Министерство транспорта Российской Федерации

Федеральное государственное автономное образовательное

учреждение высшего образования

«Российский университет транспорта»

(ФГАОУ ВО РУТ(МИИТ), РУТ (МИИТ)

Институт транспортной техники и систем управления

Кафедра «Управление и защита информации»

Лабораторная работа № 20

по дисциплине: «Программирование и основы алгоритмизации»

на тему: «Диаграммы»

Выполнил: ст. гр. ТУУ-211

Голюдбин А.В.

Вариант №3

Проверил: к.т.н., доц. Сафронов А.И.

1. **Формулировка цели работы**

Освоить навыки графического отображения статистических данных и построения диаграмм на платформе Windows Forms.

1. **Описание задачи**



Рисунок 1 – Иллюстрация задачи

Загружать данные, представленные на диаграммах, в программное обеспечение из csv-файла. Воспроизвести отображение представленных двух видов диаграмм. Предоставить пользователю возможность обмена положением проиллюстрированных диаграмм, а также возможность отображения только одного из видов диаграмм.

1. **Сеть Петри запрограммированного технологического процесса**

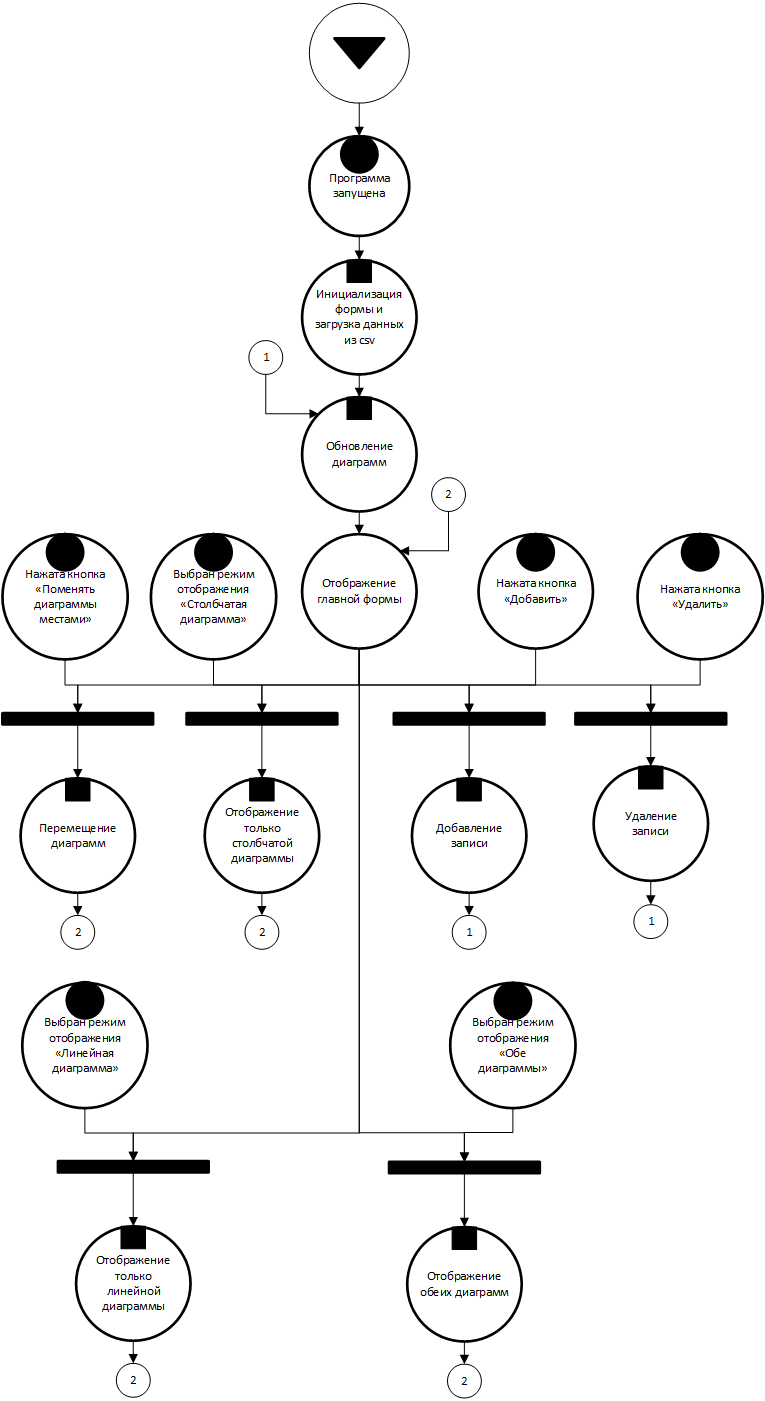


Рисунок 2 – Сеть Петри запрограммированного технологического процесса

1. **Схемы алгоритмов методов в составе решения, отмеченных на сети Петри в качестве «эффектов» (метка )**

