Приложение

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
КАФЕДРА ИНТЕЛЛЕКТУАЛЬНЫХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Разработка программы альтернативы Microsoft Excel
Код программы
КП.ПО-8. 1-40 01 01

Листов 8

Руководитель	Самолюк О. Ю.
Выполнил	Буртик Я. В.
Консультант по ЕСПД	Самолюк О.Ю.

2023

MainWindow.xaml.cs

```
using System.ComponentModel;
using System.ComponentModel;
using System.Data;

using System.Windows;
using System.Windows.Controls;


namespace CourseProject;


public partial class MainWindow : Window
{
    private bool isEdit = false;
    public MainWindow()
    {
        InitializeComponent();
        DataBase.GetTables();
        DbComboBox.ItemsSource = DataBase._tables;
    }

    private void AddNewButton_OnClick(object sender, RoutedEventArgs e)
    {
        AddNewDb addNewDb = new AddNewDb();
        addNewDb.ShowDialog();
        DbComboBox.ItemsSource = null;
        DbComboBox.ItemsSource = DataBase._tables;
    }

    private DataTable _dt;
    private void DeleteButton_OnClick(object sender, RoutedEventArgs e)
    {
        if (DbComboBox.Text != "")
        {
            switch (MessageBox.Show($"Would you like to delete group
{DbComboBox.Text}? ", "",
                        MessageBoxButton.YesNo))
            {
                case MessageBoxResult.Yes:
                    DataBase.DeleteTable(DbComboBox.Text);
                    MessageBox.Show("Success");
                    break;
                case MessageBoxResult.No:
                    return;
            }
            DbComboBox.ItemsSource = null;
            DbComboBox.ItemsSource = DataBase._tables;
            DbComboBox.SelectedIndex = 0;
        }
    }


    private void EditTablesButton_OnClick(object sender, RoutedEventArgs e)
    {
        if (DbComboBox.Text!="")
        {
            EditWindow editWindow = new EditWindow(DbComboBox.Text);
            editWindow.ShowDialog();
            Update(prevStr);
        }

    }

    private void SaveButton_OnClick(object sender, RoutedEventArgs e)
    {
        if (DbComboBox.Text != "")
        {
            DataBase.SaveTable($"{DbComboBox.Text}_a",_dt);
            MessageBox.Show("Success");
        }

    }

    private void DbComboBox_OnSelectionChanged(object sender, 
SelectionChangedEventArgs e)
    {
        if (isEdit)
        {
            if (MessageBox.Show("Would u like to save 
changed?","",MessageBoxButton.YesNo) == MessageBoxResult.Yes)
            {
                DataBase.SaveTable($"{prevStr}_a",_dt);
                MessageBox.Show("Success");
            }

        }
        if (DbComboBox.ItemsSource != null)
        {
            Update(DbComboBox.SelectedItem.ToString());
            prevStr = DbComboBox.SelectedItem.ToString();
        }


    }

    private string prevStr;
    private void Update(string? s)
    {
        var str = s;
        if (str != "")
        {
            _dt = DataBase.ShowTable(str += "_a");
            DataGridVeiw.ItemsSource = _dt.DefaultView;
        }

        isEdit = false;
    }


    private void DataGridVeiw_OnRowEditEnding(object? sender, 
DataGridRowEditEndingEventArgs e)
    {
        isEdit = true;
    }
}
```

EditWindow.xaml.cs

```
using System.Data;
using System.Windows;
using Microsoft.Data.SqlClient;

namespace CourseProject;

public partial class EditWindow : Window
{
    public EditWindow(string name)
    {
        InitializeComponent();

        _name = name;
        _name2 = name;
        _dataTableDate = DataBase.ShowTable(_name2 += "_d");
        DataGridDate.ItemsSource = _dataTableDate.DefaultView;
    }

    private string _name2;
    private string _name;
    private DataTable _dataTableDate;
    private void Save_OnClick(object sender, RoutedEventArgs e)
    {
        DataBase.SaveTable(_name2, _dataTableDate);
        DataBase.GenTableA(_name);
        MessageBox.Show("Success");
        Close();
    }
}
```

AddNewDb.xaml.cs

```
using System.Windows;

namespace CourseProject;

public partial class AddNewDb : Window
{
    public AddNewDb()
    {
        InitializeComponent();
    }

    private void ButtonAdd_OnClick(object sender, RoutedEventArgs e)
    {
        DataBase.CreateNewTable(TextBoxNameOfDb.Text);
        MessageBox.Show("Success");
        Close();
    }

    private void ButtonCreateNewDb_OnClick(object sender, RoutedEventArgs e)
    {
        DataBase.CreateDataBase();
        MessageBox.Show("Success");
    }
}
```

