

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего
образования



НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий

Кафедра Информационная безопасность вычислительных систем и сетей

Проектирование пользовательского интерфейса
(наименование работы)

ОТЧЕТ

по лабораторной работе №6

по дисциплине

Технологии программирования
(наименование дисциплины)

РУКОВОДИТЕЛЬ:

(подпись)

Капранов С.Н.
(фамилия, и., о.)

СТУДЕНТ:

(подпись)

Савельев М.А.
(фамилия, и., о.)

18-ИСТ-4
(шифр группы)

Работа защищена «__» _____

С оценкой _____

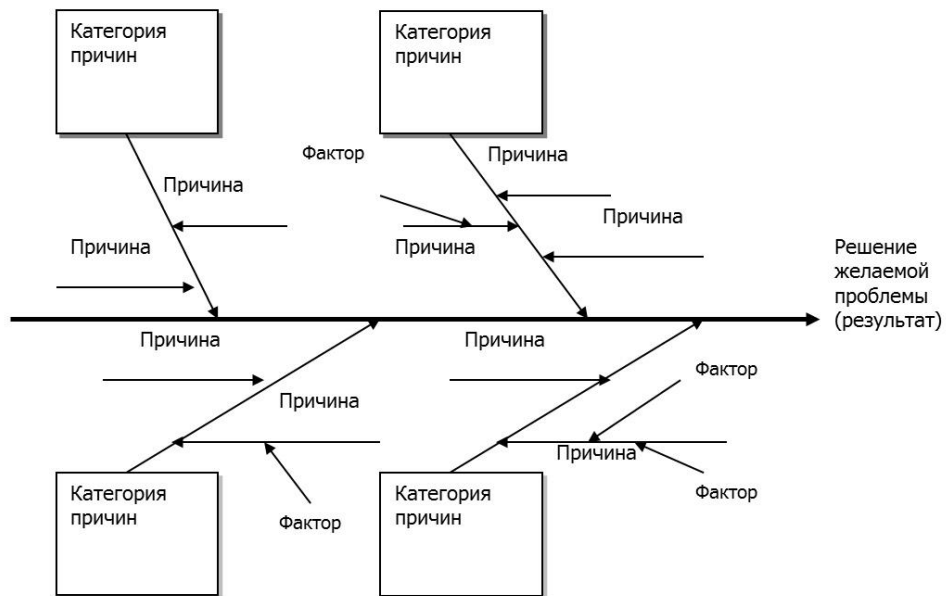
Нижний Новгород, 2020

Содержание

Задача	3
Основная часть отчета	4
Листинг программы	4
Скриншоты работы	17

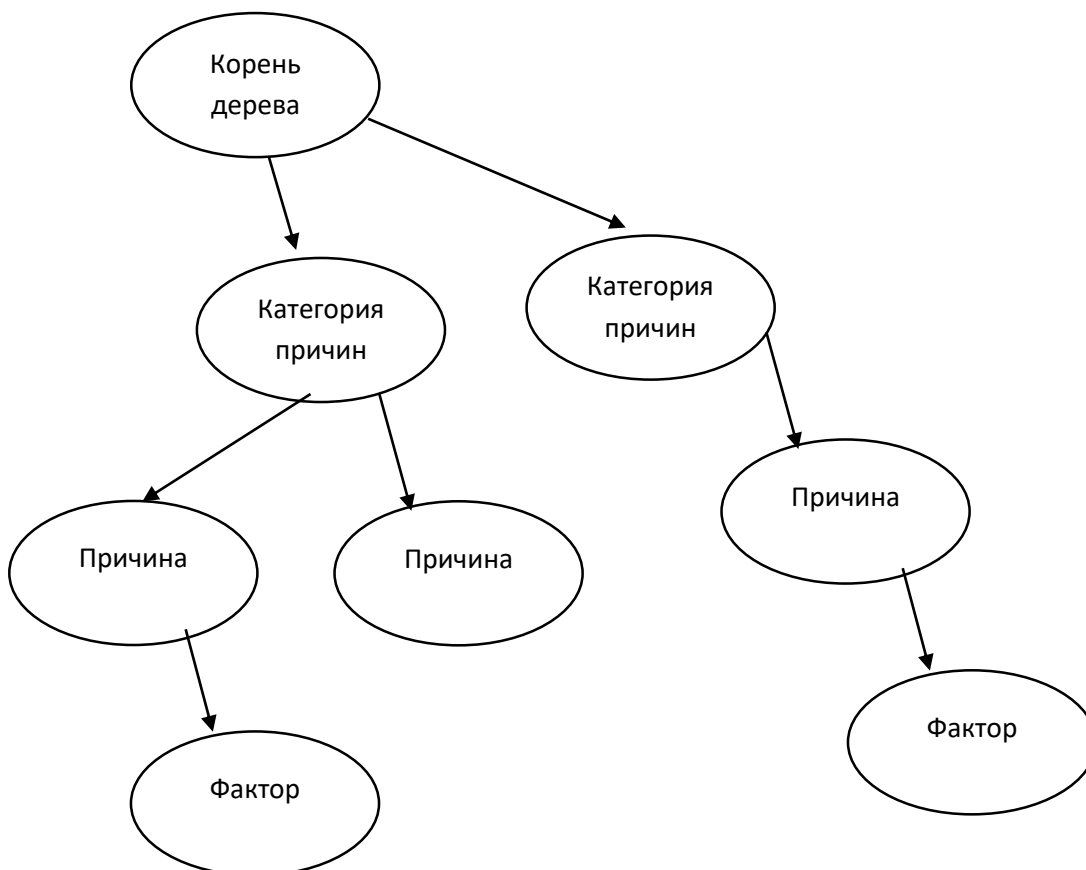
Задача

4 вариант: Графы причинно-следственных связей (диаграмма Исикавы). Как образец была взята диаграмма Исикавы (Рис. 1).



В последствии была представлена в виде дерева и выведена на экран.

Результат (корень дерева) → Категория причин (1 поколение) → Причина (2 поколение) → Фактор (3 поколение)



Основная часть отчета

Программа написана на языке C# в среде разработки VisualStudio 2017.

