**12** **Многопользовательское приложение**

Задание 1. Автоматизация учета сотрудников

Листинг задачи:

public class EmployeeViewModel : INotifyPropertyChanged

{

private readonly EmployeeService \_employeeService;

private ObservableCollection<EmployeeModel> \_employees;

private List<DepartmentModel> \_departments;

private EmployeeModel \_selectedEmployee;

private string \_filterDepartment = "Все";

private bool \_isLoading;

private int \_progressValue;

public event PropertyChangedEventHandler PropertyChanged;

public ObservableCollection<EmployeeModel> Employees

{

get => \_employees;

set

{

\_employees = value;

OnPropertyChanged(nameof(Employees));

EmployeesView = CollectionViewSource.GetDefaultView(\_employees);

EmployeesView.Filter = FilterEmployees;

}

}

public ICollectionView EmployeesView { get; private set; }

public List<DepartmentModel> Departments

{

get => \_departments;

set

{

\_departments = value;

OnPropertyChanged(nameof(Departments));

}

}

public EmployeeModel SelectedEmployee

{

get => \_selectedEmployee;

set

{

\_selectedEmployee = value;

OnPropertyChanged(nameof(SelectedEmployee));

CommandManager.InvalidateRequerySuggested();

}

}

public string FilterDepartment

{

get => \_filterDepartment;

set

{

if (\_filterDepartment != value)

{

\_filterDepartment = value;

OnPropertyChanged(nameof(FilterDepartment));

EmployeesView?.Refresh();

}

}

}

public bool IsLoading

{

get => \_isLoading;

set

{

\_isLoading = value;

OnPropertyChanged(nameof(IsLoading));

}

}

public int ProgressValue

{

get => \_progressValue;

set

{

\_progressValue = value;

OnPropertyChanged(nameof(ProgressValue));

}

}

Таблица 1.1 – Входные и выходные данные

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

Анализ результатов:

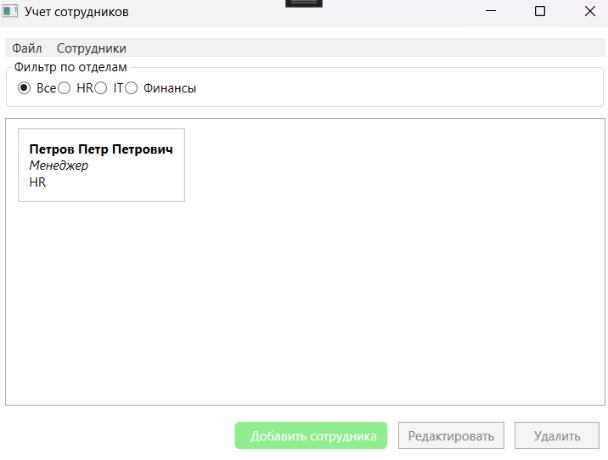


Рисунок 1.1 – Результат работы программы