT(7, 3)," +

"qsales FLOAT(7,3)," +

"qsales\_money FLOAT(7, 3)," +

"price FLOAT(1, 2));";

sqlite\_cmd.ExecuteNonQuery();

sqlite\_cmd.CommandText = "INSERT INTO data\_BEP(id, name, fixcost, vcost, qsales, qsales\_money, price) VALUES(0, 'Анальгин, N10', 12503.667, 42818.559, 129721.396, 47996.916, 0.37);";

sqlite\_cmd.ExecuteNonQuery();

sqlite\_cmd.CommandText = "INSERT INTO data\_BEP(id, name, fixcost, vcost, qsales, qsales\_money, price) VALUES(1, 'Аспаркам, N10', 30639.643, 85033.507, 534609.848, 117614.166, 0.22);";

sqlite\_cmd.ExecuteNonQuery();

sqlite\_cmd.CommandText = "INSERT INTO data\_BEP(id, name, fixcost, vcost, qsales, qsales\_money, price) VALUES(2, 'Аспаркам, N50', 8179.855, 19945.036, 22428.154, 31399.416, 1.4);";

sqlite\_cmd.ExecuteNonQuery();

sqlite\_cmd.CommandText = "INSERT INTO data\_BEP(id, name, fixcost, vcost, qsales, qsales\_money, price) VALUES(3, 'Диклофенак, №30', 6003.643, 6641.818, 8863.75, 23045.75, 2.6);";

sqlite\_cmd.ExecuteNonQuery();

sqlite\_cmd.CommandText = "INSERT INTO data\_BEP(id, name, fixcost, vcost, qsales, qsales\_money, price) VALUES(4, 'Баралгин, N10', 10690.215, 20010.056, 51294.687, 41035.75, 0.8);";

sqlite\_cmd.ExecuteNonQuery();

sqlite\_cmd.CommandText = "INSERT INTO data\_BEP(id, name, fixcost, vcost, qsales, qsales\_money, price) VALUES(5, 'Фталазол, N10', 1165.434, 3368.418, 8133.939, 4473.667, 0.55);";

sqlite\_cmd.ExecuteNonQuery();

sqlite\_cmd.CommandText = "INSERT INTO data\_BEP(id, name, fixcost, vcost, qsales, qsales\_money, price) VALUES(6, 'Цитрамон, №6', 14044.604, 31151.093, 245054.545, 53912, 0.22);";

sqlite\_cmd.ExecuteNonQuery();

sqlite\_conn.Close();

}

void zedGraph\_ContextMenuBuilder(ZedGraphControl sender,

ContextMenuStrip menuStrip,

Point mousePt,

ZedGraphControl.ContextMenuObjectState objState)

{

// !!!

// Переименуем (переведем на русский язык) некоторые пункты контекстного меню

menuStrip.Items[0].Text = "Копировать";

menuStrip.Items[1].Text = "Сохранить как картинку…";

menuStrip.Items[2].Text = "Параметры страницы…";

menuStrip.Items[3].Text = "Печать…";

menuStrip.Items[4].Text = "Показывать значения в точках…";

menuStrip.Items[7].Text = "Установить масштаб по умолчанию…";

// Некоторые пункты удалим

menuStrip.Items.RemoveAt(5);

menuStrip.Items.RemoveAt(5);

}

public void PreDrawGraph()

{

// Получим панель для рисования

GraphPane pane = zedGraphControl1.GraphPane;

// !!!

// Изменим тест надписи по оси X

pane.XAxis.Title.Text = "Выпуск";

// Изменим текст по оси Y

pane.YAxis.Title.Text = "Затраты";

// Изменим текст заголовка графика

pane.Title.Text = "Точка безубыточности";

// Сделаем шрифт не полужирным

pane.Title.FontSpec.IsBold = true;

// Очистим список кривых на тот случай, если до этого сигналы уже были нарисованы

pane.CurveList.Clear();

// Вызываем метод AxisChange (), чтобы обновить данные об осях.

zedGraphControl1.AxisChange();

// Обновляем график

zedGraphControl1.Invalidate();

}

private void button1\_Click(object sender, EventArgs e)