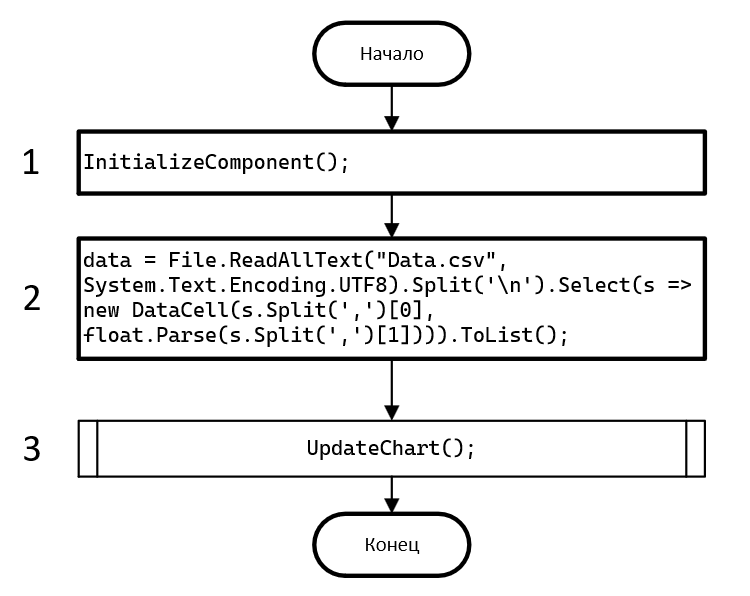


Рисунок 3 – Инициализация формы и загрузка данных из csv

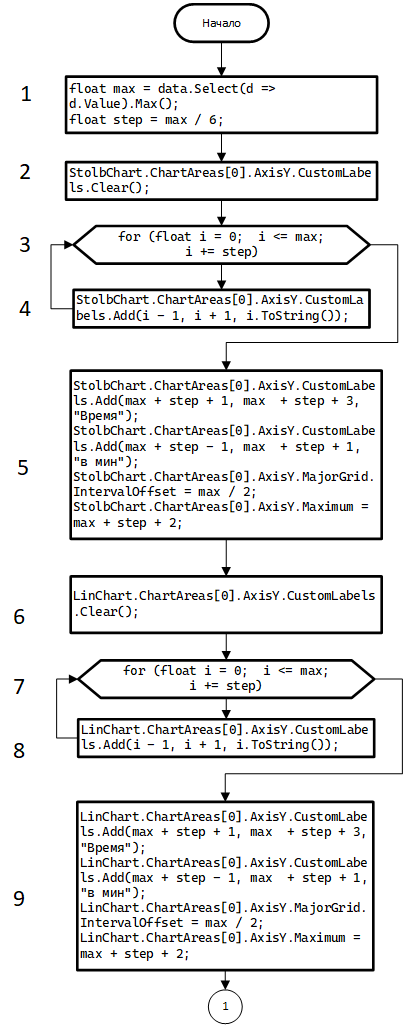


Рисунок 4 – Обновление диаграмм, часть 1

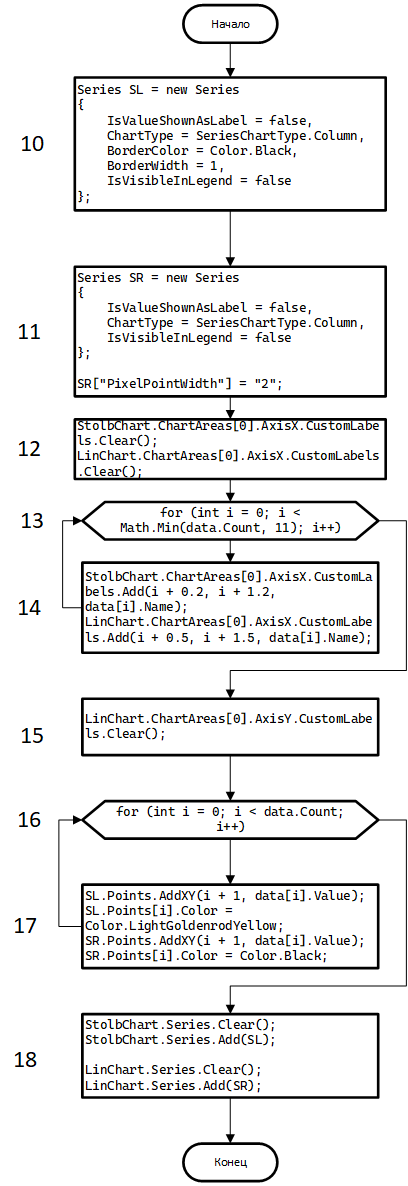


Рисунок 5 – Обновление диаграмм, часть 2

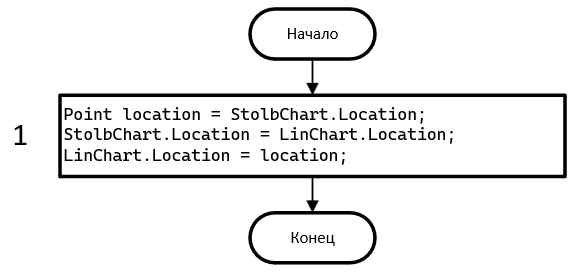


Рисунок 6 – Перемещение диаграмм

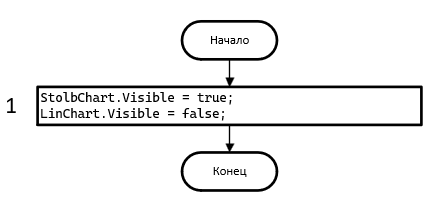


Рисунок 7 – Отображение только столбчатой диаграммы

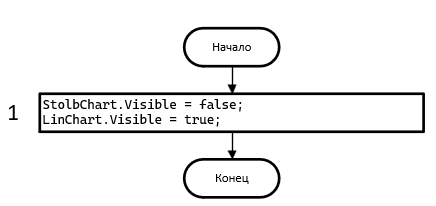


Рисунок 8 – Отображение только линейной диаграммы

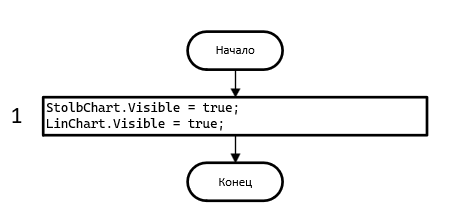


Рисунок 9 – Отображение только обеих диаграмм

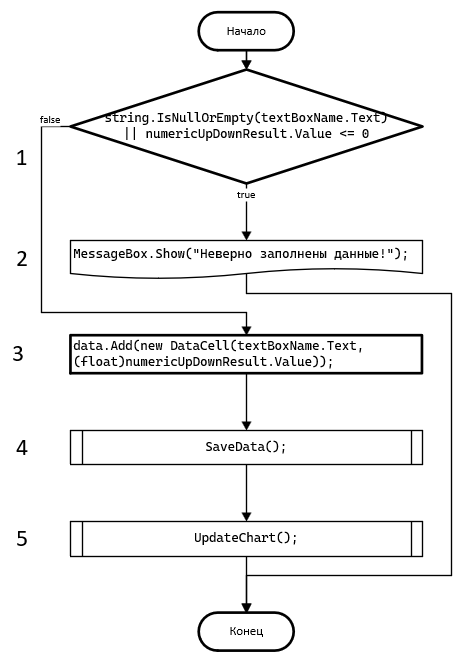


Рисунок 10 – Добавление записи

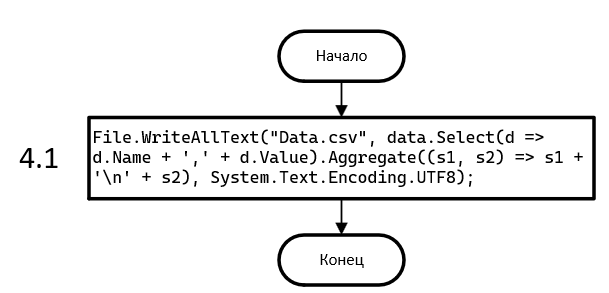


Рисунок 11 – Метод SaveData

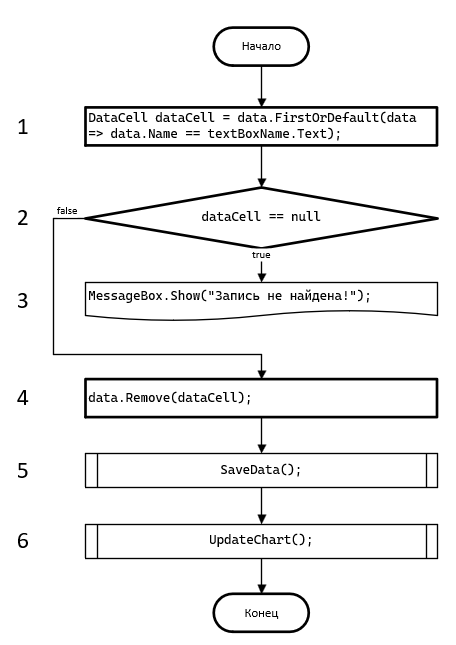


Рисунок 12 – Удаление записи

1. **Подбор тестовых примеров**

В качестве тестовых данных данные из диаграмм приведены к формату csv:

Файл Data.csv содержит следующий текст:

Таня,10

Саша,25

Оля,15

Игорь,30

Миша,15

**Листинг (код) составленного программного обеспечения**

**Файл DiagramForm.cs**

using System;

using System.Collections.Generic;

