Министерство науки и высшего образования Российской Федерации

**Муромский институт (филиал)**

федерального государственного бюджетного образовательного учреждения высшего образования

**«Владимирский государственный университет**

**Имени Александра Григорьевича и Николая Григорьевича Столетовых»**

**(МИВлГУ)**

Факультет ИТР

Кафедра ПИн

ЛАБОРАТОРНАЯ

РАБОТА №5

по Тестирование ПO

Тема Системное тестирование

Руководитель

Колпаков А. А.

(фамилия, инициалы)

(подпись) (дата)

Студент ПИн-121

(группа)

Ермилов М.В.

(фамилия, инициалы)

(подпись) (дата)

Муром 2024

**Тема: Системное тестирование**

**Цель работы:** Изучить и освоить основные принципы системного тестирования.

**Задание на лабораторную работу:**

1. Изучить материал, представленный в данных методических указаниях к лабораторной работе;

2. Разработать приложение для построения графиков на основе данных из приложенных файлов. Обеспечить совместимость «из коробки» с Windows 7. В отдельные библиотеки классов вынести: 1. работу с файлами; 2. многопоточную медианную фильтрацию;

3. нормализацию графиков 3. Провести системное тестирование готового приложения;

4. Составить отчет. В отчете отобразить: − цели и задачи лабораторной работы; − личное задание; − результаты тестирования; − исходный код классов; − выводы по данной лабораторной работе

**WindowsFormsApp1.cs**

using System;

using System.Collections.Generic;

using System.IO;

using System.Windows.Forms;

using System.Windows.Forms.DataVisualization.Charting;

namespace WindowsFormsApp1

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

// Установка свойства FormBorderStyle

this.FormBorderStyle = FormBorderStyle.FixedSingle;

// Запрет разворачивания формы на весь экран

this.MaximizeBox = false;

}

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

{

OpenFileDialog openFileDialog = new OpenFileDialog();

openFileDialog.Filter = "Текстовые файлы (\*.txt)|\*.txt|Все файлы (\*.\*)|\*.\*";

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

{

string filePath = openFileDialog.FileName;

tBpath.Text = filePath;

// Очистим ListBox перед новым чтением файла

lBdata.Items.Clear();

// Прочитаем данные из файла и добавим каждую строку в ListBox

try

{

using (StreamReader reader = new StreamReader(filePath))

{

string line;

while ((line = reader.ReadLine()) != null)

{

lBdata.Items.Add(line);

}

}

}

catch (IOException ex)

{

MessageBox.Show("Ошибка при чтении файла: " + ex.Message, "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

FilterButton.Enabled = true;

}

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

{

// Проверим, что есть данные в ListBox

if (lBdata.Items.Count == 0)

{

MessageBox.Show("No data to filter. Please load a file first.", "Info", MessageBoxButtons.OK, MessageBoxIcon.Information);

return;

}

List<string> lines = new List<string>();

// Пройдемся по каждой строке в ListBox

foreach (string line in lBdata.Items)

{

string[] columns = null;

if(line.Contains("\t"))

columns = line.Split('\t');

else columns = line.Split(',');

string newLine = columns[0] + "," + columns[1] + "," + columns[3]

+ "," + columns[4] + "," + columns[5] + "," + columns[6] + "," + columns[12];

lines.Add(newLine);

}

lBdata.Items.Clear();

foreach(string s in lines)

{

lBdata.Items.Add(s);

}

FilterButton.Enabled = false;

WriteButton.Enabled = true;

}

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

{

chart1.Series.Clear();

Series series1 = new Series();

series1.Name = "Indoor Humidity";

Series series2 = new Series();

series2.Name = "Indoor Temperature";

Series series3 = new Series();

series3.Name = "Outdoor Humidity";

Series series4 = new Series();

series4.Name = "Outdoor Temperature";

Series series5 = new Series();

series5.Name = "Dewpoint";

int i = 0;

foreach (string line in lBdata.Items)

{

if (i > 0)

