

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Дисциплина: «Программирование»

Контрольное домашнее задание

Вариант 29

Выполнил: Рябичев Алексей,

Студент гр. БПИ176.

Преподаватель: Дударев В.А.

Дата: «__»_____ 2017г.

Подпись выполнившего: _____.

Москва 2017

Оглавление

Условие задачи	2
Функции разрабатываемого приложения	2
Варианты использования	2
Описание интерфейса пользователя	2
Структура приложения	3
Диаграмма классов	3
Описание классов, их полей и методов	3
Распределение исходного кода по файлам проекта	5
Контрольный пример и описание результатов	5
Текст (код) программы (только части, написанные вручную)	7
Список литературы	20

Условие задачи

Вариант 29.

Разработать оконное приложение (Windows Forms Application), которое будет представлять аналитический инструмент. Входными данными для программы должна служить таблица формата csv или excel (на выбор разработчика) с следующими столбцами: CT, BPAD, BPAN, BPAN2, BWAD, BWAN, BWAN2, BMAD, BMAN, BMAN2.

1. Основное окно программы должно поддерживать функции загрузки таблиц .csv формата и отображение значений этих таблиц (для отображения использовать DataGridView).
2. Возможность отображения зависимости между парами столбцов данных.
3. Каждый график создавать в отдельном окне, которое не блокирует основное.
4. Возможность открытия нескольких окон с графиками.
5. Возможность изменения границ графика при помощи TrackBar.
6. Обработка пропущенных значений

Функции разрабатываемого приложения

Варианты использования

Приложение используется для анализа таблиц .csv формата.

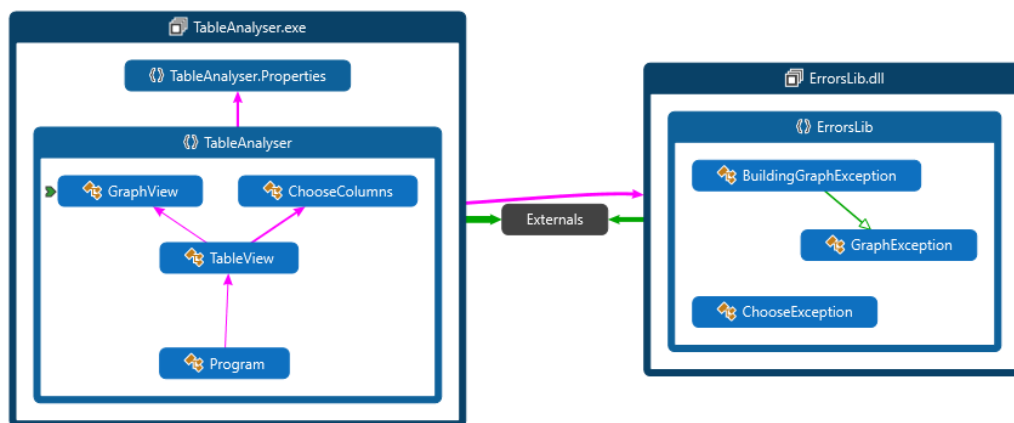
Описание интерфейса пользователя

Приложение позволяет открыть файл формата .csv и просмотреть его содержимое.

Приложение позволяет проанализировать (построить график) зависимость между парами столбцов и экспортировать полученные данные в .png формате.

Структура приложения

Диаграмма классов



Описание классов, их полей и методов

1. Класс Program – главный класс программы. Вызывает TableView
2. Класс TableView – класс, описывающий форму отображения таблицы
 - 2.1. _dataTable – таблица данных
 - 2.2. _dataGridView – элемент отображения таблицы
 - 2.3. toolTip – элемент отображения всплывающей подсказки
 - 2.4. _fileReady – переменная, обозначающая открыт ли уже файл
 - 2.5. _openFileDialog – диалог для выбора файла
 - 2.6. menuStrip – меню
 - 2.7. buildGraphToolStripMenuItem – элемент меню
 - 2.8. openFileToolStripMenuItem – элемент меню
 - 2.9. ShowTable – отображение таблицы
 - 2.10. _dataGridView_MouseHover – отображение подсказки для таблицы
 - 2.11. SmartSplit – деление строки по запятой и возвращение object[]
 - 2.12. BuildGraphToolStripMenuItem_Click – вызывает BuildGraph
 - 2.13. OpenFileToolStripMenuItem_Click – вызывает OpenFile
 - 2.14. OpenFileToolStripMenuItem_MouseHover – отображение подсказки для меню
 - 2.15. OpenFile – дает возможность выбрать файл
 - 2.16. FillTable – заолняет таблицу данными из файла
 - 2.17. TableView – конструктор данного класса
 - 2.18. TableView_Load – ладер данного класса
3. Класс ChooseColumns – класс, описывающий форму выбора столбцов для анализа

