**Тема 7: Построение графиков и диаграмм**

**Цель работы:** получить практические навыки разработки проектов с использованием стандартного компонента chart для построения графиков, диаграмм, зависимостей по таблице данных.

**Вариант 5.**

Разработать проект для ввода данных о количестве пятёрок, четверок, троек, двоек и не аттестованных по разным дисциплинам для каждой группы. Программа должна вычислять абсолютную и качественную успеваемость в процентах по каждому предмету. Абсолютная успеваемость рассчитывается как отношение суммы оценок без двоек к сумме всех оценок (в процентах), качественная успеваемость рассчитывается как сумма четвёрок и пятёрок к сумме всех оценок (в процентах). Результаты сохранять и отображать на второй вкладке в виде таблицы. Также должна быть возможность загрузить данные из текстового файла (номер группы, названия учебных дисциплин, количество пятёрок, четверок, троек, двоек и не аттестованных по разным дисциплинам) и сохранить данные в текстовый файл в таком же формате. На третьей вкладке построить столбчатую гистограмму, показывающую уровень успеваемости по разным дисциплинам (по выбранной группе).

**Задание 1**: Основной проект

Разработать программу для ввода данных по вашему варианту задания и добавления в таблицу вводимых данных. Опрос пользователей производится при помощи отдельной вкладки на форме, результаты опроса в виде таблицы должны отображаться на другой вкладке. Для организации вкладок используйте компонент TabControl. Хранение и обработку данных организовать с помощью дополнительного класса.

**Задание 2:** Сохранение и загрузка данных

Добавьте возможность сохранить результаты опроса в текстовый файл и загрузить результаты опроса из текстового файла в таблицу на форме. После загрузки файла можно продолжить ввод данных, причем результаты нового голосования суммируются с загруженными результатами. Также при попытке закрытия формы программа должна предлагать сохранение результатов голосования, если они не были сохранены.

**Задание 3:** Построение диаграммы

Добавьте в проект 3-ую вкладку, на которой требуется построить график или диаграмму (в зависимости от варианта). Построение диаграммы производиться по таблице данных с помощью компонента Сhart.

Для работы с этим компонентом необходимо подключить библиотеку System.Windows.Forms.DataVisualization.Charting

1. **Разработка подпрограмм**
   1. **Разработка подпрограммы Attestat**
      1. **Математическая модель**

**Дано:**

group – номер группы;

obj - предмет, по которому ведется аттестационная ведомость;

fives – количество пятерок;

fours – количество четверок;

threes – количество троек;

twos – количество двоек;

not\_att – количество не аттестованных;

**ОДЗ:** дополнительных ограничений нет

*Найти:*

abs – абсолютная успеваемость по предмету;

qual – качественная успеваемость по предмету.

**ОДЗ:** дополнительных ограничений нет

* + 1. **Код подпрограммы** **Attestat**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.IO;

using System.Windows.Forms;

namespace \_7lab

{

public class Attestat

{

public string group { get; set; }

public string obj { get; set; }

public int five { get; set; }

public int four { get; set; }

public int three { get; set; }

public int two { get; set; }

public int not\_att { get; set; }

public double abs

{

get

{

int totalGrades = five + four + three + two + not\_att; // Общее количество оценок

int sumWithoutTwos = five + four + three; // Сумма оценок без двоек

return (double)sumWithoutTwos / totalGrades \* 100;

}

}

public double qual

{

get

{

int totalGrades = five + four + three + two + not\_att; // Общее количество оценок

int sumFoursAndFives = five + four; // Сумма четвёрок и пятёрок

return (double)sumFoursAndFives / totalGrades \* 100;

}

}

public Attestat()

{

}

// Метод для записи данных в файл

public void SaveToFile(string filename)

{

try

{

using (StreamWriter writer = new StreamWriter(filename, true))

{

writer.WriteLine(group);

writer.WriteLine(obj);

writer.WriteLine($"5,{five}");

writer.WriteLine($"4,{four}");

writer.WriteLine($"3,{three}");

writer.WriteLine($"2,{two}");

writer.WriteLine($"н/а,{not\_att}");

// Добавляем запись абсолютной и качественной успеваемости

writer.WriteLine(abs);

writer.WriteLine(qual);