{

fixcost = Convert.ToDouble(textBox1.Text); //Постоянные затраты,b

vcost = Convert.ToDouble(textBox2.Text); // Переменные

qsales = Convert.ToDouble(textBox3.Text); //Выпуск (Объем реализации) в ед

qsales\_money = Convert.ToDouble(textBox4.Text); // Выпуск в грн

price = Convert.ToDouble(s\_price); //цена в грн за еденицу

part\_vcost = vcost / qsales\_money; //Доля переменных затрат в выпуске,a

vipusk1 = 0;

sum\_cost1 = fixcost + vipusk1;

sum\_cost2 = fixcost + vcost;

BreakEvenPoint\_money = fixcost / (1-part\_vcost); //точка безубыточности(грн)

BreakEvenPoint\_count = BreakEvenPoint\_money / price; //точка безубыточности(ед)

//-------------------------------------------------------------------

// Получим панель для рисования

GraphPane pane = zedGraphControl1.GraphPane;

double xmin = 0;

double xmax = qsales;

// Устанавливаем интересующий нас интервал по оси X

pane.XAxis.Scale.Min = xmin;

pane.XAxis.Scale.Max = xmax + 1000;

// !!!

// Устанавливаем интересующий нас интервал по оси Y

pane.YAxis.Scale.Min = 0;

pane.YAxis.Scale.Max = xmax + 1000;

// Очистим список кривых на тот случай, если до этого сигналы уже были нарисованы

pane.CurveList.Clear ();

// Создадим список точек для кривой f1(x)

PointPairList f1\_list = new PointPairList ();

// Создадим список точек для кривой f2(x)

PointPairList f2\_list = new PointPairList ();

PointPairList f3\_list = new PointPairList();

PointPairList f4\_list = new PointPairList();

PointPairList f5\_bep = new PointPairList();

PointPairList f6\_bep = new PointPairList();

for (double x = xmin; x <= xmax; x++)

{

f1\_list.Add(x, fixcost);

}

f2\_list.Add(0, 0);

f2\_list.Add(qsales, vcost);

f3\_list.Add(vipusk1, vipusk1);

f3\_list.Add(qsales, qsales\_money);

f3\_list.Add(BreakEvenPoint\_count, BreakEvenPoint\_money);

f4\_list.Add(vipusk1, sum\_cost1);

f4\_list.Add(qsales, sum\_cost2);

f5\_bep.Add(BreakEvenPoint\_count, 0);

f5\_bep.Add(BreakEvenPoint\_count, BreakEvenPoint\_money);

f6\_bep.Add(BreakEvenPoint\_count, BreakEvenPoint\_money);

f6\_bep.Add(xmax + 1000, BreakEvenPoint\_money);

LineItem f1\_curve = pane.AddCurve("Постоянные затраты", f1\_list, Color.Magenta, SymbolType.None);

LineItem f2\_curve = pane.AddCurve("Переменные затраты", f2\_list, Color.Tomato, SymbolType.None);

LineItem f3\_curve = pane.AddCurve("Объем продаж в грн", f3\_list, Color.Blue, SymbolType.None);

LineItem f4\_curve = pane.AddCurve("Суммарные затраты", f4\_list, Color.Cyan, SymbolType.None);

LineItem f5\_curve = pane.AddCurve("Точка безубыточности", f5\_bep, Color.Black, SymbolType.None);

LineItem f6\_curve = pane.AddCurve("Точка безубыточности", f6\_bep, Color.Black, SymbolType.None);

f5\_curve.Line.Style = DashStyle.DashDot;

f6\_curve.Line.Style = DashStyle.DashDot;

f5\_curve.Line.IsSmooth = true;

f6\_curve.Line.IsSmooth = true;

// Cделаем чуть помальче шрифт, чтобы уместилось побольше меток

pane.XAxis.Scale.FontSpec.Size = 10;

pane.YAxis.Scale.FontSpec.Size = 10;

// !!! Просто уберем отображение степени в подписи оси X

pane.XAxis.Title.IsOmitMag = true;

pane.YAxis.Title.IsOmitMag = true;

// !!! Сами установим коэффициент, на который умножается значение по оси X

// !!! В данном случае значение будет умножаться на 10^-9

pane.XAxis.Scale.Mag = 0;

// Параметры оси Y

// !!! Установим коэффициент, на который умножается значение по оси Y

// !!! В данном случае значение будет умножаться на 10^0 = 1, то есть умножения не будет

pane.YAxis.Scale.Mag = 0;

// Крупные риски по оси X будут идти с периодом 5

pane.XAxis.Scale.MajorStep = 2000.0;