- 3.1. FirstColumn – название первого столбца
- 3.2. SecondColumn – название второго столбца
- 3.3. toolTip - элемент отображения всплывающей подсказки
- 3.4. button – кнопка «ОК»
- 3.5. checkedListBox – элемент отображения списка названий столбцов
- 3.6. _tableHeaders - список названий столбцов
- 3.7. _dataTable – таблица данных
- 3.8. Button_MouseHover – отображение подсказки для кнопки
- 3.9. CheckedListBox_MouseHover - отображение подсказки для списка
- 3.10. Button_Click – обработка нажатия кнопки «ОК»
- 3.11. ChooseColumns – конструктор данного класса
- 3.12. ChooseColumns_Load – лодер данного класса
- 4. Класс GraphView – класс, описывающий форму отображения графика
 - 4.1. changeColorToolStripMenuItem – элемент меню для смены цвета графика
 - 4.2. labelMaximumValue – текстовый элемент
 - 4.3. labelMinimumValue – текстовый элемент
 - 4.4. exportToolStripMenuItem – элемент меню для экспорта графика
 - 4.5. menuStrip - меню
 - 4.6. _yXDictionary – словарь координат первого столбца от второго
 - 4.7. _xYDictionary - словарь координат второго столбца от первого
 - 4.8. _dataTable - таблица
 - 4.9. toolTip – элемент отображения всплывающей подсказки
 - 4.10. chart - график
 - 4.11. _xY – список значений координат второго столбца от первого
 - 4.12. _yX - список значений координат первого столбца от второго
 - 4.13. _flag – переменная обозначающая какая зависимость сейчас отображается
 - 4.14. _firstColumn – название первого столбца
 - 4.15. _secondColumn – название второго столбца
 - 4.16. trackBarMinimum – трэкбар для минимального значения
 - 4.17. trackBarMaximum – трэкбар для максимального значения
 - 4.18. exportToolStripMenuItem_MouseHover – отображение подсказки для меню
 - 4.19. trackBarMinimum_MouseHover – отображение подсказки для трэкбара
 - 4.20. trackBarMaximum_MouseHover – отображение подсказки для трэкбара
 - 4.21. Chart_MouseHover – отображение подсказки для графика
 - 4.22. trackBarMaximum_Scroll – изменение значения максимального значения
 - 4.23. trackBarMinimum_Scroll – изменение значения для минимального значения
 - 4.24. exportToolStripMenuItem_Click – экспорт графика
 - 4.25. BuildChart – отображение графика
 - 4.26. SortDictionaries – сортировка координат по возрастанию
 - 4.27. Chart_Click – смена осей x и y
 - 4.28. ParseTable – парсинг таблицы