{

string[] columns = line.Split(',');

string[] time = null;

if (columns[1].Contains(" "))

time = columns[1].Split(' ');

series1.Points.Add(new DataPoint(ConvertToMinutes(time[1]), columns[2]));

DataPoint point = new DataPoint(ConvertToMinutes(time[1]), RemoveDecimalPart(columns[3]));

series2.Points.Add(point);

point = new DataPoint(ConvertToMinutes(time[1]), RemoveDecimalPart(columns[4]));

series3.Points.Add(point);

point = new DataPoint(ConvertToMinutes(time[1]), RemoveDecimalPart(columns[5]));

series4.Points.Add(point);

point = new DataPoint(ConvertToMinutes(time[1]), RemoveDecimalPart(columns[6]));

series5.Points.Add(point);

}

i++;

}

series1.ChartType = SeriesChartType.Line;

series2.ChartType = SeriesChartType.Line;

series3.ChartType = SeriesChartType.Line;

series4.ChartType = SeriesChartType.Line;

series5.ChartType = SeriesChartType.Line;

chart1.Series.Add(series1);

chart1.Series.Add(series2);

chart1.Series.Add(series3);

chart1.Series.Add(series4);

chart1.Series.Add(series5);

WriteButton.Enabled = false;

}

static double ConvertToMinutes(string timeString)

{

// Разбиваем строку времени на часы и минуты

string[] timeParts = timeString.Split(':');

// Парсим часы и минуты из строкового представления в числа

if (timeParts.Length == 2 && int.TryParse(timeParts[0], out int hours) && int.TryParse(timeParts[1], out int minutes))

{

// Преобразуем часы в минуты и складываем с минутами

return hours \* 60 + minutes;

}

else

{

// В случае ошибки ввода возвращаем значение, которое можно обработать

return -1;

}

}

static string RemoveDecimalPart(string input)

{

// Находим индекс точки в строке

int dotIndex = input.IndexOf('.');

// Если точка не найдена, возвращаем исходную строку

if (dotIndex == -1)

{

return input;

}

// Обрезаем строку до точки, включительно

string result = input.Substring(0, dotIndex + 1);

return result;