using System.Data;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Windows.Forms;

using System.Windows.Forms.DataVisualization.Charting;

namespace DiagrammApp

{

public partial class DiagramForm : Form

{

private List<DataCell> data = new List<DataCell>();

public DiagramForm()

{

InitializeComponent();

data = File.ReadAllText("Data.csv", System.Text.Encoding.UTF8).Split('\n').Select(s => new DataCell(s.Split(',')[0], float.Parse(s.Split(',')[1]))).ToList();

UpdateChart();

}

private void UpdateChart()

{

float max = data.Select(d => d.Value).Max();

float step = max / 6;

StolbChart.ChartAreas[0].AxisY.CustomLabels.Clear();

for (float i = 0; i <= max; i += step)

StolbChart.ChartAreas[0].AxisY.CustomLabels.Add(i - 1, i + 1, i.ToString());

StolbChart.ChartAreas[0].AxisY.CustomLabels.Add(max + step + 1, max + step + 3, "Время");

StolbChart.ChartAreas[0].AxisY.CustomLabels.Add(max + step - 1, max + step + 1, "в мин");

StolbChart.ChartAreas[0].AxisY.MajorGrid.IntervalOffset = max / 2;

StolbChart.ChartAreas[0].AxisY.Maximum = max + step + 2;

LinChart.ChartAreas[0].AxisY.CustomLabels.Clear();