}

}

catch

{

throw new Exception("Ошибка при сохранении данных: ");

}

}

public Attestat(string group, string obj, int five, int four, int three, int two, int not\_att, double absalut, double quality)

{

this.group = group;

this.obj = obj;

this.five = five;

this.four = four;

this.three = three;

this.two = two;

this.not\_att = not\_att;

}

public static List<Attestat> LoadFromFile1(string filename)

{

List<Attestat> attList = new List<Attestat>();

try

{

using (StreamReader reader = new StreamReader(filename))

{

// проверяем, что файл не пустой

if (reader.EndOfStream)

{

throw new Exception("Файл пуст");

}

// проверяем, что первая строка содержит имя группы

string group = reader.ReadLine();

if (string.IsNullOrWhiteSpace(group))

{

MessageBox.Show("Неверный формат файла: отсутствует имя группы");

}

// проверяем, что вторая строка содержит имя предмета

string obj = reader.ReadLine();

if (string.IsNullOrWhiteSpace(obj))

{

MessageBox.Show("Неверный формат файла: отсутствует имя предмета");

}

// читаем оценки и проверяем, что они соответствуют формату "оценка,количество"

string[] grades = reader.ReadLine().Split(',');

if (grades.Length != 2)

{

MessageBox.Show("Неверный формат файла: неверный формат оценок");

}

int five = int.Parse(grades[1]);

string[] grades1 = reader.ReadLine().Split(',');

if (grades1.Length != 2)

{

MessageBox.Show("Неверный формат файла: неверный формат оценок");

}

int four = int.Parse(grades1[1]);

string[] grades2 = reader.ReadLine().Split(',');

if (grades2.Length != 2)

{

MessageBox.Show("Неверный формат файла: неверный формат оценок");

}

int three = int.Parse(grades2[1]);

string[] grades3 = reader.ReadLine().Split(',');

if (grades3.Length != 2)

{

MessageBox.Show("Неверный формат файла: неверный формат оценок");

}

int two = int.Parse(grades3[1]);

string[] grades4 = reader.ReadLine().Split(',');

if (grades4.Length != 2)

{

MessageBox.Show("Неверный формат файла: неверный формат оценок");

}

int not\_att = int.Parse(grades4[1]);

// проверяем, что файл содержит две строки с абсолютной и качественной успеваемостью

double absalut = double.Parse(reader.ReadLine());

double quality = double.Parse(reader.ReadLine());

// создаем экземпляр класса Attestat и добавляем его в список

Attestat att = new Attestat(group, obj, five, four, three, two, not\_att, absalut, quality);

attList.Add(att);

}

}

catch (Exception ex)

{

throw new Exception("Ошибка при загрузке данных: " + ex.Message);

}

return attList;

}

}

}

* + 1. **Код модуля Form1.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Windows.Forms.DataVisualization.Charting;

using System.Windows.Forms;

using System.IO;

namespace \_7lab

{

public partial class Form1 : Form

{

SaveFileDialog saveFileDialog;

string name ="";

int ck = 0;

public Form1()

{

InitializeComponent();

}

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

{

if (name!="")

{

Attestat at = new Attestat();

at.group = comboBox1.Text;

at.obj = comboBox2.Text;

at.five = (int)numericUpDown1.Value;

at.four = (int)numericUpDown2.Value;

at.three = (int)numericUpDown3.Value;

at.two = (int)numericUpDown4.Value;

at.not\_att = (int)numericUpDown5.Value;

at.SaveToFile(name);

}

else

{ ck++;

saveFileDialog = new SaveFileDialog();

saveFileDialog.Filter = "TXT files (\*.txt)|\*.txt";

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

{

Attestat at = new Attestat();

at.group = comboBox1.Text;

at.obj = comboBox2.Text;

at.five = (int)numericUpDown1.Value;

at.four = (int)numericUpDown2.Value;

at.three = (int)numericUpDown3.Value;

at.two = (int)numericUpDown4.Value;

at.not\_att = (int)numericUpDown5.Value;

name = saveFileDialog.FileName;

at.SaveToFile(saveFileDialog.FileName);