Листинг программы

```
using System;
using System.Windows.Forms;
using Excel = Microsoft.Office.Interop.Excel;
using System.Data.OleDb;
using System.Data;

namespace Laba6
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            private void addResultButtonClick(object sender, EventArgs e)
            {
                resultField.Enabled = false;
                addResultButton.Enabled = false;
                newResultButton.Enabled = true;
                catOfReasonField.Enabled = true;
                treeView1.Nodes.Add(new TreeNode(resultField.Text));
                treeView1.ExpandAll();
            }

            private void newResultButtonClick(object sender, EventArgs e)
            {
                DialogResult result = MessageBox.Show("Все введенные данные будут удалены." + "\r\n" +
                "Вы уверена, что хотите продолжить?", "Начать работу заново", MessageBoxButtons.YesNo);
                if (result == DialogResult.Yes)
                {
                    catOfReasonField.Enabled = false;
                    catOfReasonField.Clear();
                    reasonField.Enabled = false;
                    reasonField.Clear();
                    factorField.Enabled = false;
                    factorField.Clear();
                    resultField.Enabled = true;
                    resultField.Clear();
                    addResultButton.Enabled = false;
                    addCatOfReasonButton.Enabled = false;
                    newCatOfReasonButton.Enabled = false;
                    removeCatOfReasonButton.Enabled = false;
                    addReasonButton.Enabled = false;
                }
            }
        }
    }
}
```

```

        newReasonButton.Enabled = false;
        removeReasonButton.Enabled = false;
        addFactorButton.Enabled = false;
        removeFactorButton.Enabled = false;
        dataGridView1.Columns.Clear();
        dataGridView2.Columns.Clear();
        treeView1.Nodes.Clear();
        catOfReasonList.Items.Clear();
        addReasonList.Items.Clear();
        addFactorList.Items.Clear();
        reasonList.Items.Clear();
        rootField.Enabled = false;
        addRootButton.Enabled = false;
        rootField.Clear();
        count_of_cat_of_reason = 1;
        MessageBox.Show("Все данные были удалены!");
    }
}

private void troubleButtonEnabled(object sender, EventArgs e)
{
    if (resultField.Text == "")
    {
        addResultButton.Enabled = false;
    }
    else addResultButton.Enabled = true;
}

private void catOfReasonButtonEnabled(object sender, EventArgs e)
{
    if (catOfReasonField.Text == "")
    {
        addCatOfReasonButton.Enabled = false;
    }
    else addCatOfReasonButton.Enabled = true;
}

int count_of_cat_of_reason = 1;
private void addCatOfReasonButtonClick(object sender, EventArgs e)
{
    dataGridView1.Columns.Add("#" + count_of_cat_of_reason + ": " + catOfReasonField.Text, "#" +
count_of_cat_of_reason + ": " + catOfReasonField.Text);
    catOfReasonList.Items.Add("#" + count_of_cat_of_reason + ": " + catOfReasonField.Text);
    addReasonList.Items.Add("#" + count_of_cat_of_reason + ": " + catOfReasonField.Text);
    count_of_cat_of_reason++;
    treeView1.Nodes[0].Nodes.Add(new TreeNode(catOfReasonField.Text));
    treeView1.ExpandAll();
    addCatOfReasonButton.Enabled = false;
    catOfReasonField.Enabled = false;
}

```

```

        newCatOfReasonButton.Enabled = true;
        removeCatOfReasonButton.Enabled = true;
        reasonField.Enabled = true;
    }

    private void newCatOfReasonButtonClick(object sender, EventArgs e)
    {
        addCatOfReasonButton.Enabled = true;
        catOfReasonField.Enabled = true;
        newCatOfReasonButton.Enabled = false;
        removeCatOfReasonButton.Enabled = false;
        addReasonButton.Enabled = false;
        reasonField.Enabled = false;
    }

    int check = 0;
    private void removeCatOfReasonButtonClick(object sender, EventArgs e)
    {
        for (int i = 0; i < catOfReasonList.Items.Count; ++i)
        {
            if (catOfReasonList.GetSelected(i) == true)
            {
                ++check;
            }
        }
        if (check == 0)
        {
            MessageBox.Show("Не выбрана категория, которую нужно удалить." + "\r\n" +
                "Выберите номер категории из списка и повторите попытку!", "Ошибка",
                MessageBoxButtons.OK);
        }
        else
        {
            treeView1.Nodes[0].Nodes.RemoveAt(catOfReasonList.SelectedIndex);
            for (int i = 0; i < dataGridView1.Rows.Count; ++i) {
                if (Convert.ToString(dataGridView1.Rows[i].Cells[catOfReasonList.SelectedIndex].Value) !=
                    "")
                {
                    dataGridView2.Columns.Remove(Convert.ToString(dataGridView1.Rows[i].Cells[catOfReasonList.SelectedIndex].Value));
                }
            }
            dataGridView1.Columns.Remove(Convert.ToString(catOfReasonList.SelectedItem));
            addReasonList.Items.Remove(catOfReasonList.SelectedItem);
            catOfReasonList.Items.Remove(catOfReasonList.SelectedItem);
            count_of_cat_of_reason--;
        }
    }

```

```

        check = 0;
    }

    private void list_of_reason_SelectedIndexChanged(object sender, EventArgs e)
    {
        if (catOfReasonList.Items.Count == 0)
        {
            removeCatOfReasonButton.Enabled = false;
            addReasonButton.Enabled = false;
            reasonField.Enabled = false;
        }
    }

    private void list_of_reason_SelectedValueChanged(object sender, EventArgs e)
    {
        for (int i = 0; i < catOfReasonList.Items.Count; ++i)
        {
            if (catOfReasonList.GetSelected(i) == true)
            {
                ++check;
            }
        }
        if (check == 0)
        {
            removeCatOfReasonButton.Enabled = false;
        }
        else removeCatOfReasonButton.Enabled = true;
        check = 0;
    }

    int b = 0;
    private void addReasonButtonClick(object sender, EventArgs e)
    {
        for (int i = 0; i < addReasonList.Items.Count; ++i)
        {
            if (addReasonList.GetSelected(i) == true)
            {
                ++check;
            }
        }
        if (check == 0)
        {
            MessageBox.Show("Не выбрана ячейка добавления причины." + "\r\n" +
                "Выберите ячейку из списка и повторите попытку!", "Ошибка", MessageBoxButtons.OK);
        }
        else
        {
            dataGridview1.Rows.Add();
        }
    }