for (float i = 0; i <= max; i += step)

LinChart.ChartAreas[0].AxisY.CustomLabels.Add(i - 1, i + 1, i.ToString());

LinChart.ChartAreas[0].AxisY.CustomLabels.Add(max + step + 1, max + step + 3, "Время");

LinChart.ChartAreas[0].AxisY.CustomLabels.Add(max + step - 1, max + step + 1, "в мин");

LinChart.ChartAreas[0].AxisY.MajorGrid.IntervalOffset = max / 2;

LinChart.ChartAreas[0].AxisY.Maximum = max + step + 2;

Series SL = new Series

{

IsValueShownAsLabel = false,

ChartType = SeriesChartType.Column,

BorderColor = Color.Black,

BorderWidth = 1,

IsVisibleInLegend = false

};

Series SR = new Series

{

IsValueShownAsLabel = false,

ChartType = SeriesChartType.Column,

IsVisibleInLegend = false

};

SR["PixelPointWidth"] = "2";

StolbChart.ChartAreas[0].AxisX.CustomLabels.Clear();

LinChart.ChartAreas[0].AxisX.CustomLabels.Clear();

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

{

StolbChart.ChartAreas[0].AxisX.CustomLabels.Add(i + 0.2, i + 1.2, data[i].Name);

LinChart.ChartAreas[0].AxisX.CustomLabels.Add(i + 0.5, i + 1.5, data[i].Name);

}

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

{

SL.Points.AddXY(i + 1, data[i].Value);