// Мелкие риски будут идти с периодом 1

// Таким образом, между крупными рисками будет 3 делений или 2 риски

pane.XAxis.Scale.MinorStep = 1000.0;

// Крупные риски по оси Y будут идти с периодом 0.1

pane.YAxis.Scale.MajorStep = 2000.0;

// А мелкие риски - с периодом 0.05

// Между крупными рисками по оси Y будет два отсчета или одна риска

pane.YAxis.Scale.MinorStep = 1000.0;

/\*

// Включаем отображение сетки напротив крупных рисок по оси X

pane.XAxis.MajorGrid.IsVisible = true;

// Задаем вид пунктирной линии для крупных рисок по оси X:

// Длина штрихов равна 10 пикселям, ...

pane.XAxis.MajorGrid.DashOn = 10;

// затем 5 пикселей - пропуск

pane.XAxis.MajorGrid.DashOff = 5;

// Включаем отображение сетки напротив крупных рисок по оси Y

pane.YAxis.MajorGrid.IsVisible = true;

// Аналогично задаем вид пунктирной линии для крупных рисок по оси Y

pane.YAxis.MajorGrid.DashOn = 10;

pane.YAxis.MajorGrid.DashOff = 5;

// Включаем отображение сетки напротив мелких рисок по оси X

pane.YAxis.MinorGrid.IsVisible = true;

// Задаем вид пунктирной линии для крупных рисок по оси Y:

// Длина штрихов равна одному пикселю, ...

pane.YAxis.MinorGrid.DashOn = 1;

// затем 2 пикселя - пропуск

pane.YAxis.MinorGrid.DashOff = 2;

// Включаем отображение сетки напротив мелких рисок по оси Y

pane.XAxis.MinorGrid.IsVisible = true;

// Аналогично задаем вид пунктирной линии для крупных рисок по оси Y

pane.XAxis.MinorGrid.DashOn = 1;

pane.XAxis.MinorGrid.DashOff = 2;

\*/

zedGraphControl1.AxisChange();

zedGraphControl1.Invalidate();

}

public double Qsales(double x)

{

return (part\_vcost \* x) / 2;

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

id\_now = comboBox1.SelectedIndex;

sqlite\_conn = new SQLiteConnection("Data Source=dbBEP.db;Version=3;New=False;Compress=False;");

sqlite\_conn.Open();

sqlite\_cmd = sqlite\_conn.CreateCommand();

//switch

switch (id\_now)

{

case 0:

sqlite\_cmd.CommandText = "SELECT fixcost, vcost, qsales, qsales\_money, price FROM data\_BEP WHERE id=0";

break;

case 1:

sqlite\_cmd.CommandText = "SELECT fixcost, vcost, qsales, qsales\_money, price FROM data\_BEP WHERE id=1";

break;

case 2:

sqlite\_cmd.CommandText = "SELECT fixcost, vcost, qsales, qsales\_money, price FROM data\_BEP WHERE id=2";

break;

case 3:

sqlite\_cmd.CommandText = "SELECT fixcost, vcost, qsales, qsales\_money, price FROM data\_BEP WHERE id=3";

break;

case 4:

sqlite\_cmd.CommandText = "SELECT fixcost, vcost, qsales, qsales\_money, price FROM data\_BEP WHERE id=4";

break;

case 5:

sqlite\_cmd.CommandText = "SELECT fixcost, vcost, qsales, qsales\_money, price FROM data\_BEP WHERE id=5";

break;

case 6:

sqlite\_cmd.CommandText = "SELECT fixcost, vcost, qsales, qsales\_money, price FROM data\_BEP WHERE id=6";

break;

}

//switch

sqlite\_datareader = sqlite\_cmd.ExecuteReader();

while (sqlite\_datareader.Read())

{

s\_fixcost = Convert.ToString(sqlite\_datareader["fixcost"]);

s\_vcost = Convert.ToString(sqlite\_datareader["vcost"]);

s\_qsales = Convert.ToString(sqlite\_datareader["qsales"]);

s\_qsales\_money = Convert.ToString(sqlite\_datareader["qsales\_money"]);

s\_price = Convert.ToString(sqlite\_datareader["price"]);

}

sqlite\_conn.Close();

textBox1.Clear();

textBox2.Clear();

textBox3.Clear();

textBox4.Clear();

textBox1.AppendText(s\_fixcost);

textBox2.AppendText(s\_vcost);

textBox3.AppendText(s\_qsales);

textBox4.AppendText(s\_qsales\_money);

}

}

}