DataBase.cs

```
using System.Data;
using System.IO;
using System.Windows.Documents;
using Microsoft.Data.SqlClient;

namespace CourseProject;

public class DataBase
{
    public static List<string> _tables = new List<string>();
    private static string _connectionString =
        "Server=YAKOVLAPTOP;Database=CourseWork;TrustServerCertificate=True;
Trusted_Connection=True;";
    private static string _conForCreate =
"Server=YAKOVLAPTOP;Database=master;TrustServerCertificate=True;
Trusted_Connection=True;";

    public static int CreateDataBase()
    {
        using (SqlConnection connection = new SqlConnection(_conForCreate))
        {
            connection.Open();
            string sqlExpression = "CREATE DATABASE CourseWork";
            SqlCommand command = new SqlCommand(sqlExpression, connection);
            return command.ExecuteNonQuery();
        }
    }

    public static void CreateDB()
    {
        using (SqlConnection connection = new SqlConnection(_connectionString))
        {
            connection.Open();
            SqlCommand command =
                new SqlCommand("CREATE TABLE datatables (Id INT NOT NULL
IDENTITY, Name NVARCHAR(100) NOT NULL )");
            command.ExecuteNonQuery();
        }
    }

    public static void CreateNewTable(string name)
    {
        using (SqlConnection connection = new SqlConnection(_connectionString))
        {
            connection.Open();
            string sqlExpression = $"CREATE TABLE {name}_d (Id INT PRIMARY KEY
IDENTITY, Date NVARCHAR(100) NOT NULL );" +
                                   $"CREATE TABLE {name}_a (Id INT PRIMARY KEY
identity);";
            SqlCommand command = new SqlCommand(sqlExpression, connection);
            AddTables(name);
            GetTables();
            command.ExecuteNonQuery();
        }
    }

    public static void DeleteTable(string name)
    {
        using (SqlConnection connection = new SqlConnection(_connectionString))
        {
            connection.Open();
            string sqlExpression = $"DROP TABLE {name}_d;" +
                                   $"DROP TABLE {name}_a;";
            SqlCommand command = new SqlCommand(sqlExpression, connection);
            DeleteTables(name);
            GetTables();
            command.ExecuteNonQuery();
        }
    }

    public static DataTable ShowTable(string name)
    {
        SqlConnection connection = new SqlConnection(_connectionString);
        SqlCommand command = new SqlCommand($"SELECT * FROM 
{name};",connection);
        SqlDataAdapter sqlDataAdapter = new SqlDataAdapter(command);
        DataTable dataTable = new DataTable();
        sqlDataAdapter.Fill(dataTable);
        return dataTable;
    }



    public static void DropTableA(string name)
    {
        using (SqlConnection connection = new SqlConnection(_connectionString))
        {
            connection.Open();
            string sqlExpression = $"DROP TABLE {name}_a;";
            SqlCommand command = new SqlCommand(sqlExpression, connection);
            _tables.Remove(name);
            command.ExecuteNonQuery();
        }
    }

    public static void SaveTable(string name,DataTable dataTable)
    {
        SqlConnection connection = new SqlConnection(_connectionString);
        SqlDataAdapter sqlDataAdapter = new SqlDataAdapter($"SELECT * FROM 
{name}",connection);
        new SqlCommandBuilder(sqlDataAdapter);
        sqlDataAdapter.Update(dataTable);
    }

    public static void GetTables()
    {
        _tables = new List<string>();
        using (SqlConnection connection = new SqlConnection(_connectionString))
        {
            connection.Open();
            SqlCommand command = new SqlCommand("SELECT datatables.Name FROM 
datatables",connection);
            SqlDataReader reader = command.ExecuteReader();
            if (reader.HasRows)
            {
                while (reader.Read())
                {
                    _tables.Add(reader.GetString(0));
                }
            }
        }
    }

    public static void AddTables(string name)
    {
        using (SqlConnection connection = new SqlConnection(_connectionString))
        {
            connection.Open();
            SqlCommand command = new SqlCommand($"INSERT datatables VALUES
('{name}');",connection);
            command.ExecuteNonQuery();
        }
    }

    public static void DeleteTables(string name)
    {
        using (SqlConnection connection = new SqlConnection(_connectionString))
        {
            connection.Open();
            SqlCommand command = new SqlCommand($"DELETE datatables WHERE Name =
'{name}';",connection);
            command.ExecuteNonQuery();
        }
    }

    public static void GenTableA(string name)
    {
        DropTableA(name);
        List<string> list = new List<string>();
        using (SqlConnection connection = new SqlConnection(_connectionString))
        {
            connection.Open();
            string sqlExpression = $"SELECT * FROM {name}_d";
            SqlCommand command = new SqlCommand(sqlExpression, connection);
            SqlDataReader reader = command.ExecuteReader();
            if (reader.HasRows)
            {
                while (reader.Read()) // построчно считываем данные
                {
                    string date = reader.GetString(1);
                    list.Add(date);
                }
            }
        }

        string str = $"CREATE TABLE {name}_a (Id INT PRIMARY KEY IDENTITY, Name
NVARCHAR(100)";
        foreach (var l in list)
        {
            str += $", d{l} NVARCHAR(100)";
        }

        str += ");";
        using (SqlConnection connection = new SqlConnection(_connectionString))
        {
            connection.Open();
            SqlCommand command = new SqlCommand(str, connection);
            _tables.Add(name);
            command.ExecuteNonQuery();
        }

    }


}
```