}

}

}

private void comboBox3\_TextChanged(object sender, EventArgs e)

{

chart1.Series.Clear();

chart1.ChartAreas[0].BackColor = Color.White;

chart1.Series.Add(new Series() { ChartType = SeriesChartType.Bar });

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

List<double> listy = new List<double>();

List<Attestat> attList = Attestat.LoadFromFile1(name);

foreach (Attestat pr in attList)

{

if (pr.group == comboBox3.Text)

{

list.Add(pr.obj.ToString());

listy.Add(pr.abs);

}

}

string[] xValues =list.ToArray();

double[] yValues = listy.ToArray() ;

chart1.ChartAreas[0].AxisY.Title = "Успеваемость";

chart1.ChartAreas[0].AxisX.IsReversed = true;

chart1.Series[0].Points.DataBindXY(xValues, yValues);

}

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

{

groupBox1.Enabled = true;

groupBox1.Visible = true;

}

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

{

saveFileDialog = new SaveFileDialog();

saveFileDialog.Filter = "TXT files (\*.txt)|\*.txt";

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

{

// Записываем данные в файл

using (StreamWriter writer = new StreamWriter(saveFileDialog.FileName))

{

// Записываем данные из таблицы данных в файл

foreach (DataGridViewRow row in dataGridView1.Rows)

{

writer.WriteLine(row.Cells["Номер группы"].Value);

writer.WriteLine(row.Cells["Предмет"].Value);

writer.WriteLine($"5,{row.Cells["5 (отл)"].Value}");

writer.WriteLine($"4,{row.Cells["4 (хор)"].Value}");

writer.WriteLine($"3,{row.Cells["3 (удв)"].Value}");

writer.WriteLine($"2,{row.Cells["2 (неудв)"].Value}");

writer.WriteLine($"н/а,{row.Cells["не аттестованно"].Value}");

writer.WriteLine(row.Cells["Абсолютная успеваемость"].Value);

writer.WriteLine(row.Cells["Качественная успеваемость"].Value);

}

name = saveFileDialog.FileName;

}

MessageBox.Show("Данные успешно экспортированы!");

}

}

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

{

// Выбираем файл с данными

OpenFileDialog openFileDialog = new OpenFileDialog();

openFileDialog.Filter = "TXT files (\*.txt)|\*.txt";

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

{

DataTable dataTable = new DataTable();

dataTable.Columns.Add("Номер группы", typeof(string));

dataTable.Columns.Add("Предмет", typeof(string));

dataTable.Columns.Add("5 (отл)", typeof(int));

dataTable.Columns.Add("4 (хор)", typeof(int));

dataTable.Columns.Add("3 (удв)", typeof(int));

dataTable.Columns.Add("2 (неудв)", typeof(int));

dataTable.Columns.Add("не аттестованно", typeof(int));

dataTable.Columns.Add("Абсолютная успеваемость", typeof(int));

dataTable.Columns.Add("Качественная успеваемость", typeof(int));

foreach (var val in Attestat.LoadFromFile1(openFileDialog.FileName))

{

dataTable.Rows.Add(val.group, val.obj, val.five, val.four, val.three, val.two, val.not\_att, val.abs, val.qual);

}

foreach (DataGridViewColumn column in dataGridView1.Columns)

{

column.AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;

}

name = openFileDialog.FileName;

dataGridView1.DataSource = dataTable;

dataGridView1.RowHeadersVisible = false;

}

}

private void dataGridView1\_MouseEnter(object sender, EventArgs e)

{

if (name != "")

{

DataTable dataTable = new DataTable();

dataTable.Columns.Add("Номер группы", typeof(string));

dataTable.Columns.Add("Предмет", typeof(string));

dataTable.Columns.Add("5 (отл)", typeof(int));

dataTable.Columns.Add("4 (хор)", typeof(int));

dataTable.Columns.Add("3 (удв)", typeof(int));

dataTable.Columns.Add("2 (неудв)", typeof(int));

dataTable.Columns.Add("не аттестованно", typeof(int));

dataTable.Columns.Add("Абсолютная успеваемость", typeof(int));

dataTable.Columns.Add("Качественная успеваемость", typeof(int));

foreach (var val in Attestat.LoadFromFile1(name))