SL.Points[i].Color = Color.LightGoldenrodYellow;

SR.Points.AddXY(i + 1, data[i].Value);

SR.Points[i].Color = Color.Black;

}

StolbChart.Series.Clear();

StolbChart.Series.Add(SL);

LinChart.Series.Clear();

LinChart.Series.Add(SR);

}

private void SaveData()

{

File.WriteAllText("Data.csv", data.Select(d => d.Name + ',' + d.Value).Aggregate((s1, s2) => s1 + '\n' + s2), System.Text.Encoding.UTF8);

}

private void RBStolb\_CheckedChanged(object sender, EventArgs e)

{

StolbChart.Visible = true;

LinChart.Visible = false;

}

private void RBLin\_CheckedChanged(object sender, EventArgs e)

{

StolbChart.Visible = false;

LinChart.Visible = true;

}

private void RBBoth\_CheckedChanged(object sender, EventArgs e)

{

StolbChart.Visible = true;

LinChart.Visible = true;

}

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

{

Point location = StolbChart.Location;

StolbChart.Location = LinChart.Location;

LinChart.Location = location;

}

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

{

if (string.IsNullOrEmpty(textBoxName.Text) || numericUpDownResult.Value <= 0)

{

MessageBox.Show("Неверно заполнены данные!");

return;

}

data.Add(new DataCell(textBoxName.Text, (float)numericUpDownResult.Value));

SaveData();

UpdateChart();

}

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

{

DataCell dataCell = data.FirstOrDefault(data => data.Name == textBoxName.Text);

if (dataCell == null)

{

MessageBox.Show("Запись не найдена!");

return;

}

data.Remove(dataCell);

SaveData();

UpdateChart();

}

}

}

**Файл DataCell.cs**

namespace DiagrammApp

{

public class DataCell

{

public string Name { get; set; }

public float Value { get; set; }

public DataCell(string name, float value)

{

Name = name;

Value = value;

}

}

}

1. **Графический пользовательский интерфейс программного обеспечения и его описание**

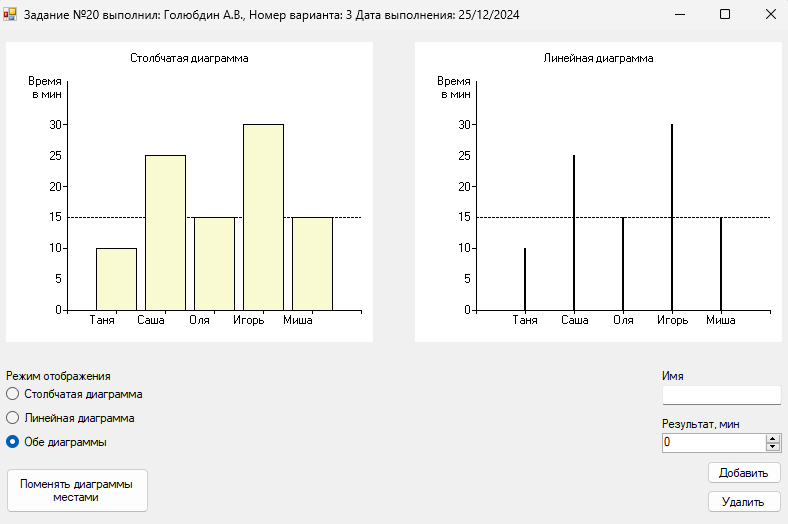
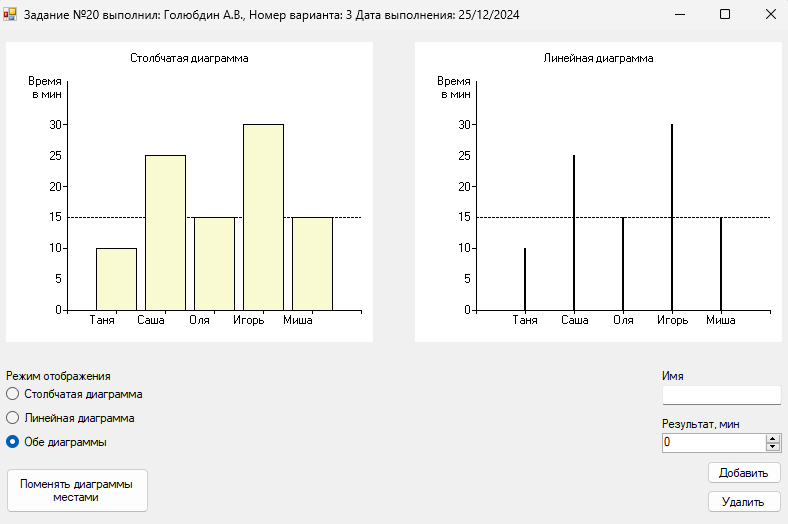