```

```

        for (int i = 0; i < dataGridView1.Rows.Count; ++i)
        {
            if (Convert.ToString(dataGridView1.Rows[i].Cells[addReasonList.SelectedIndex].Value) == "")
            {
                treeView1.Nodes[0].Nodes[addReasonList.SelectedIndex].Nodes.Add(new
                TreeNode(reasonField.Text));
                treeView1.ExpandAll();
                dataGridView1.Rows[i].Cells[addReasonList.SelectedIndex].Value = (i + 1) + ": " +
                reasonField.Text;

                dataGridView2.Columns.Add(Convert.ToString(dataGridView1.Rows[i].Cells[addReasonList.SelectedIndex].Value),
                Convert.ToString(dataGridView1.Rows[i].Cells[addReasonList.SelectedIndex].Value));

                addFactorList.Items.Add(Convert.ToString(dataGridView1.Rows[i].Cells[addReasonList.SelectedIndex].Value));

                for (int a = 0; a < reasonList.Items.Count; ++a)
                {
                    if (addReasonList.SelectedItem == reasonList.Items[a]) {
                        ++b;
                    }
                }
                if (b == 0)
                {
                    reasonList.Items.Add(addReasonList.SelectedItem);
                    b = 0;
                    break;
                }
            }
            newReasonButton.Enabled = true;
            removeReasonButton.Enabled = true;
            reasonField.Enabled = false;
            addReasonButton.Enabled = false;
            factorField.Enabled = true;
        }
        check = 0;
    }

    private void reasonButtonEnabled(object sender, EventArgs e)
    {
        if (reasonField.Text == "")
        {
            addReasonButton.Enabled = false;
        }
        else addReasonButton.Enabled = true;
    }

    private void newReasonButtonClick(object sender, EventArgs e)
    {

```



```

        reasonField.Enabled = true;
        addReasonButton.Enabled = true;
        newReasonButton.Enabled = false;
    }

    string temp = "";
    private void removeReasonButtonClick(object sender, EventArgs e)
    {
        for (int i = 0; i < addReasonList.Items.Count; ++i)
        {
            if (addReasonList.GetSelected(i) == true)
            {
                ++check;
            }
        }
        if (check == 0)
        {
            MessageBox.Show("Не выбрана ячейка добавления причины." + "\r\n" +
                "Выберите ячейку из списка и повторите попытку!", "Ошибка", MessageBoxButtons.OK);
        }
        else
        {
            for (int i = 0; i < dataGridView1.Rows.Count; ++i)
            {
                if (Convert.ToString(dataGridView1.Rows[i].Cells[addReasonList.SelectedIndex].Value) == "")
                {
                    if (i == 0)
                    {
                        MessageBox.Show("В выбранном столбце удалить нечего!", "Ошибка",
                            MessageBoxButtons.OK);
                        break;
                    }
                    else
                    {
                        temp = Convert.ToString(dataGridView1.Rows[i -
                            1].Cells[addReasonList.SelectedIndex].Value);
                        string f = Convert.ToString(dataGridView1.Rows[i -
                            1].Cells[addReasonList.SelectedIndex].Value);
                        string a = Convert.ToString(f[0]);

                        treeView1.Nodes[0].Nodes[addReasonList.SelectedIndex].Nodes.RemoveAt(Convert.ToInt32(a) - 1);
                        dataGridView2.Columns.Remove(temp);
                        addFactorList.Items.Remove(Convert.ToString(dataGridView1.Rows[i -
                            1].Cells[addReasonList.SelectedIndex].Value));
                        Convert.ToString(dataGridView1.Rows[i - 1].Cells[addReasonList.SelectedIndex].Value =
                            "");
                        break;
                    }
                }
            }
        }
    }

```

```

        }
    }
}
check = 0;
temp = "";
}

private void addFactorButtonEnabled(object sender, EventArgs e)
{
    if (factorField.Text == "")
    {
        addFactorButton.Enabled = false;
    }
    else addFactorButton.Enabled = true;
}

private void addFactorButtonClick(object sender, EventArgs e)
{
    for (int i = 0; i < addFactorList.Items.Count; ++i)
    {
        if (addFactorList.GetSelected(i) == true)
        {
            ++check;
        }
    }
    for (int i = 0; i < reasonList.Items.Count; ++i)
    {
        if (reasonList.GetSelected(i) == true)
        {
            ++check;
        }
    }
    if (check == 0 || check == 1)
    {
        MessageBox.Show("Не выбрана(ы) ячейка(и) добавления фактора." + "\r\n" +
            "Выберите ячейку(и) из списка и повторите попытку!", "Ошибка",
            MessageBoxButtons.OK);
    }
    else
    {
        int ccheck = 0;
        for (int i = 0; i < dataGridView1.Rows.Count; ++i)
        {
            string ff = Convert.ToString(reasonList.SelectedItem);
            string aa = Convert.ToString(ff[1]);
            if (Convert.ToString(dataGridView1.Rows[i].Cells[Convert.ToInt32(aa) - 1].Value) ==
                Convert.ToString(addFactorList.SelectedItem))
            {

```

```

        ccheck++;
    }
}
if (ccheck == 0)
{
    MessageBox.Show("Категория " + Convert.ToString(reasonList.SelectedItem) + " не
содержит причину " + Convert.ToString(addFactorList.SelectedItem)
    + "\r\n" + "Пожалуйста, проверьте выбранные значения!");
}
else
{
    dataGridView2.Rows.Add();
    for (int i = 0; i < dataGridView2.Rows.Count; ++i)
    {
        if (Convert.ToString(dataGridView2.Rows[i].Cells[addFactorList.SelectedIndex].Value) ==
""")
        {
            dataGridView2.Rows[i].Cells[addFactorList.SelectedIndex].Value = (i + 1) + ":" +
factorField.Text;
            string f = Convert.ToString(addFactorList.SelectedItem);
            string a = Convert.ToString(f[0]);
            string g = Convert.ToString(reasonList.SelectedItem);
            string b = Convert.ToString(g[1]);
            treeView1.Nodes[0].Nodes[Convert.ToInt32(b) - 1].Nodes[Convert.ToInt32(a) -
1].Nodes.Add(new TreeNode(factorField.Text));
            treeView1.ExpandAll();
            break;
        }
    }
    newReasonButton.Enabled = true;
    removeReasonButton.Enabled = true;
    reasonField.Enabled = false;
    addReasonButton.Enabled = false;
    factorField.Enabled = true;
    removeFactorButton.Enabled = true;
}
}
check = 0;
}

private void removeFactorButtonClick(object sender, EventArgs e)
{
    for (int i = 0; i < addFactorList.Items.Count; ++i)
    {
        if (addFactorList.GetSelected(i) == true)
        {
            ++check;
        }
    }
}