Рябичев 29 вариант

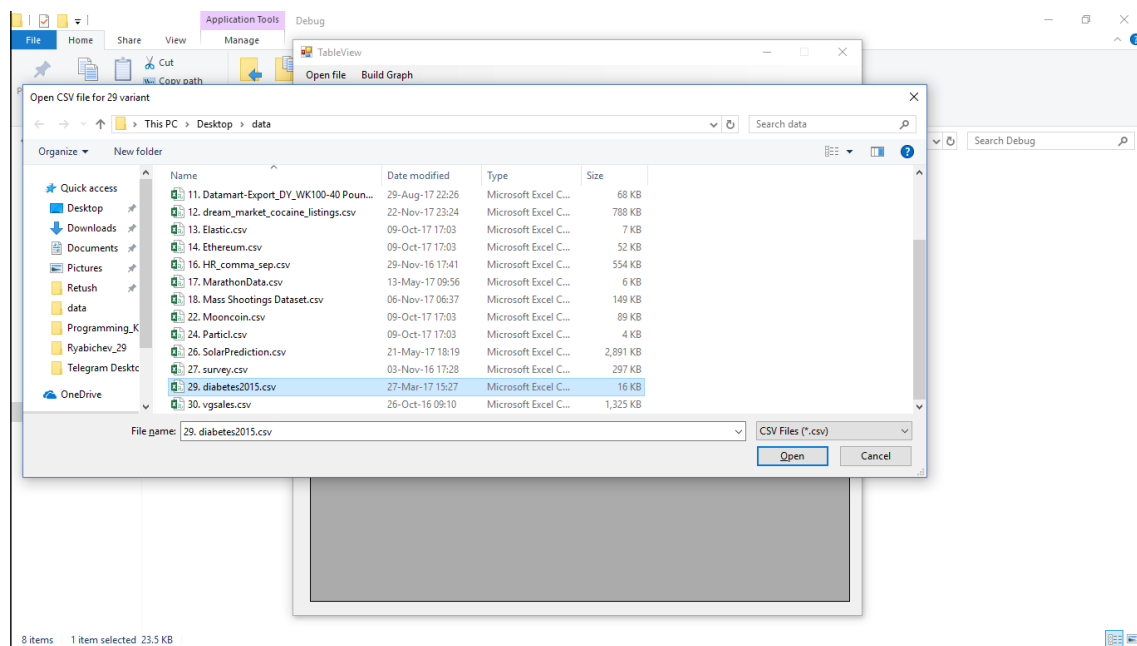
- 4.29. SetTrackBarsMinMax – настройка минимальных и максимальных значений трэкбаров
- 4.30. GraphView – конструктор данного класса
- 5. ErrorsLib – библиотека классов пользовательских ошибок (Exception)
 - 5.1. ChooseException - наследник класса Exception
 - 5.1.1. ChooseException – конструктор от одной переменной
 - 5.2. GraphException - наследник класса Exception
 - 5.2.1. GraphException – конструктор от двух переменных
 - 5.2.2. GraphException – конструктор от одной переменной
 - 5.3. BuildingGraphException - наследник класса GraphException
 - 5.3.1. BuildingGraphException – конструктор от одной переменной

Распределение исходного кода программы по файлам проекта

Классы ChooseException, GraphException, BuildingGraphException находятся в файлах ChooseException.cs, GraphException.cs, BuildingGraphException.cs соответственно и в библиотеке файлов ErrorsLib, которая находится в файле ErrorsLib.dll

Классы GraphView, ChooseColumns, TableView, Program находятся в файлах GraphView.cs, ChooseColumns.cs, TableView.cs, Program.cs соответственно и в проекте TableAnalyser, который находится в файле TableAnalyser.exe

