**11** **Технологии создания графического пользовательского интерфейса**

Задание 1. Автоматизация учета сотрудников

Листинг задачи:

public partial class MainWindow : Window

{

private const string SavePath = "employees.json";

private ObservableCollection<Employee> \_employees = new ObservableCollection<Employee>();

private CollectionViewSource \_employeesViewSource;

public MainWindow()

{

InitializeComponent();

LoadData();

InitializeViewSource();

InitializeFilters();

}

private void InitializeViewSource()

{

\_employeesViewSource = new CollectionViewSource { Source = \_employees };

\_employeesViewSource.Filter += ApplyFilter;

dgEmployees.ItemsSource = \_employeesViewSource.View;

}

private void InitializeFilters()

{

rbAll.Checked += (s, e) => RefreshFilter();

rbManager.Checked += (s, e) => RefreshFilter();

rbDeveloper.Checked += (s, e) => RefreshFilter();

}

private void RefreshFilter() => \_employeesViewSource.View.Refresh();

private void ApplyFilter(object sender, FilterEventArgs e)

{

var emp = e.Item as Employee;

if (emp == null) return;

if (rbAll.IsChecked == true) e.Accepted = true;

else if (rbManager.IsChecked == true) e.Accepted = emp.Position == "Менеджер";

else if (rbDeveloper.IsChecked == true) e.Accepted = emp.Position == "Разработчик";

}

private void LoadData()

{

if (File.Exists(SavePath))

{

var json = File.ReadAllText(SavePath);

var data = JsonConvert.DeserializeObject<ObservableCollection<Employee>>(json);

\_employees = data ?? new ObservableCollection<Employee>();

}

}

private void SaveData()

{

var json = JsonConvert.SerializeObject(\_employees, Formatting.Indented);

File.WriteAllText(SavePath, json);

}

private void BtnAdd\_Click(object sender, RoutedEventArgs e)

{

if (string.IsNullOrWhiteSpace(txtFullName.Text) || string.IsNullOrWhiteSpace(txtPosition.Text))

{

return;

}

\_employees.Add(new Employee

{

FullName = txtFullName.Text,

Position = txtPosition.Text

});

ClearFields();

SaveData();

}

private void BtnEdit\_Click(object sender, RoutedEventArgs e)

{

if (dgEmployees.SelectedItem is not Employee selectedEmployee) return;

selectedEmployee.FullName = txtFullName.Text;

selectedEmployee.Position = txtPosition.Text;

ClearFields();

SaveData();

RefreshFilter();

}

private void BtnDelete\_Click(object sender, RoutedEventArgs e)

{

if (dgEmployees.SelectedItem is not Employee selectedEmployee) return;

\_employees.Remove(selectedEmployee);

SaveData();

}

public class Employee

{

public string FullName { get; set; }

public string Position { get; set; }

}

}

Таблица 1.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| ФИО, Должность | Добавление сотрудника. Сообщение об ошибке |

Анализ результатов:

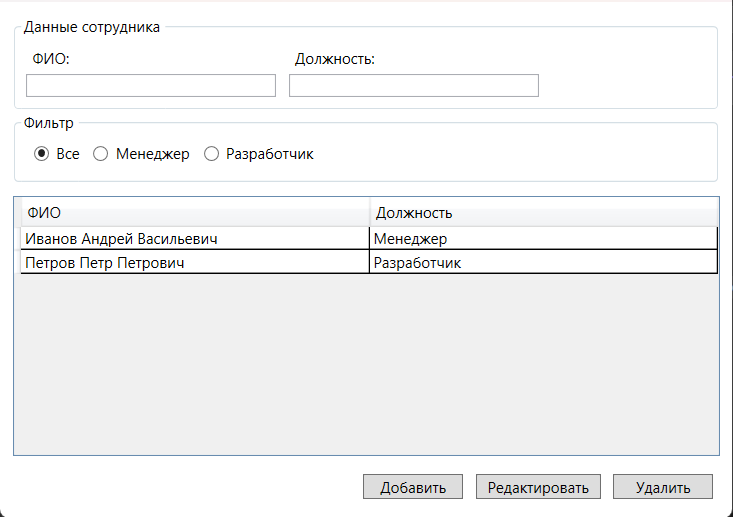


Рисунок 1.1 – Результат работы программы