```

```

    }
    if (check == 0)
    {
        MessageBox.Show("Не выбрана ячейка добавления причины." + "\r\n" +
            "Выберите ячейку из списка и повторите попытку!", "Ошибка", MessageBoxButtons.OK);
    }
    else
    {
        int ccheck = 0;
        for (int i = 0; i < dataGridView1.Rows.Count; ++i)
        {
            string ff = Convert.ToString(reasonList.SelectedItem);
            string aa = Convert.ToString(ff[1]);
            if (Convert.ToString(dataGridView1.Rows[i].Cells[Convert.ToInt32(aa) - 1].Value) ==
                Convert.ToString(addFactorList.SelectedItem))
            {
                ccheck++;
            }
        }
        if (ccheck == 0)
        {
            MessageBox.Show("Категория " + Convert.ToString(reasonList.SelectedItem) + " не
                содержит причину " + Convert.ToString(addFactorList.SelectedItem)
                + "\r\n" + "Пожалуйста, проверьте выбранные значения!");
        }
        else
        {
            for (int i = 0; i < dataGridView2.Rows.Count; ++i)
            {
                if (Convert.ToString(dataGridView2.Rows[i].Cells[addFactorList.SelectedIndex].Value) ==
                    "")
                {
                    if (i == 0)
                    {
                        MessageBox.Show("В выбранном столбце удалить нечего!", "Ошибка",
                            MessageBoxButtons.OK);
                        break;
                    }
                    else
                    {
                        string f = Convert.ToString(addFactorList.SelectedItem);
                        string a = Convert.ToString(f[0]);
                        string g = Convert.ToString(reasonList.SelectedItem);
                        string b = Convert.ToString(g[1]);
                        treeView1.Nodes[0].Nodes[Convert.ToInt32(b) - 1].Nodes[Convert.ToInt32(a) -
                            1].Nodes.RemoveAt(Convert.ToInt32(a) - 1);
                        treeView1.ExpandAll();
                    }
                }
            }
        }
    }
}

```

```

Convert.ToString(dataGridView2.Rows[i - 1].Cells[addFactorList.SelectedIndex].Value =
""");
        break;
    }
}
}
}
}
check = 0;
}

```

```

private void saveDataButtonClick(object sender, EventArgs e)
{
    Excel.Application XlApp = new Excel.Application();
    Excel.Workbook XlWorkBook = XlApp.Workbooks.Open(@"C:\Users\Д\HC\Desktop\save1.xlsx");
    Excel.Worksheet XlWorkSheet = (Excel.Worksheet)XlWorkBook.Worksheets.get_Item(1);
    for (int i = 1; i < dataGridView1.Columns.Count + 1; ++i)
    {
        XlWorkSheet.Cells[1, i] = dataGridView1.Columns[i - 1].HeaderText;
    }

    for (int i = 0; i < dataGridView1.Rows.Count; i++)
    {
        for (int j = 0; j < dataGridView1.ColumnCount; j++)
        {
            XlWorkSheet.Cells[i + 2, j + 1] = dataGridView1.Rows[i].Cells[j].Value;
        }
    }
    XlApp.Visible = true;

    Excel.Application XlApp2 = new Excel.Application();
    Excel.Workbook XlWorkBook2 = XlApp.Workbooks.Open(@"C:\Users\Д\HC\Desktop\save2.xlsx");
    Excel.Worksheet XlWorkSheet2 = (Excel.Worksheet)XlWorkBook2.Worksheets.get_Item(1);
    for (int i = 1; i < dataGridView2.Columns.Count + 1; ++i)
    {
        XlWorkSheet2.Cells[1, i] = dataGridView2.Columns[i - 1].HeaderText;
    }

    for (int i = 0; i < dataGridView2.Rows.Count; i++)
    {
        for (int j = 0; j < dataGridView2.ColumnCount; j++)
        {
            XlWorkSheet2.Cells[i + 2, j + 1] = dataGridView2.Rows[i].Cells[j].Value;
        }
    }
    XlApp2.Visible = true;
    MessageBox.Show("Сохранено!", "Уведомление", MessageBoxButtons.OK);
}

```

```

private void loadDataButtonClick(object sender, EventArgs e)
{
    resultField.Enabled = false;
    resultField.Clear();
    catOfReasonField.Enabled = false;
    catOfReasonField.Clear();
    reasonField.Enabled = false;
    reasonField.Clear();
    factorField.Enabled = false;
    factorField.Clear();
    addResultButton.Enabled = false;
    addCatOfReasonButton.Enabled = false;
    newCatOfReasonButton.Enabled = false;
    removeCatOfReasonButton.Enabled = false;
    addReasonButton.Enabled = false;
    newReasonButton.Enabled = false;
    removeReasonButton.Enabled = false;
    addFactorButton.Enabled = false;
    removeFactorButton.Enabled = false;
    dataGridView1.Columns.Clear();
    dataGridView2.Columns.Clear();
    treeView1.Nodes.Clear();
    catOfReasonList.Items.Clear();
    addReasonList.Items.Clear();
    addFactorList.Items.Clear();
    reasonList.Items.Clear();
    rootField.Enabled = true;

    OpenFileDialog opf = new OpenFileDialog();
    opf.Filter = "Excel (*.XLSX)|*.XLSX";
    opf.ShowDialog();
    DataTable tb = new DataTable();
    string filename = opf.FileName;
    string ConStr = string.Format("Provider=Microsoft.ACE.OLEDB.12.0;Data Source=" + filename + ";
Extended Properties=Excel 12.0;", filename);
    System.Data.DataSet ds = new System.Data.DataSet("EXCEL");
    OleDbConnection cn = new OleDbConnection(ConStr);
    cn.Open();
    DataTable schemaTable = cn.GetOleDbSchemaTable(OleDbSchemaGuid.Tables, new object[] {
null, null, null, "TABLE" });
    string sheet1 = (string)schemaTable.Rows[0].ItemArray[2];
    string select = String.Format("SELECT * FROM [{0}]", sheet1);
    OleDbDataAdapter ad = new OleDbDataAdapter(select, cn);
    ad.Fill(ds);
    tb = ds.Tables[0];
    cn.Close();
    dataGridView1.DataSource = tb;