Контрольный пример и описание результатов



Выбор файла

Рябичев 29 вариант

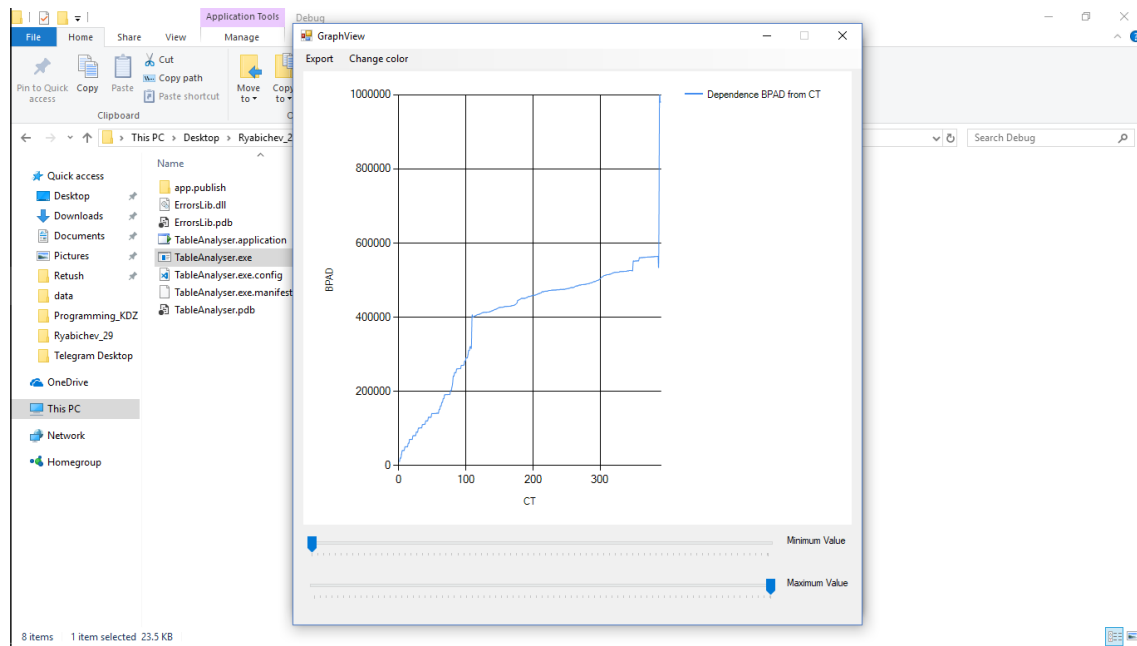
CT	BPAD	BPAN	BPAN2	BWAD	BWAN	BWAN2	BMAD
100500	1301	198	113	750	115	58	551
101100	1170	112	70	653	65	38	517
101400	1960	202	121	1081	106	61	879
101600	633	52	33	370	35	23	263
101700	844	87	58	501	52	38	343
101800	1589	195	119	894	108	72	695
10300	614	38	34	276	21	19	338
110200	2160	160	95	1194	78	47	966
110600	1576	98	64	869	58	36	707
111300	1339	109	69	784	61	35	555
111400	749	55	36	406	41	26	343
111500	1764	204	124	1056	137	82	708
120300	894	95	63	544	62	40	350
120400	441	46	27	254	27	16	187
120700	417	53	30	242	37	17	175
120800	396	51	24	236	37	17	160
130100	1148	109	65	712	68	34	436
130200	662	86	52	413	59	37	249
130300	750	88	49	443	57	29	307
130400	631	70	40	375	43	26	256
130600	2040	200	120	1238	125	72	802
140100	1490	86	44	777	37	14	713
140200	1011	45	22	517	20	12	494
140300	2072	133	60	1097	72	30	975
140400	1572	68	40	813	33	19	759
140500	992	99	56	562	53	28	430
140600	1959	126	70	1059	71	33	900
140800	2887	175	92	1491	80	41	1396
141000	608	43	22	337	24	11	271

Отображение таблицы

CT	BPAD	BPAN	BPAN2	BWAD	BWAN	BWAN2	BMAD
100500	1301	198	113	750	115	58	551
101100	1170	112	70	653	65	38	517
101400	1960	202	121	1081	106	61	879
101600	633	52	33	370	35	23	263
101700	844	87	58	501	52	38	343
101800	1589	195	119	894	108	72	695
10300	614	38	34	276	21	19	338
110200	2160	160	95	1194	78	47	966
110600	1576	98	64	869	58	36	707
111300	1339	109	69	784	61	35	555
111400	749	55	36	406	41	26	343
111500	1764	204	124	1056	137	82	708
120300	894	95	63	544	62	40	350
120400	441	46	27	254	27	16	187
120700	417	53	30	242	37	17	175
120800	396	51	24	236	37	17	160
130100	1148	109	65	712	68	34	436
130200	662	86	52	413	59	37	249
130300	750	88	49	443	57	29	307
130400	631	70	40	375	43	26	256
130600	2040	200	120	1238	125	72	802
140100	1490	86	44	777	37	14	713
140200	1011	45	22	517	20	12	494
140300	2072	133	60	1097	72	30	975
140400	1572	68	40	813	33	19	759
140500	992	99	56	562	53	28	430
140600	1959	126	70	1059	71	33	900
140800	2887	175	92	1491	80	41	1396
141000	608	43	22	337	24	11	271

Выбор столбцов

Рябичев 29 вариант



Отображение графика

Текст (код) программы (только части, написанные вручную)

TableView.cs

```
using System;

using System.Data;

using System.Drawing;

using System.IO;

using System.Windows.Forms;

using ErrorsLib;

using TableAnalyser.Properties;

namespace TableAnalyser
{
    public sealed partial class TableView : Form
    {
        private readonly OpenFileDialog _openFileDialog = new OpenFileDialog
        {
            Filter = Resources._openFileDialog_Filter,
            Title = Resources._openFileDialog_Title
        };

        private DataTable _dataTable;

        private bool _fileReady;

        /// <summary>
```

