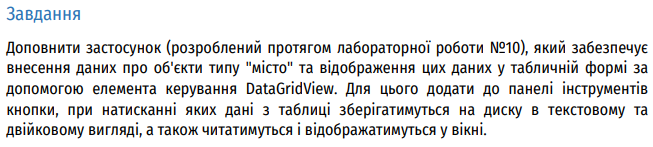
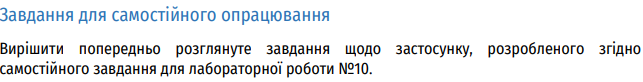
Звіт з лабораторної роботи №13





Лістинг (оновлений fMain.cs):

using System;

using System.Windows.Forms;

using System.IO;

using System.Text;

namespace Lab13

{

public partial class fMain : Form

{

public fMain()

{

InitializeComponent();

}

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

{

// Set exit button at right upper angle of form:

int buttonsSize = 9 \* btnAdd.Width + 3 \* tsSeparator1.Width + 30;

btnExit.Margin = new Padding(Width - buttonsSize, 0, 0, 0);

gvProcessors.AutoGenerateColumns = false;

DataGridViewColumn column = new DataGridViewTextBoxColumn

{

DataPropertyName = "Name",

Name = "Назва"

};

gvProcessors.Columns.Add(column);

column = new DataGridViewTextBoxColumn

{

DataPropertyName = "Socket",

Name = "Сокет"

};

gvProcessors.Columns.Add(column);

column = new DataGridViewTextBoxColumn

{

DataPropertyName = "Architecture",

Name = "Архітектура"

};

gvProcessors.Columns.Add(column);

column = new DataGridViewTextBoxColumn

{

DataPropertyName = "CoresCount",

Name = "К-сть ядер"

};

gvProcessors.Columns.Add(column);

column = new DataGridViewTextBoxColumn

{

DataPropertyName = "ClockFrequency",

Name = "Тактова частота"

};

gvProcessors.Columns.Add(column);

column = new DataGridViewTextBoxColumn

{

DataPropertyName = "EnergyConsumption",

Name = "Енергоспож.",

Width = 80

};

gvProcessors.Columns.Add(column);

column = new DataGridViewCheckBoxColumn

{

DataPropertyName = "HasIntegratedVideocard",

Name = "Є інтегр. граф.",

Width = 60

};

gvProcessors.Columns.Add(column);

column = new DataGridViewCheckBoxColumn

{

DataPropertyName = "HasUnlockedMultiplier",

Name = "Є розблок. множ.",

Width = 60

};

gvProcessors.Columns.Add(column);

EventArgs args = new EventArgs();

OnResize(args);

}

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

{

Processor processor = new Processor();

fProcessor ft = new fProcessor(processor);

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

{

bindSrcProcessors.Add(processor);

}

}

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

{

Processor processor = (Processor)bindSrcProcessors.List[bindSrcProcessors.Position];

fProcessor fp = new fProcessor(processor);

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

{

bindSrcProcessors.List[bindSrcProcessors.Position] = processor;

}

}

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

{

if (MessageBox.Show("Видалити поточний запис?",

"Видалення запису", MessageBoxButtons.OKCancel,

MessageBoxIcon.Warning) == DialogResult.OK)

{

bindSrcProcessors.RemoveCurrent();

}

}

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

{

if (MessageBox.Show("Очистити таблицю?\nВсі дані будуть втрачені",

"Очищення даних", MessageBoxButtons.OKCancel,

MessageBoxIcon.Question) == DialogResult.OK)

{

bindSrcProcessors.Clear();

}

}

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

{

if (MessageBox.Show("Закрити застосунок?",

"Вихід з програми", MessageBoxButtons.OKCancel,

MessageBoxIcon.Question) == DialogResult.OK)

{

Application.Exit();

}

}

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

{

saveFileDialog.Filter = "Текстові файли (\*.txt)|\*.txt|All files (\*.\*)|\*.\*";

saveFileDialog.Title = "Зберегти дані у текстовому форматі";

saveFileDialog.InitialDirectory = Application.StartupPath;

StreamWriter sw;

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

{

sw = new StreamWriter(saveFileDialog.FileName, false, Encoding.UTF8);

try

{

foreach (Processor processor in bindSrcProcessors.List)