```

```

OpenFileDialog opf2 = new OpenFileDialog();
opf2.Filter = "Excel (*.XLSX)|*.XLSX";
opf2.ShowDialog();
DataTable tb2 = new DataTable();
string filename2 = opf2.FileName;
string ConStr2 = string.Format("Provider=Microsoft.ACE.OLEDB.12.0;Data Source=" + filename2 +
"; Extended Properties=Excel 12.0;", filename2);
System.Data.DataSet ds2 = new System.Data.DataSet("EXCEL");
OleDbConnection cn2 = new OleDbConnection(ConStr2);
cn2.Open();
DataTable schemaTable2 = cn2.GetOleDbSchemaTable(OleDbSchemaGuid.Tables, new object[] {
null, null, null, "TABLE" });
string sheet12 = (string)schemaTable2.Rows[0].ItemArray[2];
string select2 = String.Format("SELECT * FROM [{0}]", sheet12);
OleDbDataAdapter ad2 = new OleDbDataAdapter(select2, cn2);
ad2.Fill(ds2);
tb2 = ds2.Tables[0];
cn2.Close();
dataGridView2.DataSource = tb2;

```

```

}

```

```

string f;
private void showTreeButtonClick(object sender, EventArgs e)
{
    if (rootField.Text == "")
    {
        MessageBox.Show("Сначала введите корень!");
    }
    else
    {
        for (int i = 0; i < dataGridView1.ColumnCount; ++i)
        {
            treeView1.Nodes[0].Nodes.Add(new TreeNode(dataGridView1.Columns[i].Name));
        }

        for (int i = 0; i < dataGridView1.ColumnCount; ++i)
        {
            for (int j = 0; j < dataGridView1.Rows.Count; ++j)
            {
                if (Convert.ToString(dataGridView1.Rows[j].Cells[i].Value) == "")
                {
                    continue;
                }
            }
        }
    }
}

```

```

        treeView1.Nodes[0].Nodes[i].Nodes.Add(new
TreeNode(Convert.ToString(dataGridView1.Rows[j].Cells[i].Value)));
    }
}

for (int i = 0; i < dataGridView2.ColumnCount; ++i)
{
    for (int j = 0; j < dataGridView2.Rows.Count; ++j)
    {
        if (Convert.ToString(dataGridView2.Rows[j].Cells[i].Value) == "")
        {
            continue;
        }
        else
        {
            for (int k = 0; k < dataGridView1.ColumnCount; ++k)
            {
                for (int l = 0; l < dataGridView1.Rows.Count; ++l)
                {
                    if (Convert.ToString(dataGridView1.Rows[l].Cells[k].Value) == "")
                    {
                        continue;
                    }
                    if (Convert.ToString(dataGridView1.Rows[l].Cells[k].Value) ==
Convert.ToString(dataGridView2.Columns[i].Name))
                    {
                        f = dataGridView1.Columns[k].Name;
                        break;
                    }
                }
            }
            string g = Convert.ToString(dataGridView2.Columns[i].Name);
            string a = Convert.ToString(f[1]);
            string b = Convert.ToString(g[0]);
            treeView1.Nodes[0].Nodes[Convert.ToInt32(a) - 1].Nodes[Convert.ToInt32(b) -
1].Nodes.Add(new TreeNode(Convert.ToString(dataGridView2.Rows[j].Cells[i].Value)));
        }
    }
}

MessageBox.Show("Дерево построено!");
}
}

private void addRootButtonClick(object sender, EventArgs e)
{
    showTreeButton.Enabled = true;
    treeView1.Nodes.Add(new TreeNode(rootField.Text));
}

```



```

        addRootButton.Enabled = false;
        rootField.Enabled = false;
    }

    private void addRootButtonEnabled(object sender, EventArgs e)
    {
        if (rootField.Text == "")
        {
            addRootButton.Enabled = false;
        }
        else addRootButton.Enabled = true;
    }
}
}