Рябичев 29 вариант

```
///      Create form of TableView
/// </summary>

public TableView()
{
    InitializeComponent();

    StartPosition = FormStartPosition.CenterScreen;

    FormBorderStyle = FormBorderStyle.FixedSingle;

    Font = new Font("Bandera", 8);

    Load += TableView_Load;
}

/// <summary>
///      Loader for TableView
/// </summary>
/// <param name="sender">sender</param>
/// <param name="e">event arguments</param>
private void TableView_Load(object sender, EventArgs e)
{
    Show();

    OpenFile();
}

/// <summary>
///      Open file
/// </summary>
private void OpenFile()
{
    bool flag = false;

    while (!flag)

        try
        {
            DialogResult dialogResult = _openFileDialog.ShowDialog();

            if (dialogResult != DialogResult.OK && !_fileReady)

                throw new Exception("Please, choose file firstly");

            Cursor.Current = Cursors.WaitCursor;

            FillTable();

            Cursor.Current = Cursors.Default;

            Show();

            ShowTable();
        }
    }
}
```


Рябичев 29 вариант

```
        flag = true;

        _fileReady = true;
    }

    catch (Exception exception)
    {
        MessageBox.Show(exception.Message);
    }
}

/// <summary>
///     Fill table of elements
/// </summary>
private void FillTable()
{
    _dataTable = new DataTable();

    string[] tmp = File.ReadAllLines(_openFileDialog.FileName);

    for (int i = 0; i < tmp.Length; i++)
        if (i == 0)
            foreach (string s in tmp[i].Split(','))
                _dataTable.Columns.Add(s);
            else
                _dataTable.Rows.Add(_dataTable.NewRow().ItemArray
SmartSplit(tmp[i]));
    }

/// <summary>
///     Show table of elements
/// </summary>
private void ShowTable()
{
    _dataGridView.DataSource = _dataTable;
    _dataGridView.ColumnHeadersDefaultCellStyle.Font =
        new Font(FontFamily.GenericSansSerif, 10, FontStyle.Bold);
    _dataGridView.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.AllCells;
    _dataGridView.AutoSizeRowsMode = DataGridViewAutoSizeRowsMode.AllCells;
    _dataGridView.ScrollBars = ScrollBars.Both;
    _dataGridView.Anchor = AnchorStyles.Top;
}

/// <summary>
```

Рябичев 29 вариант

```
/// Clicker for open file menu
/// </summary>
/// <param name="sender">sender</param>
/// <param name="e">event arguments</param>
private void OpenFileToolStripMenuItem_Click(object sender, EventArgs e) =>
OpenFile();

/// <summary>
/// Clicker for build graph menu
/// </summary>
/// <param name="sender">sender</param>
/// <param name="e">event arguments</param>
private void BuildGraphToolStripMenuItem_Click(object sender, EventArgs e)
{
    ChooseColumns chooseColumns = new ChooseColumns(_dataTable);
    try
    {
        DialogResult dialogResult = chooseColumns.ShowDialog();
        if (dialogResult == DialogResult.OK)
        {
            Cursor.Current = Cursors.WaitCursor;
            GraphView graphView =
                new GraphView(_dataTable, chooseColumns.FirstColumn,
chooseColumns.SecondColumn);
            graphView.Show();
        }
        else if (dialogResult != DialogResult.Abort && dialogResult !=
DialogResult.Cancel)
        {
            throw new ChooseException();
        }
    }
    catch (ChooseException exception)
    {
        MessageBox.Show(Resources.ChooseExceptionMessage + exception.Message);
        BuildGraphToolStripMenuItem_Click(sender, e);
    }
    catch (GraphException exception)
    {
        if (exception.InnerException != null)
        MessageBox.Show(exception.InnerException.Message);
    }
    catch (Exception exception)
```

Рябичев 29 вариант

```
{
    MessageBox.Show(exception.Message);
}

}

/// <summary>
///     Split string by coma
/// </summary>
/// <param name="s">string</param>
/// <returns>object[]</returns>
private object[] SmartSplit(string s)
{
    if (s == null) throw new ArgumentNullException(nameof(s));
    object[] objects = new object[_dataTable.Columns.Count];
    for (int j = 0; j < s.Split(',').Length; j++)
        objects[j] = string.IsNullOrEmpty(s.Split(',')[j]) ? "0" :
s.Split(',')[j];
    return objects;
}

/// <summary>
/// Listener for mouse hover of table
/// </summary>
/// <param name="sender">sender</param>
/// <param name="e">event arguments</param>
private void _dataGridView_MouseHover(object sender, EventArgs e) =>
ToolTip.SetToolTip(_dataGridView, "Table of elements from file");

/// <summary>
/// Listener for mouse hover of menu
/// </summary>
/// <param name="sender">sender</param>
/// <param name="e">event arguments</param>
private void OpenFileToolStripMenuItem_MouseHover(object sender, EventArgs e) =>
ToolTip.SetToolTip(menuStrip, "Menu");
}
}
```

