using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Drawing;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.IO;

using System.Xml;

namespace MetodyInzWiedzy

{

public partial class Project1 : UserControl

{

public List<List<string>> dane, daneZnormalizowane;

public Project1()

{

InitializeComponent();

}

private void Project1\_Load(object sender, EventArgs e)

{

}

private void button2\_Click(object sender, EventArgs e)

{

SendToBack();

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

public void button3\_Click(object sender, EventArgs e)

{

dane = new List<List<string>>(); // Tworzenie listy dwuwymiarowej

if (openFileDialog1.ShowDialog() == DialogResult.OK) // sprawdzamy czy plik został wybrany

{

dataGridView1.Rows.Clear(); //reset okna

string filename = openFileDialog1.FileName;

var table = File.ReadAllLines(filename); // odczytujemy z pliku linijki

dataGridView1.ColumnCount = table[0].Split(' ', ',', '\t').GetLength(0); // określenie ilości kolumn

for (int i = 0; i < table.Length; i++)

{

dane.Add(new List<string>()); // dodanie elemenru do listy

DataGridViewRow deklaracja = new DataGridViewRow(); //deklaracja rekordu w grid

deklaracja.CreateCells(dataGridView1); // tworzenie komórek

for (int j = 0; j < table[i].Split(' ', ',','\t').GetLength(0); j++)

{

dane[i].Add(table[i].Split(' ', ',', '\t')[j]); // dodawanie danych z plików do listy

deklaracja.Cells[j].Value = dane[i][j]; // dodawanie danych w celu wyswietlenia

}

dataGridView1.Rows.Add(deklaracja); // dodanie Rows

}

}

}

public void btnSaveXML\_Click(object sender, EventArgs e)

{

SaveFileDialog saveFileDialog4 = new SaveFileDialog();

saveFileDialog4.Filter = "xml files (\*.xml)|\*.xml";

saveFileDialog4.FilterIndex = 2;

saveFileDialog4.RestoreDirectory = true;

saveFileDialog4.OverwritePrompt = true;

if (saveFileDialog4.ShowDialog() == DialogResult.OK)

{

WriteXML2(dane, saveFileDialog4.FileName);

}

}

public void WriteXML2(List<List<string>> dane, string path)

{

XmlDocument doc = new XmlDocument();

XmlElement element1 = doc.CreateElement(string.Empty, "table", string.Empty);

doc.AppendChild(element1);

for (int r = 0; r < dane.Count; r++)

{

XmlElement element2 = doc.CreateElement(string.Empty, "rec" + r, string.Empty);

element1.AppendChild(element2);

for (int c = 0; c < dane[0].Count; c++)

{

XmlElement element3 = doc.CreateElement(string.Empty, "atr" + c, string.Empty);

XmlText text1 = doc.CreateTextNode(dane[r][c]);

element3.AppendChild(text1);

element2.AppendChild(element3);

}

}

doc.Save(path);

}

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

{

daneZnormalizowane = dane;

for (int j = 0; j < dane[0].Count; j++)

{

double minValue = double.MaxValue; // przypisanie maxymalnej wartości

double maxValue = double.MinValue; // przypisanie minimalnej wartosci

for (int i = 0; i < dane.Count; i++)

{

if (Convert.ToDouble(dane[i][j].Replace(".", ",")) > maxValue) // sprawdzanie wartosci

{

maxValue = Convert.ToDouble(dane[i][j].Replace(".", ","));

}

if (Convert.ToDouble(dane[i][j].Replace(".", ",")) < minValue)

{

minValue = Convert.ToDouble(dane[i][j].Replace(".", ","));

}

}

for (int i = 0; i < dane.Count; i++)

{

daneZnormalizowane[i][j] = ((Convert.ToDouble(dane[i][j].Replace(".", ",")) - minValue) / (maxValue - minValue)).ToString(); // normalizacja danych

}

}

dataGridView1.Rows.Clear();

for (int i = 0; i < daneZnormalizowane.Count; i++)

{

DataGridViewRow deklaracja = new DataGridViewRow(); //deklaracja rekordu w grid

deklaracja.CreateCells(dataGridView1);

for (int j = 0; j < daneZnormalizowane[0].Count; j++)

{

deklaracja.Cells[j].Value = daneZnormalizowane[i][j];

}

dataGridView1.Rows.Add(deklaracja);

}

}

}

}