```

Скриншоты работы

Рис. 2. Форма

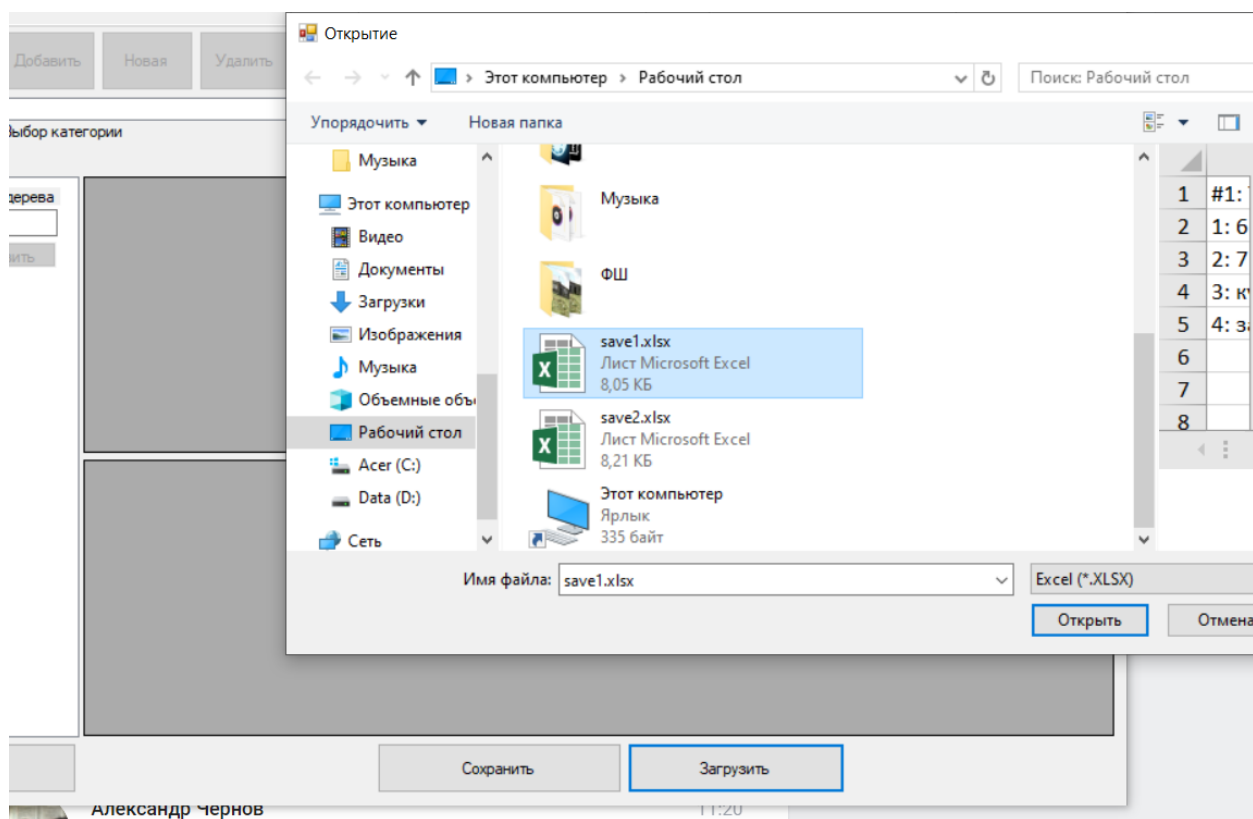


Рис. 3. Импорт данных из Excel

Form1

Введите результат (корень дерева)

Добавить Новый

Категория причин

Добавить Новая Удалить

Причина

Добавить Новая Удалить

Фактор

Добавить Удалить

Выбор категории

Выбор категории добавления/удаления

Выбор категории и причины добавления/удаления

Корень дерева	#1: ТП	#2: ГДИ	#3: ТИДЗ
1: 6 лаба	1: 5 лаба	1: 2 лаба	
2: 7 лаба	2: зачет?	2: экзамен	
3: курсач			
4: зачет			
*			

Корень дерева	#1: ТП	#2: ГДИ	#3: ТИДЗ	1: 6 лаба	2: 7 лаба	3: курсач	4: зачет	1: 5 лаба	2: зачет?	1: 2 лаба	2
1: на почту	1: на почту	1: на курсачу	1: оф, альфа бет...	1: с оценкой	1: на диск	1: с оценкой?	1: звонить деду	1:			
*											

Вывести на экран Сохранить Загрузить

Рис. 4. Загруженные данные

Введите результат (корень дерева)

Категория причин

Причина

Фактор

Добавить Новый

Добавить Новая Удалить

Добавить Новая Удалить

Добавить Удалить

Выбор категории

Выбор категории добавления/удаления

Выбор категории и причины добавления/удаления

4 семестр

#1: ТП

1: 6 лаба

1: на почту

2: 7 лаба

1: курсач

3: курсач

1: оф, альфа бета, документация

4: зачет

1: с оценкой

#2: ГДИ

1: 5 лаба

1: на диск

2: зачет?

1: с оценкой?

#3: ТИДЗ

1: 2 лаба

1: звонить деду

2: экзамен

1: после лаб

Корень дерева

4 семестр

Добавить

#1: ТП	#2: ГДИ	#3: ТИДЗ
1: 6 лаба	1: 5 лаба	1: 2 лаба
2: 7 лаба	2: зачет?	2: экзамен
3: курсач		
4: зачет		
*		

1: 6 лаба	2: 7 лаба	3: курсач	4: зачет	1: 5 лаба	2: зачет?	1: 2 лаба	2:
1: на почту	1: курсач	1: оф, альфа бет...	1: с оценкой	1: на диск	1: с оценкой?	1: звонить деду	1:
*							