GraphView.cs

```
using System;
using System.Collections.Generic;
using System.Data;
```

Рябичев 29 вариант

```
using System.Linq;

using System.Windows.Forms;

using System.Windows.Forms.DataVisualization.Charting;

using ErrorsLib;

using TableAnalyser.Properties;

using Cursor = System.Windows.Forms.Cursor;

namespace TableAnalyser
{
    public partial class GraphView : Form
    {
        private readonly DataTable _dataTable;
        private readonly string _firstColumn;
        private readonly string _secondColumn;
        private readonly Dictionary<int, int> _xYDictionary = new Dictionary<int, int>();
        private readonly Dictionary<int, int> _yXDictionary = new Dictionary<int, int>();
        private bool _flag;
        private List<KeyValuePair<int, int>> _xY = new List<KeyValuePair<int, int>>();
        private List<KeyValuePair<int, int>> _yX = new List<KeyValuePair<int, int>>();

        /// <summary>
        ///     Creates GraphView Form
        /// </summary>
        /// <param name="dataTable">Table of elements</param>
        /// <param name="firstColumn">Name of first column</param>
        /// <param name="secondColumn">Name of second column</param>
        public GraphView(DataTable dataTable, string firstColumn, string secondColumn)
        {
            InitializeComponent();

            _dataTable = dataTable;
            _firstColumn = firstColumn;
            _secondColumn = secondColumn;

            FormBorderStyle = FormBorderStyle.FixedSingle;

            _flag = false;

            ParseTable();

            SortDictionaries();

            Cursor.Current = Cursors.Default;

            BuildChart(_xY, _firstColumn, _secondColumn);

            MessageBox.Show(Resources.GraphViewChartClickHint);
        }
    }
}
```

Рябичев 29 вариант

```
/// <summary>
///     Method for revert chart
/// </summary>
/// <param name="sender">sender</param>
/// <param name="e">event arguments</param>
private void Chart_Click(object sender, EventArgs e)
{
    _flag = !_flag;
    if (_flag)
        BuildChart(_xY, _firstColumn, _secondColumn);
    else
        BuildChart(_yX, _secondColumn, _firstColumn);
}

/// <summary>
///     Build chart
/// </summary>
/// <param name="xy">Dicitionary of coordinates</param>
/// <param name="ox">Name for Ox</param>
/// <param name="oy">Name for Oy</param>
private void BuildChart(List<KeyValuePair<int, int>> xy, string ox, string oy)
{
    chart.Series[0].Points.Clear();
    chart.Series[0].Name = "Graph";
    chart.Series[0].LegendText = "Dependence " + oy + " from " + ox;
    chart.ChartAreas[0].AxisX.Title = ox;
    chart.ChartAreas[0].AxisY.Title = oy;
    chart.Series[0].ChartType = SeriesChartType.Spline;

    foreach (KeyValuePair<int, int> pair in xy)
        chart.Series[0].Points.Add(pair.Key, pair.Value);

    trackBarMinimum.Minimum = (int) chart.ChartAreas[0].AxisX.Minimum;
    trackBarMinimum.Maximum = (int) chart.ChartAreas[0].AxisX.Maximum;
    trackBarMaximum.Minimum = (int) chart.ChartAreas[0].AxisX.Minimum;
    trackBarMaximum.Maximum = (int) chart.ChartAreas[0].AxisX.Maximum;
    trackBarMaximum.TickFrequency = 5;
    trackBarMinimum.TickFrequency = 5;
    trackBarMaximum.Value = (int) chart.ChartAreas[0].AxisX.Maximum;
    trackBarMinimum.Value = (int) chart.ChartAreas[0].AxisX.Minimum;
}
```