{

dataTable.Rows.Add(val.group, val.obj, val.five, val.four, val.three, val.two, val.not\_att, val.abs, val.qual);

}

foreach (DataGridViewColumn column in dataGridView1.Columns)

{

column.AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;

}

dataGridView1.DataSource = dataTable;

}

}

}

**1.1.4 Алгоритмы**

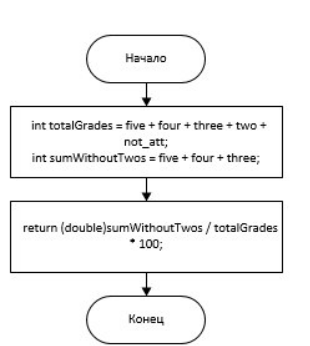


Рис. 1.1.1-алгоритм метода abs

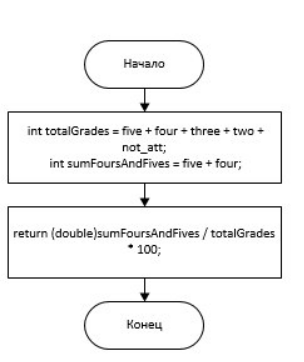
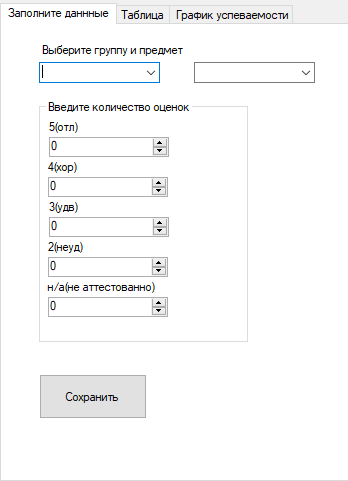


Рис. 1.1.2-алгоритм метода qual

1. **Создание пользовательского интерфейса**
   1. **Проектирование пользовательского интерфейса**

Первая вкладка проекта представлена на рис. 2.1, вторая и третья вкладка представлены на рис. 2.2 и рис.2.3 соответственно. Значения измененных свойств компонентов представлены в таблице 2.1. 2.2, 2.3 и 2.4.