Вывести на экран

Сохранить

Загрузить

Рис. 5. Данные, представленные в виде дерева

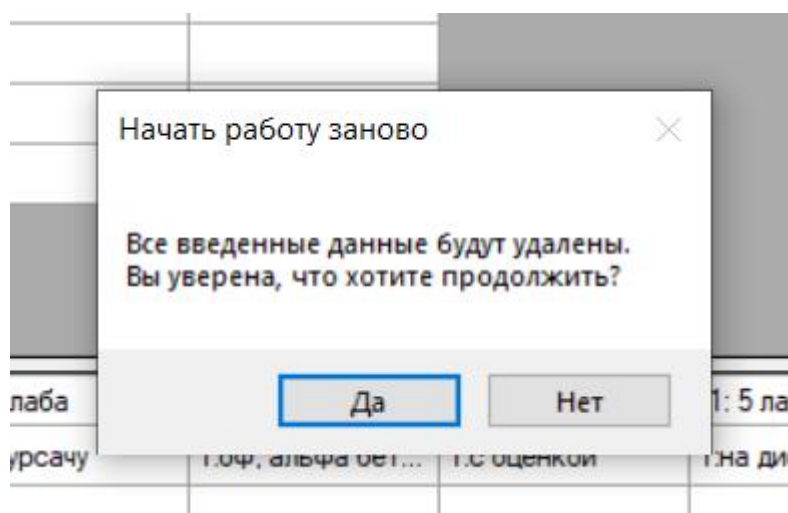


Рис. 6. Очистка данных

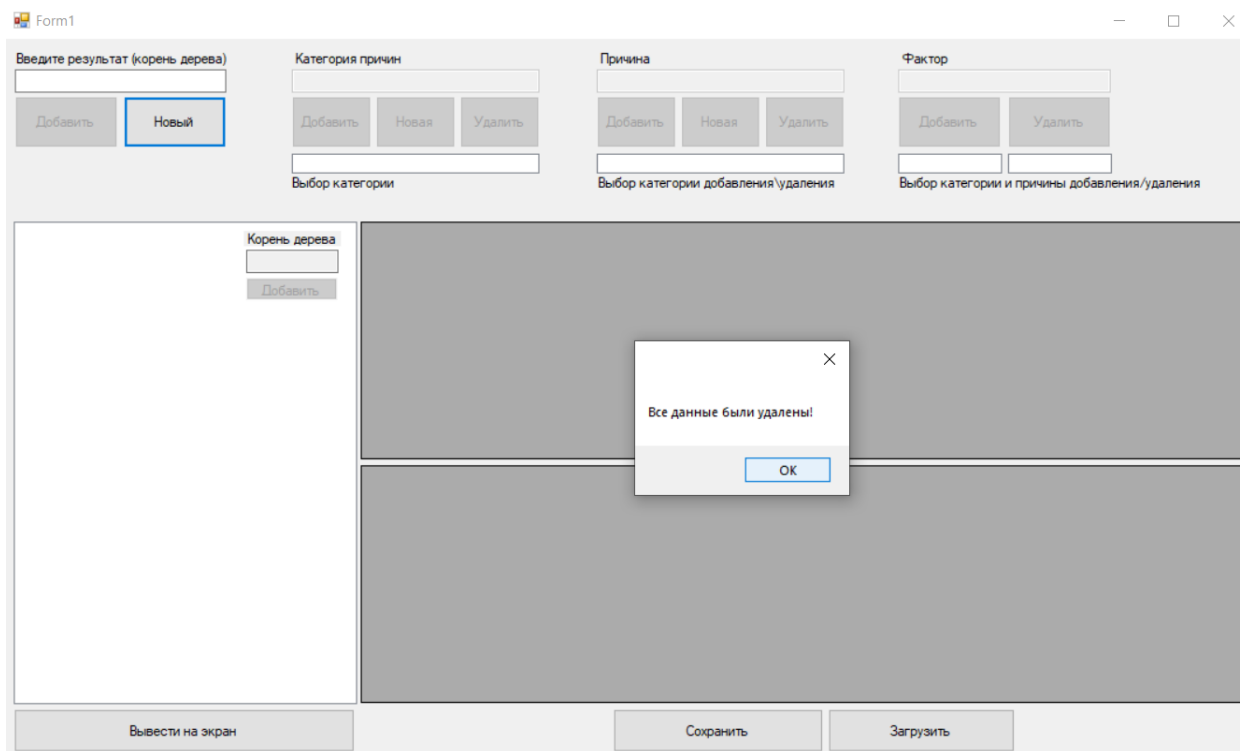


Рис. 7. Очистка

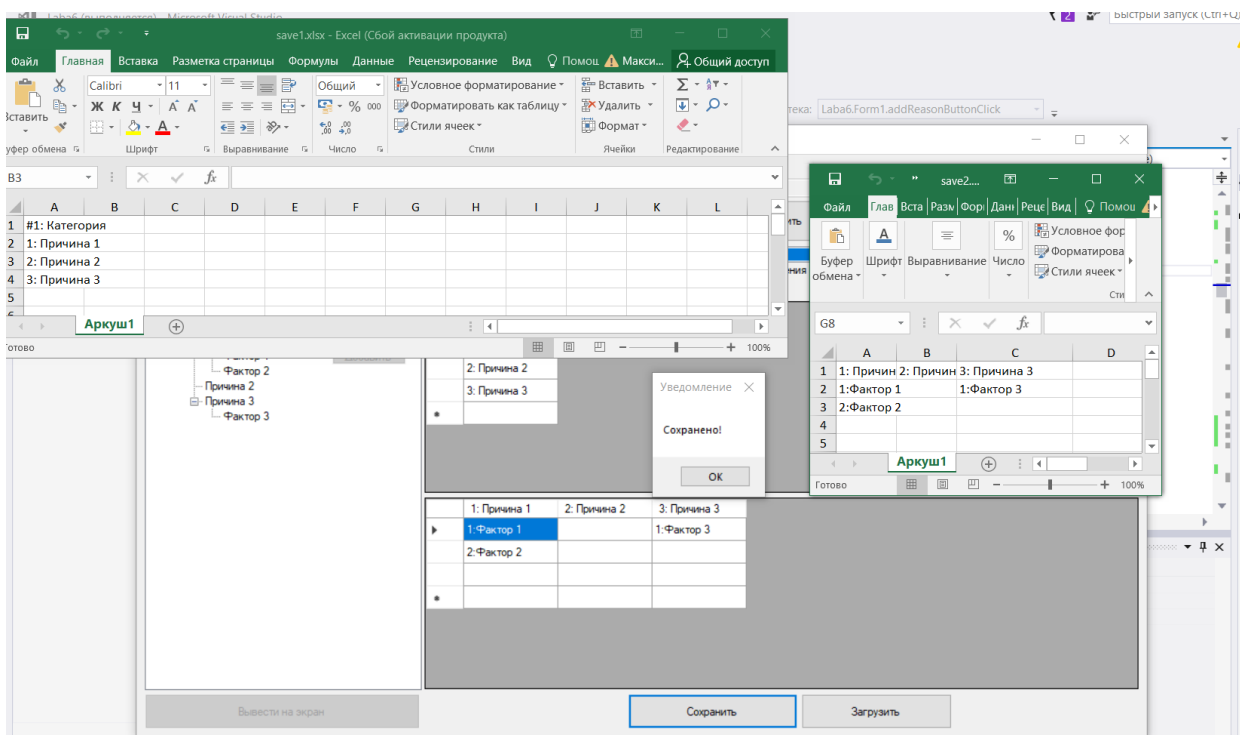


Рис. 8. Сохранение в файлах Excel