Рябичев 29 вариант

```
private void SetTrackbarsMinMax()
{
    trackBarMaximum.Minimum = trackBarMinimum.Value;
    trackBarMinimum.Maximum = trackBarMaximum.Value;
}

/// <summary>
///     Parsing table of elements
/// </summary>
private void ParseTable()
{
    foreach (DataRow row in _dataTable.Rows)
    {
        try
        {
            if (!_xYDictionary.ContainsKey(
                int.Parse(row[_firstColumn] as string ?? throw new
BuildingGraphException()))))
                _xYDictionary[int.Parse(row[_firstColumn] as string)] =
                (_xYDictionary[int.Parse(row[_firstColumn] as string)] +
                (row[_secondColumn] == null
                    ? 0
                    : int.Parse(row[_secondColumn] as string ?? throw new
BuildingGraphException())))) / 2;
            else
                _xYDictionary.Add(
                    row[_firstColumn] == null
                        ? 0
                        : int.Parse(row[_firstColumn] as string ?? throw new
BuildingGraphException()),
                    row[_secondColumn] == null
                        ? 0
                        : int.Parse(row[_secondColumn] as string ?? throw new
BuildingGraphException()));
            if (!_yXDictionary.ContainsKey(
                int.Parse(row[_secondColumn] as string ?? throw new
BuildingGraphException()))))
                _yXDictionary[int.Parse(row[_secondColumn] as string)] =
                (_yXDictionary[int.Parse(row[_secondColumn] as string)] +
                (row[_firstColumn] == null
                    ? 0
                    : int.Parse(row[_firstColumn] as string ?? throw new
BuildingGraphException())))) / 2;
            else
                _yXDictionary.Add(
```

Рябичев 29 вариант

```
        row[_secondColumn] == null
            ? 0
            : int.Parse(row[_secondColumn] as string ?? throw new
BuildingGraphException()),
        row[_firstColumn] == null
            ? 0
            : int.Parse(row[_firstColumn] as string ?? throw new
BuildingGraphException()));
    }
    catch (Exception exception)
    {
        throw new GraphException(exception);
    }
}

/// <summary>
///     Listener for maximum trackbar scrolling
/// </summary>
/// <param name="sender">sender</param>
/// <param name="e">event arguments</param>
private void TrackBarMaximum_Scroll(object sender, EventArgs e)
{
    SetTrackbarsMinMax();
    chart.ChartAreas[0].AxisX.Maximum = trackBarMaximum.Value;
}

/// <summary>
///     Listener for minimum trackbar scrolling
/// </summary>
/// <param name="sender">sender</param>
/// <param name="e">event arguments</param>
private void TrackBarMinimum_Scroll(object sender, EventArgs e)
{
    SetTrackbarsMinMax();
    chart.ChartAreas[0].AxisX.Minimum = trackBarMinimum.Value;
}

/// <summary>
///     Cast dicitonaries to lists and sort them
/// </summary>
private void SortDictionaries()
{

```

Рябичев 29 вариант

```
_xY = _xYDictionary.ToList();
_xY.Sort((pair1, pair2) => pair1.Key.CompareTo(pair2.Key));

_yX = _yXDictionary.ToList();
_yX.Sort((pair1, pair2) => pair1.Key.CompareTo(pair2.Key));
}

/// <summary>
///     Change color of chart
/// </summary>
/// <param name="sender">sender</param>
/// <param name="e">event arguments</param>
public void ChangeColorToolStripMenuItem_Click(object sender, EventArgs e)
{
    ColorDialog colorDialog = new ColorDialog();
    if (colorDialog.ShowDialog() == DialogResult.OK)
        chart.Series[0].Color = colorDialog.Color;
}

/// <summary>
///     Export chart as png
/// </summary>
/// <param name="sender">sender</param>
/// <param name="e">event arguments</param>
private void ExportToolStripMenuItem_Click(object sender, EventArgs e)
{
    FolderBrowserDialog folderBrowserDialog = new FolderBrowserDialog();
    if (folderBrowserDialog.ShowDialog() != DialogResult.OK) return;
    if (_flag)
        chart.SaveImage(
            folderBrowserDialog.SelectedPath + @"\" + "Dependence" + _secondColumn
+ "from" + _firstColumn +
            ".png", ChartImageFormat.Png);
    else
        chart.SaveImage(
            folderBrowserDialog.SelectedPath + @"\" + "Dependence" + _firstColumn
+ "from" + _secondColumn+
            ".png", ChartImageFormat.Png);
    MessageBox.Show(Resources.FileSave + folderBrowserDialog.SelectedPath);
}

/// <summary>
```