{

sw.Write(processor.Name + "\t" + processor.Socket + "\t" +

processor.Architecture + "\t" + processor.CoresCount + "\t" +

processor.ClockFrequency + "\t" + processor.EnergyConsumption + "\t" +

processor.HasIntegratedVideocard + "\t" + processor.HasUnlockedMultiplier + "\t\n");

}

}

catch (Exception ex)

{

MessageBox.Show("Сталась помилка: \n{0}", ex.Message,

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

finally

{

sw.Close();

}

}

}

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

{

saveFileDialog.Filter = "Файли даних (\*.processors)|\*.towns|All files (\*.\*)|\*.\*";

saveFileDialog.Title = "Зберегти дані у бінарному форматі";

saveFileDialog.InitialDirectory = Application.StartupPath;

BinaryWriter bw;

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

{

bw = new BinaryWriter(saveFileDialog.OpenFile());

try

{

foreach (Processor processor in bindSrcProcessors.List)

{

bw.Write(processor.Name);

bw.Write(processor.Socket);

bw.Write(processor.Architecture);

bw.Write(processor.CoresCount);

bw.Write(processor.ClockFrequency);

bw.Write(processor.EnergyConsumption);

bw.Write(processor.HasIntegratedVideocard);

bw.Write(processor.HasUnlockedMultiplier);

}

}

catch (Exception ex)

{

MessageBox.Show("Сталась помилка: \n{0}", ex.Message,

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

finally

{

bw.Close();

}

}

}

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

{

openFileDialog.Filter = "Текстові файли (\*.txt)|\*.txt|All files (\*.\*) | \*.\* ";

openFileDialog.Title = "Прочитати дані у текстовому форматі";

openFileDialog.InitialDirectory = Application.StartupPath;

StreamReader sr;

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

{

bindSrcProcessors.Clear();

sr = new StreamReader(openFileDialog.FileName, Encoding.UTF8);

string s;

try

{

while ((s = sr.ReadLine()) != null)

{

string[] split = s.Split('\t');

Processor processor = new Processor(split[0], split[1], split[2],

int.Parse(split[3]), double.Parse(split[4]), double.Parse(split[5]), bool.Parse(split[6]), bool.Parse(split[7]));

bindSrcProcessors.Add(processor);

}

}

catch (Exception ex)

{

MessageBox.Show("Сталась помилка: \n{0}", ex.Message,

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

finally

{

sr.Close();

}

}

}

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

{

openFileDialog.Filter = "Файли даних (\*.towns)|\*.towns|All files (\*.\*) | \*.\* ";

openFileDialog.Title = "Прочитати дані у бінарному форматі";

openFileDialog.InitialDirectory = Application.StartupPath;

BinaryReader br;

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

{

bindSrcProcessors.Clear();

br = new BinaryReader(openFileDialog.OpenFile());

try

{

Processor processor;

while (br.BaseStream.Position < br.BaseStream.Length)

{

processor = new Processor();

for (int i = 1; i <= 8; i++)

{

switch (i)

{

case 1: processor.Name = br.ReadString(); break;

case 2: processor.Socket = br.ReadString(); break;

case 3: processor.Architecture = br.ReadString(); break;

case 4: processor.CoresCount = br.ReadInt32(); break;

case 5: processor.ClockFrequency = br.ReadDouble(); break;

case 6: processor.EnergyConsumption = br.ReadDouble(); break;

case 7: processor.HasIntegratedVideocard = br.ReadBoolean(); break;

case 8: processor.HasUnlockedMultiplier = br.ReadBoolean(); break;

}

}

bindSrcProcessors.Add(processor);

}

}

catch (Exception ex)

{

MessageBox.Show("Сталась помилка: \n{0}", ex.Message, MessageBoxButtons.OK, MessageBoxIcon.Error);

}

finally

{

br.Close();