Рисунок 13 – Интерфейс главной формы

На форме находятся две диаграммы: столбчатая и линейная. Под левой диаграммой находятся переключатели для выбора режима отображения: с их помощью можно скрыть одну из диаграмм или показать обе. В левом нижнем углу находится кнопка для изменения положений диаграмм.

Под правой диаграммой находятся поля и кнопки для редактирования данных.

1. **Подтверждение соответствия графического пользовательского интерфейса требованиям к оформлению**

 Рисунок 14 – Подтверждение соответствия графического пользовательского интерфейса требованиям к оформлению.

На форме воспроизведено отображение диаграмм, представленное в требованиях.

С помощью переключателей и кнопки под левой диаграммой пользователю предоставлена обмена положением проиллюстрированных диаграмм, а также возможность отображения только одного из видов диаграмм.

1. **Расчёт тестовых примеров с использованием составленного программного обеспечения**

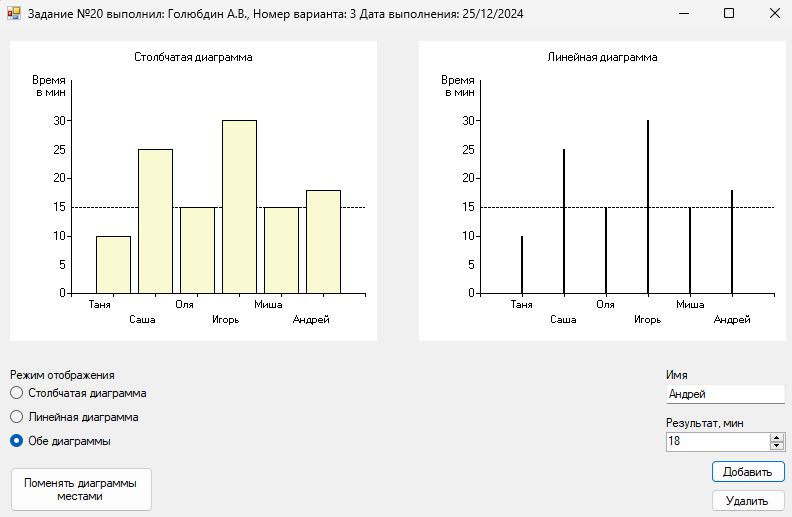


Рисунок 15 – Добавление записи

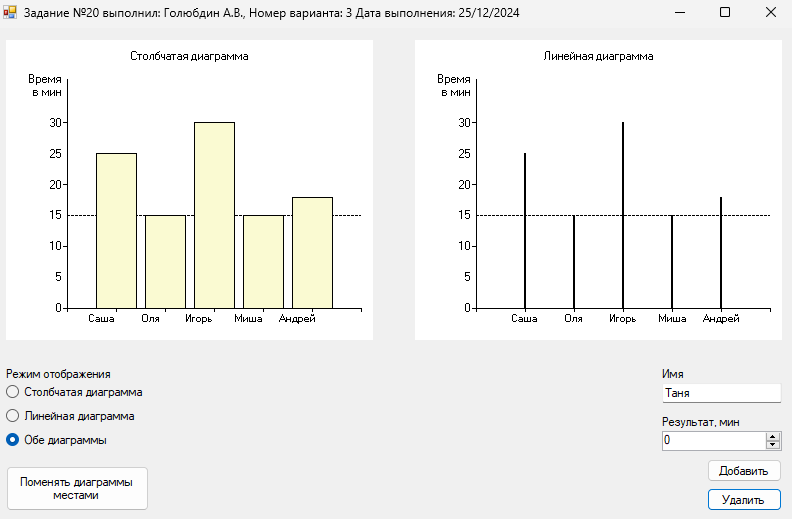


Рисунок 16 – Удаление записи

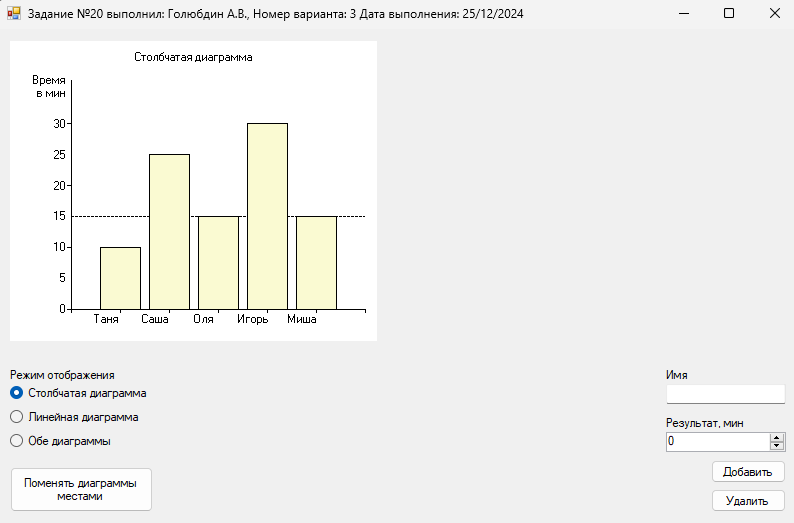


Рисунок 17 – Отображение только столбчатой диаграммы

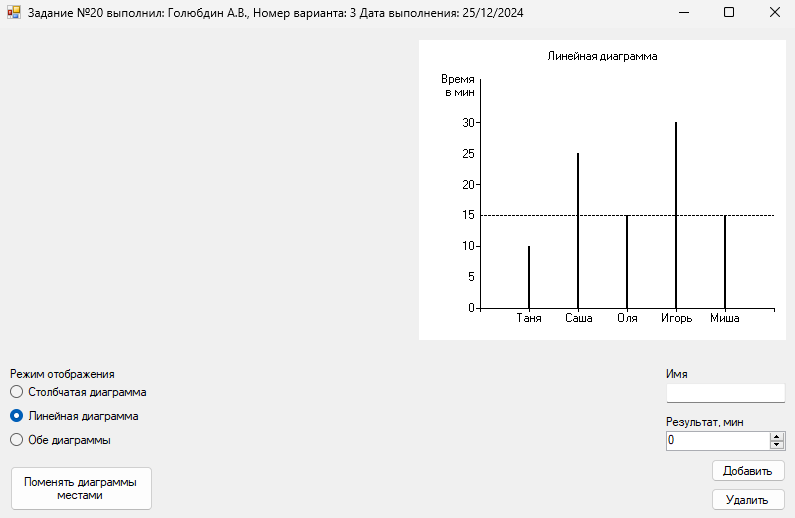


Рисунок 18 – Отображение только линейной диаграммы

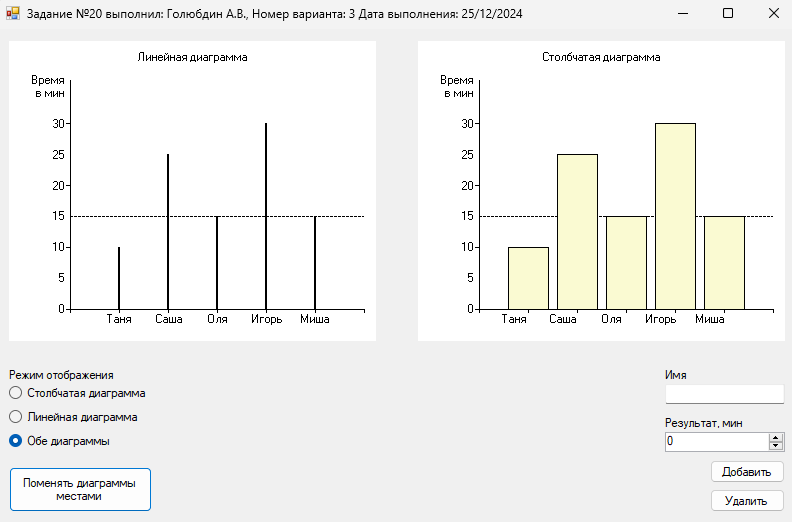


Рисунок 19 – Обмен диаграмм положением

1. **Формулировка вывода о проделанной работе**

Освоил навыки графического отображения статистических данных и построения диаграмм на платформе Windows Forms и применил их на практике.