Рябичев 29 вариант

```
///      Listener for mouse hover of chart
/// </summary>
/// <param name="sender">sender</param>
/// <param name="e">event arguments</param>
private void Chart_MouseHover(object sender, EventArgs e) =>
toolTip.SetToolTip(chart, "Chart");

/// <summary>
///      Listener for mouse hover of menu
/// </summary>
/// <param name="sender">sender</param>
/// <param name="e">event arguments</param>
private void ExportToolStripMenuItem_MouseHover(object sender, EventArgs e) =>
toolTip.SetToolTip(menuStrip, "Menu");

/// <summary>
///      Listener for mouse hover of maximum trackbar
/// </summary>
/// <param name="sender">sender</param>
/// <param name="e">event arguments</param>
private void TrackBarMaximum_MouseHover(object sender, EventArgs e) =>
toolTip.SetToolTip(trackBarMaximum, "Maximum trackbar");

/// <summary>
///      Listener for mouse hover of minimum trackbar
/// </summary>
/// <param name="sender">sender</param>
/// <param name="e">event arguments</param>
private void TrackBarMinimum_MouseHover(object sender, EventArgs e) =>
toolTip.SetToolTip(trackBarMinimum, "Minimum trackbar");
}
}
```

ChooseColumns.cs

```
using System;
using System.Data;
using System.Drawing;
using System.Windows.Forms;
using ErrorsLib;

namespace TableAnalyser
{
    public partial class ChooseColumns : Form
```

Рябичев 29 вариант

```
{

    private readonly DataTable _dataTable;

    private string[] _tableHeaders;

    /// <summary>
    ///     Create form of ChooseColumn
    /// </summary>
    /// <param name="dataTable"></param>
    public ChooseColumns(DataTable dataTable)
    {
        InitializeComponent();

        Size = new Size(400, 400);

        StartPosition = FormStartPosition.CenterScreen;

        FormBorderStyle = FormBorderStyle.FixedSingle;

        _dataTable = dataTable;

        Load += ChooseColumns_Load;
    }

    public string FirstColumn { get; private set; }

    public string SecondColumn { get; private set; }

    /// <summary>
    ///     Loader for ChooseColumn
    /// </summary>
    /// <param name="sender">sender</param>
    /// <param name="e">event arguments</param>
    private void ChooseColumns_Load(object sender, EventArgs e)
    {
        _tableHeaders = new string[_dataTable.Columns.Count];
        for (int i = 0; i < _tableHeaders.Length; i++)
        {
            _tableHeaders[i] = _dataTable.Columns[i].ColumnName;
            checkedListBox.Items.Add(_tableHeaders[i]);
        }
    }

    /// <summary>
    ///     Clicker for button
    /// </summary>
    /// <param name="sender">sender</param>
```

Рябичев 29 вариант

```
/// <param name="e">event arguments</param>

private void Button_Click(object sender, EventArgs e)

{

    if (checkedListBox.CheckedIndices.Count != 2)

        throw new ChooseException("Choose only two columns");

    FirstColumn = _tableHeaders[checkedListBox.CheckedIndices[0]];

    SecondColumn = _tableHeaders[checkedListBox.CheckedIndices[1]];

    DialogResult = DialogResult.OK;

    Close();

}

/// <summary>

///     Listener for mouse hover of listbox

/// </summary>

/// <param name="sender">sender</param>

/// <param name="e">event arguments</param>

private void CheckedListBox_MouseHover(object sender, EventArgs e) =>
tooltip.SetToolTip(checkedListBox, "List of column names");

/// <summary>

///     Listener for mouse hover of button

/// </summary>

/// <param name="sender">sender</param>

/// <param name="e">event arguments</param>

private void Button_MouseHover(object sender, EventArgs e) =>
tooltip.SetToolTip(button, "OK button");

}

}
```

BuildingGraphException.cs

```
using System;

namespace ErrorsLib

{

    public class BuildingGraphException : GraphException

    {

        public BuildingGraphException(string message = "Some of elements are null") :
        base(message) { }

    }

}
```

GraphException.cs

```
using System;
```

Рябичев 29 вариант

```
namespace ErrorsLib
{
    public class GraphException : Exception
    {
        public GraphException(Exception exception) : base(null, exception) { }

        public GraphException(string message) : base(message) { }
    }
}
```

ChooseException.cs

```
using System;

namespace ErrorsLib
{
    public class ChooseException : Exception
    {
        public ChooseException(string message = "Choose columns firstly") : base(message)
        { }
    }
}
```

Список литературы

1. MSDN Library: <https://msdn.microsoft.com/en-us/library/ms310241>