}

}

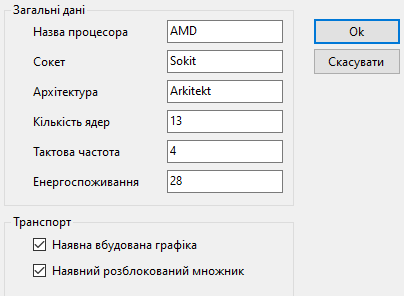
}

}

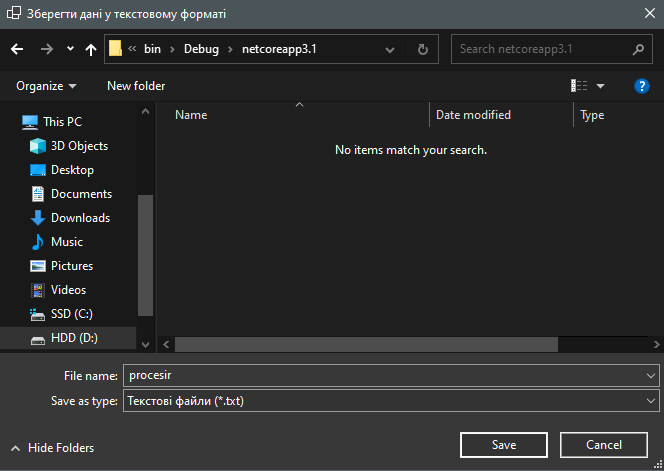
}

Результат:

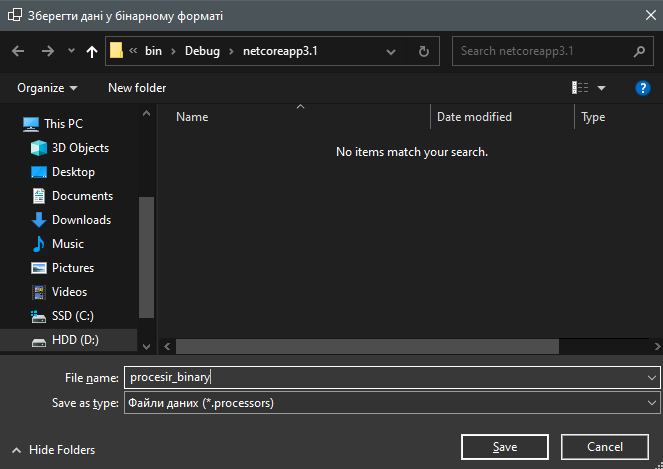
1. Додавання нових даних:



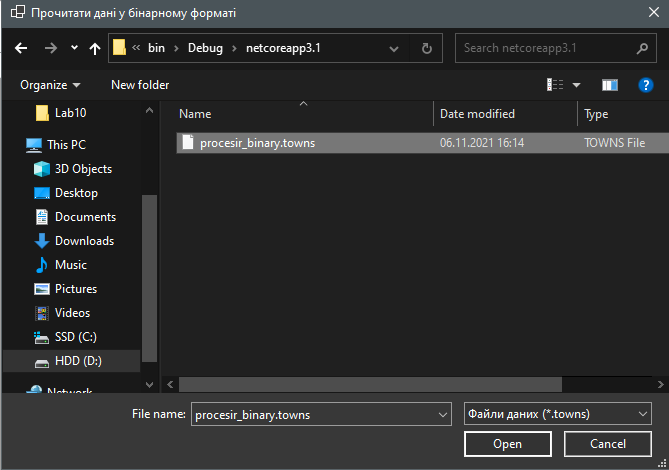
1. Збереження файлу в текстовому форматі:



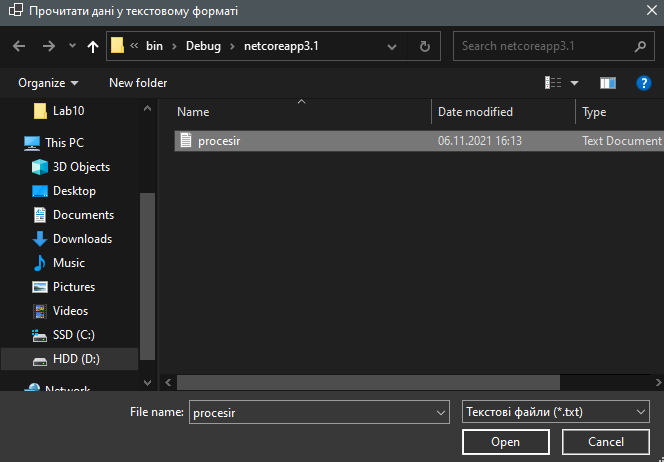
1. Збереження в двійковому форматі:



1. Запит на читання двійкового файлу:



1. Запит на читання текстового файлу:



**Кінець звіту**