****

label5

label4

label3

label2

label1

groupBox1

label6

comboBox1

comboBox2

tabControl1

numericUpDown1

button1

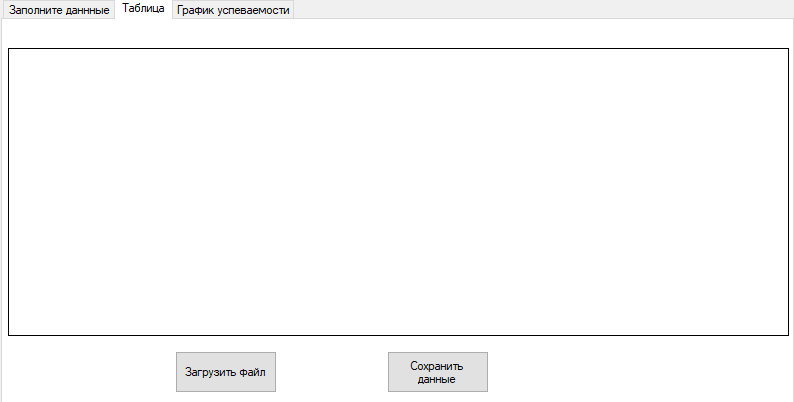
numericUpDown5

numericUpDown4

numericUpDown3

numericUpDown2

**Рис. 2.1**

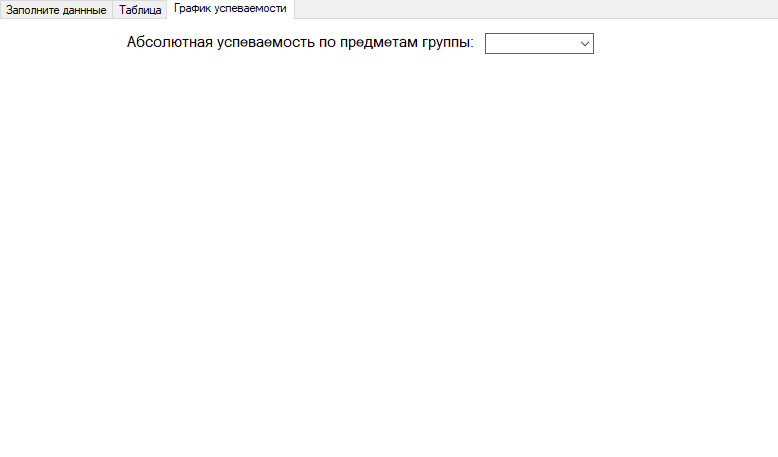
****

dataGridView1

Button2

Button3

**Рис. 2.2**

**Рис. 2.3**

label7

chart1

comboBox3

* 1. **.Свойства компонентов формы**

*Таблица 2.1.* Значения свойств, установленные на этапе конструирования интерфейса

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| form1 | Text | «Построение графиков и диаграмм» | | |
| tabControl1 | TabPages | tabPage1 | Text | «Заполните даннные» |
| tabPage2 | Text | «Таблица» |
| tabPage2 | Text | «График успеваемости» |

*Таблица 2.2.* Значения свойств, установленные на этапе конструирования интерфейса

|  |  |  |
| --- | --- | --- |
| groupBox1 | Text | «Введите количество оценок» |
| label1 | Text | «5(отл)» |
| label2 | Text | «4(хор)» |
| label3 | Text | «3(удв)» |
| label4 | Text | «2(неуд)» |
| label5 | Text | «н/а(не аттестованно)» |
| label6 | Text | «Выберите группу и предмет» |
| button1 | Text | «Сохранить» |

*Таблица2.3.* Значения свойств, установленные на этапе конструирования интерфейса

|  |  |  |
| --- | --- | --- |
| button2 | Location | 170; 331 |
| Size | 102; 42 |
| button3 | Text | «Загрузить файл» |
| Text | «Сохранить данные» |
| dataGridView1 | Location | 3; 28 |
| Size | 781; 288 |
| BorderStyle | FixedSingle |

*Таблица 2.4.* Значения свойств, установленные на этапе конструирования интерфейса

|  |  |  |
| --- | --- | --- |
| Label7 | Text | «Абсолютная успеваемость по предметам группы» |

* 1. **Сценарий использования программы**
     1. Запуск программы.
     2. Выбор группы из выпадающего списка comboBox1.
     3. Выбор предмета из выпадающего списка comboBox2.
     4. Ввод количества студентов , получивших оценки "5", "4", "3", "2" и "не аттестованно" в соответствующие поля numericUpDown1-5 в groupBox1.
     5. Нажатие кнопки "Сохранить" (button1) для обработки данных и выдачи соответствующих ответов.
     6. Выбор группы из выпадающего списка comboBox3 для просмотра графика успеваемости по предметам.
     7. После выбора группы в графике chart1 отображается столбчатая диаграмма успеваемости по предметам.
     8. Выбор пункта "Экспорт в файл" (button3) для сохранения данных из таблицы dataGridView1 в файл. После выбора пункта программа предлагает выбрать файл для сохранения с помощью диалогового окна saveFileDialog. После выбора файла данные из таблицы сохраняются в файл.
     9. Выбор пункта "Импорт из файла" (button2) для загрузки данных из файла в таблицу dataGridView1. После выбора пункта программа предлагает выбрать файл для загрузки с помощью диалогового окна openFileDialog. После выбора файла данные из файла загружаются в таблицу.
     10. Возможность вернуться к пнктам 1.1.2-1.1.9
     11. Завершение работы с программой.
  2. **Протокол испытаний**

