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

Задание 1. Автоматизация учета сотрудников

Листинг задачи:

public partial class MainWindow : Window

{

private const string SavePath = "employees.json";

public ObservableCollection<Employee> Employees { get; set; } = new ObservableCollection<Employee>();

public ICommand AddEmployeeCommand { get; }

public ICommand EditEmployeeCommand { get; }

public ICommand DeleteEmployeeCommand { get; }

public MainWindow()

{

InitializeComponent();

DataContext = this;

LoadData();

AddEmployeeCommand = new RelayCommand(\_ => AddEmployee());

EditEmployeeCommand = new RelayCommand(\_ => EditEmployee(), \_ => lbEmployees.SelectedItem != null);

DeleteEmployeeCommand = new RelayCommand(\_ => DeleteEmployee(), \_ => lbEmployees.SelectedItem != null);

}

private void AddEmployee()

{

var dialog = new EmployeeWindow();

if (dialog.ShowDialog() == true)

{

Employees.Add(dialog.Employee);

SaveData();

}

}

private void EditEmployee()

{

var selected = lbEmployees.SelectedItem as Employee;

var dialog = new EmployeeWindow(selected);

if (dialog.ShowDialog() == true)

{

int index = Employees.IndexOf(selected);

Employees[index] = dialog.Employee;

SaveData();

}

}

private void DeleteEmployee()

{

var result = MessageBox.Show("Удалить сотрудника?", "Подтверждение",

MessageBoxButton.YesNo, MessageBoxImage.Warning);

if (result == MessageBoxResult.Yes && lbEmployees.SelectedItem is Employee employee)

{

Employees.Remove(employee);

SaveData();

}

}

private void Filter\_Checked(object sender, RoutedEventArgs e)

{

var radio = sender as RadioButton;

if (radio == null || radio.IsChecked != true) return;

var view = CollectionViewSource.GetDefaultView(Employees);

if (radio == rbAll)

{

view.Filter = null;

}

else

{

string department = radio.Content.ToString();

view.Filter = item => (item as Employee)?.Department == department;

}

}

private void LoadData()

{

if (File.Exists(SavePath))

{

var json = File.ReadAllText(SavePath);

Employees = JsonConvert.DeserializeObject<ObservableCollection<Employee>>(json)

?? new ObservableCollection<Employee>();

}

}

Таблица 1.1 – Входные и выходные данные

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

Анализ результатов:

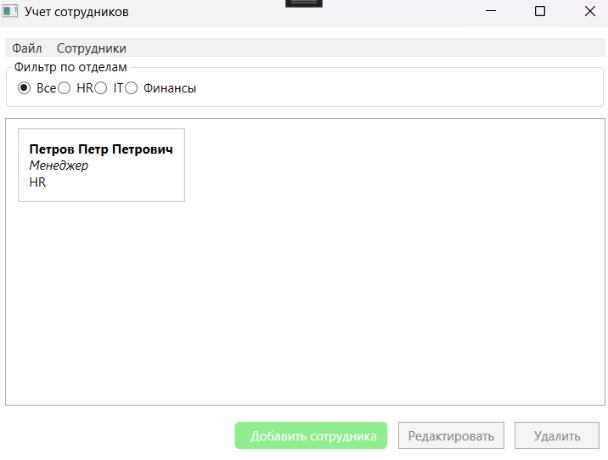


Рисунок 1.1 – Результат работы программы