}

}

}

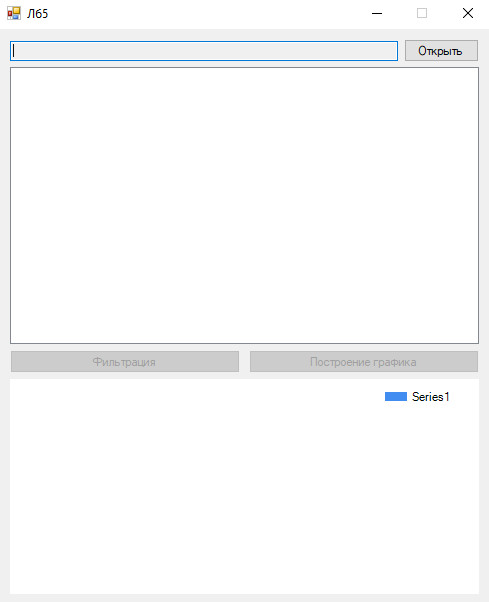


Рисунок 1 - Отображение формы при открытии

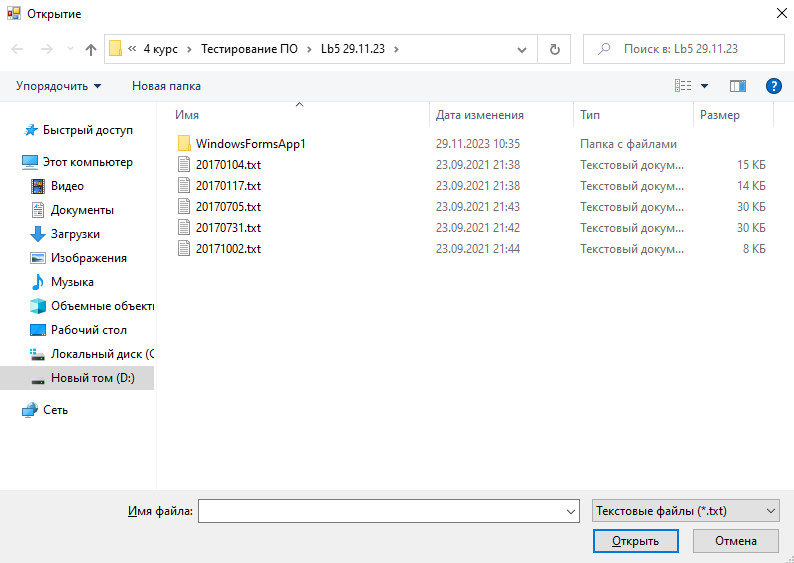


Рисунок 2 - Открытие файлового диалога, для выбора необходимого файла

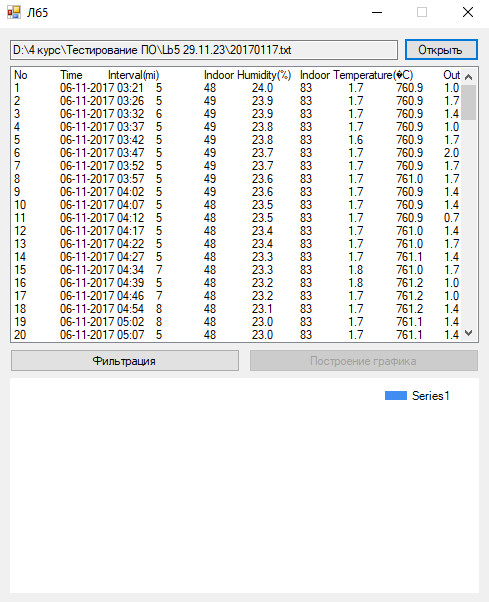


Рисунок 3 - Отображение загруженных файлов на форме

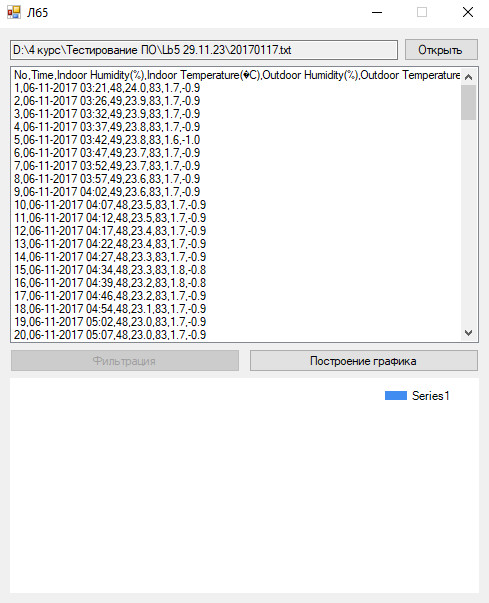


Рисунок 4 - Фильтрация загруженных данных

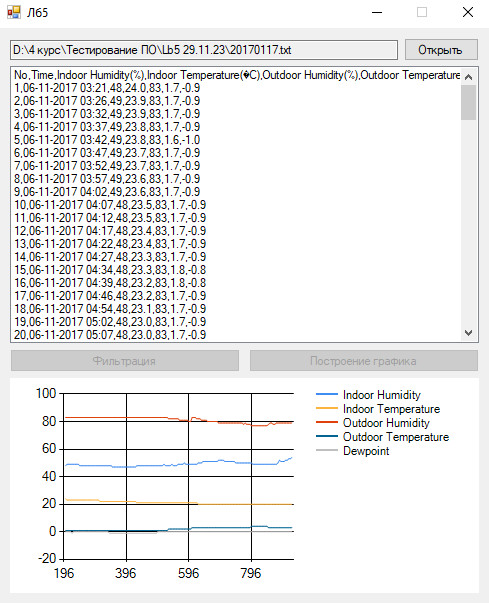


Рисунок 5 - Отображение графиков с нормализованными данными

**Вывод:** в ходе выполнения лабораторной работы были изучены и освоены основные принципы построения модульных тестов на основе классов в Visual Studio.