*Таблица 3.1. протокол испытаний*

|  |  |  |  |
| --- | --- | --- | --- |
| **Проверяемые требования** | **Сообщения программы и вводимые значения** | **Ожидаемые результаты** | **Фактические результаты** |
| Способность программы обеспечить ввод данных и сохранение введеных данных во время работы программы | Ввести количество пятерок, четверок, троек, двоек и не аттестаций. Нажать на кнопку «Сохранить» | При переключении групп или предметов данные по предмету по которому были введены будут соответствовать введенным | Соответствуют ожидаемым результатам (см. Рисунки 3.1.1-3.1.2) |
| Способность программы обеспечить просмотр данных о абсолютной и качественной успеваемости | При значениях оценок: 5-11, 4-22, 3-30, 2-30, на-30. Перейти на вкладку Таблица | При переключении на против предмета будет подсчитана абсолютная успеваемость = 61 и качественная = 32 | Соответствуют ожидаемым результатам (см. Рисунок 3.1.3) |
| При значениях оценок: 5-11, 4-22, 3-30, 2-30, на-30. Перейти на вкладку Диаграмма | При переключении в столбцах, отвечающих за данные по предмету, будут совпадать по шкале с абсолютной успеваемости | Соответствуют ожидаемым результатам (см. Рисунок 3.1.4) |
| Способность программы обеспечить выдачу ошибки при загрузке файла неправильного формата | Нажать на кнопку «Загрузить из файла». Выбрать файл с неправильным форматом данных | После закрытия диалога открытия файла появится сообщение об ошибке загрузки файла | Соответствуют ожидаемым результатам (см. Рисунки 3.1.5-3.1.6) |
| Способность программы обеспечить загрузку данных из файла | Нажать на кнопку «Загрузить из файла». Выбрать исправный файл | После закрытия диалога открытия файла появятся обновленные данные в таблице и обновятся в списке групп и предметов | Соответствуют ожидаемым результатам (см. Рисунки 3.1.7-3.1.9) |
| Способность программы обеспечить выгрузку данных в файл | Нажать на кнопку «Сохранить в файл». Выбрать файл для сохранения в него данных | После закрытия диалога сохранения файла и его открытия в проводнике данные в нем будут соответствовать данным с формы | Соответствуют ожидаемым результатам (см. Рисунки 3.1.10-3.1.11) |