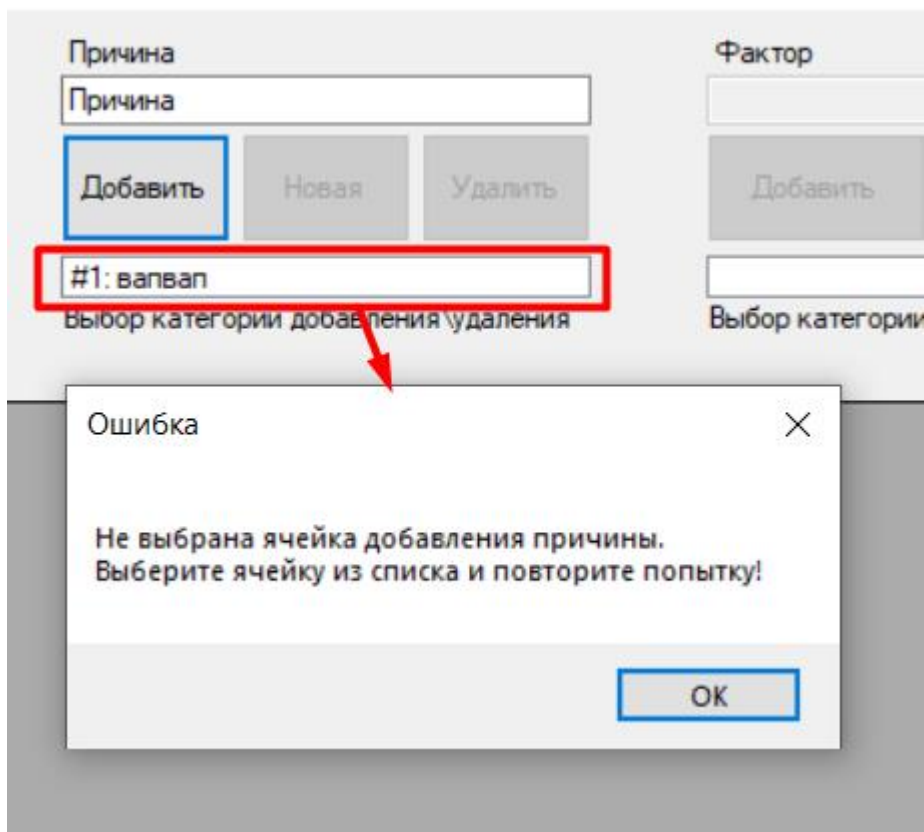


Рис. 9. Сообщение об ошибке, если не выбрана категория

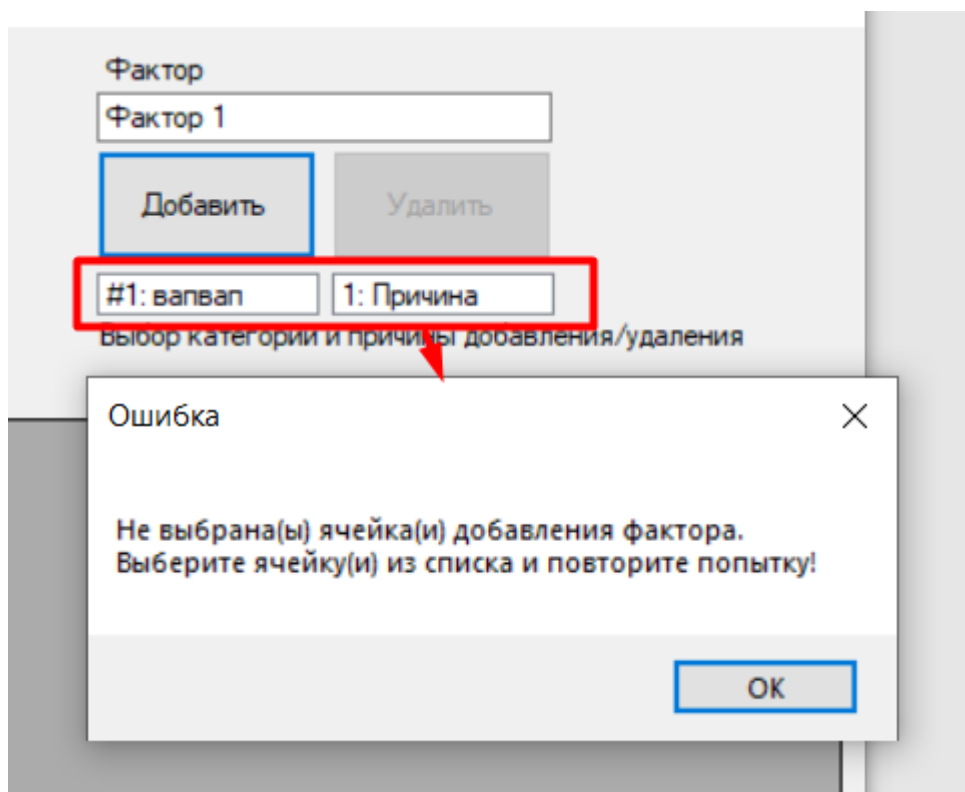


Рис. 10. Сообщение об ошибке, если не выбраны категория и причина

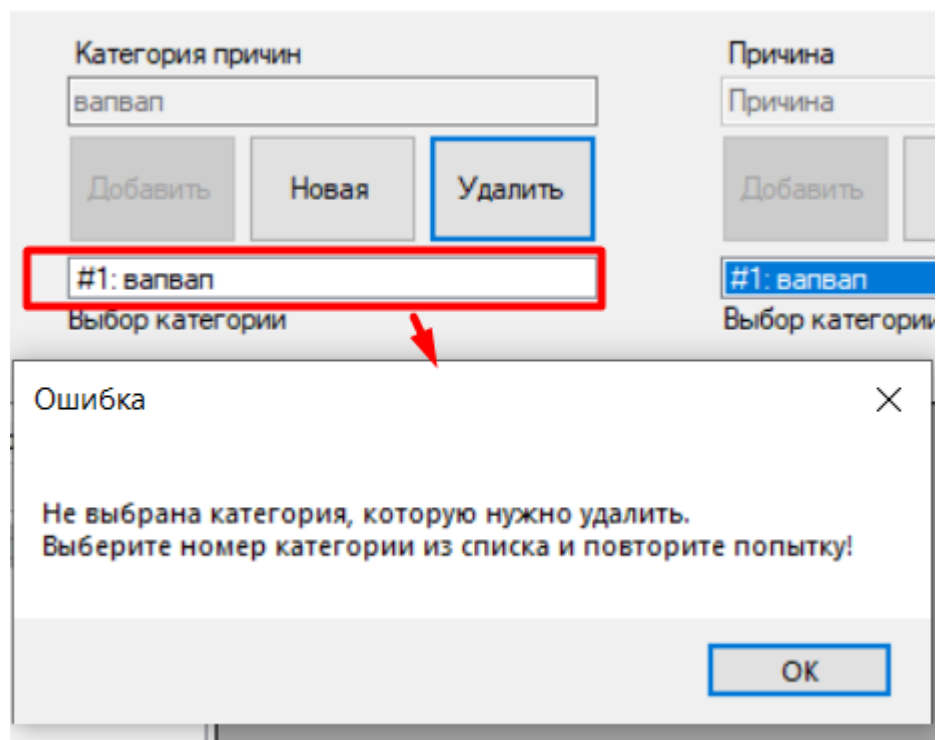


Рис. 11. Сообщение об ошибке, если не выбрана категория, которую нужно удалить