* 1. **Результаты тестировния проекта**

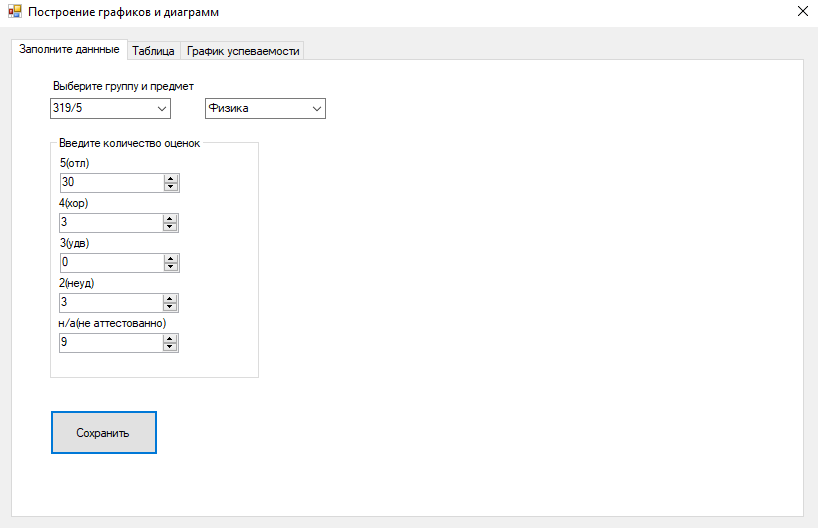
****

Рис. 3.1.1

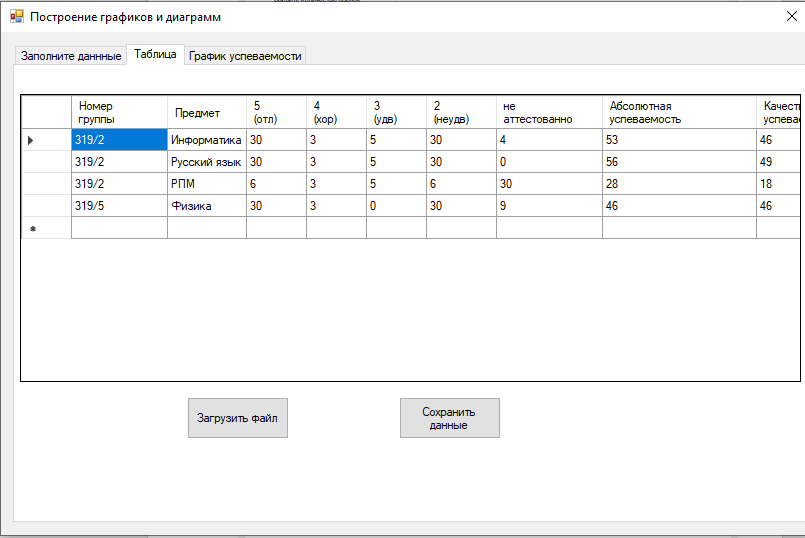
****

Рис.3.1.2

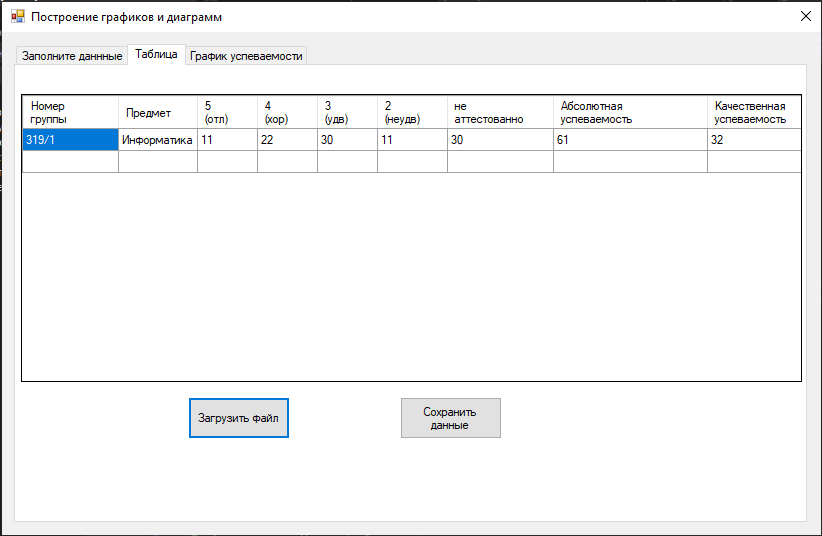
****

Рис.3.1.3

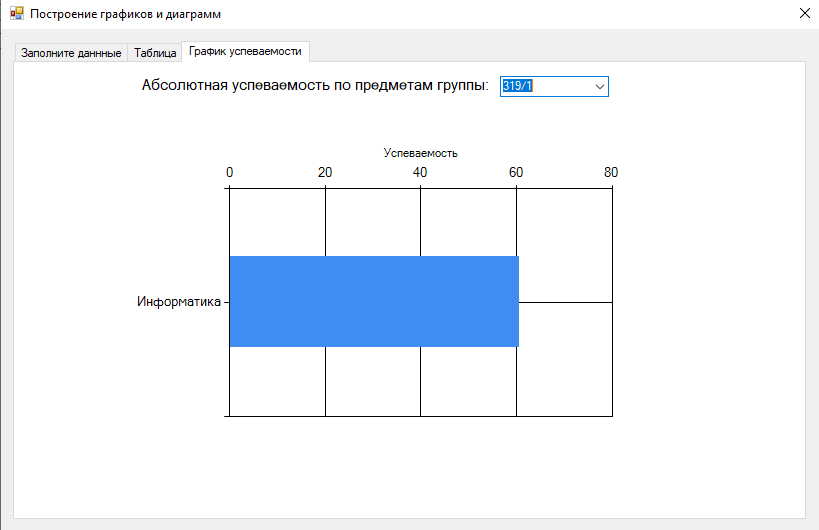
****

Рис.3.1.4

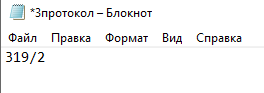
****

Рис.3.7.5

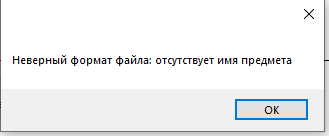
****

Рис.3.1.6

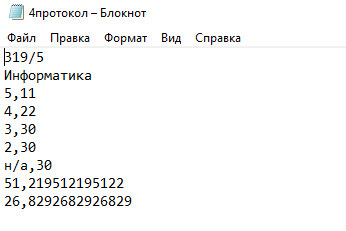
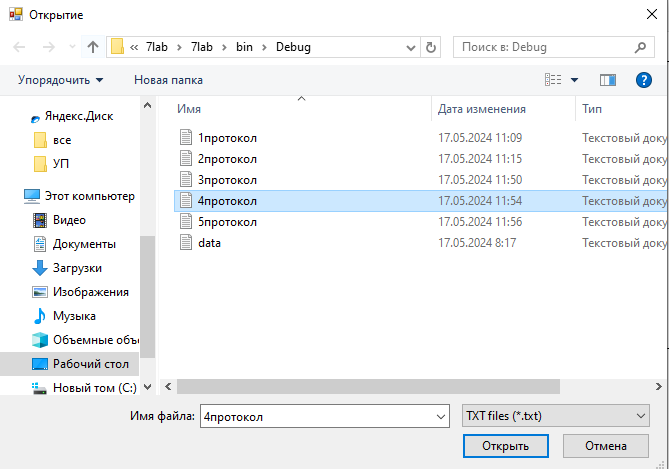
****

Рис.3.1.7



3.1.8

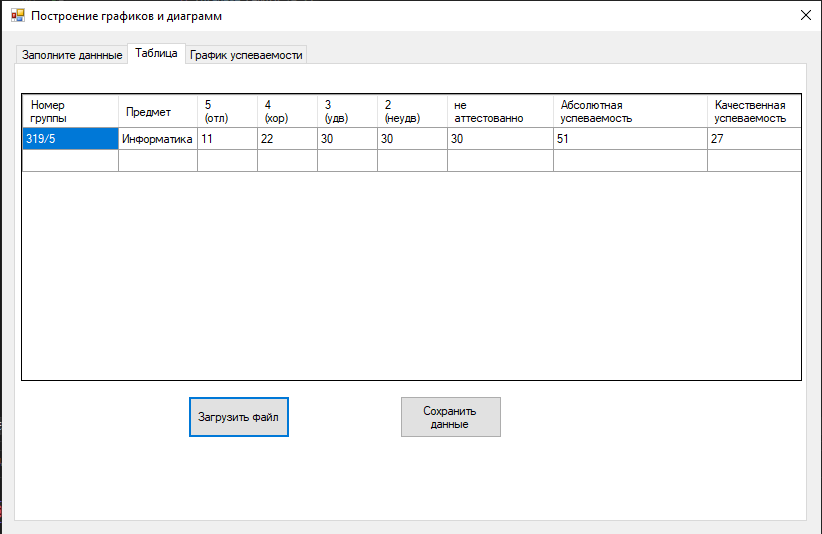
****

Рис.3.1.9

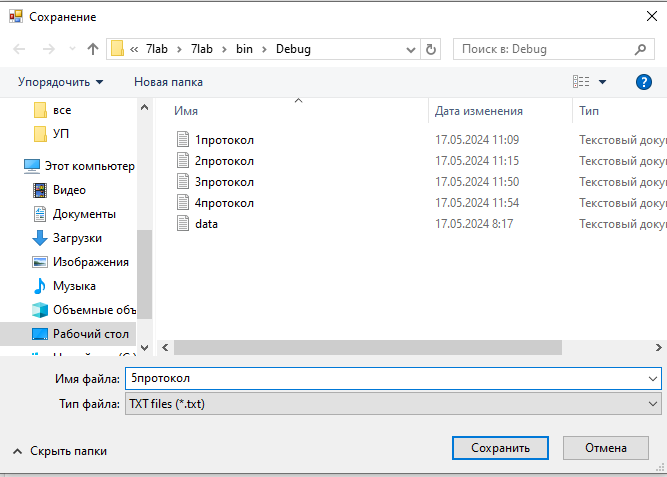
****

Рис.3.1.10

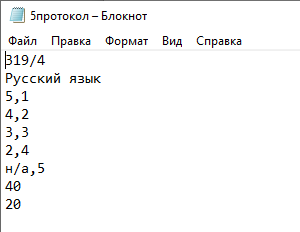
****

Рис.3.1.11