**Технологии создания графического пользовательского**

**интерфейса**

Задание 1. Электронный дневник студента

Функционал: ведение оценок студентов.

✅ MVVM:

Создать StudentModel, GradeModel.

Реализовать StudentViewModel, содержащую список студентов и их

оценки.

Добавить команды для выставления оценок.

✅ Разделение логики и представления:

Вынести логику работы с оценками в GradeService.

В View оставить ListBox для отображения студентов.

✅ Асинхронное программирование:

Сделать загрузку списка студентов асинхронной.

При загрузке показывать индикатор.

Листинг программы:

public class StudentViewModel : INotifyPropertyChanged

{

private readonly GradeService \_gradeService;

private bool \_isLoading;

private ObservableCollection<StudentModel> \_students;

public ObservableCollection<StudentModel> Students

{

get => \_students;

set

{

\_students = value;

OnPropertyChanged(nameof(Students));

}

}

public bool IsLoading

{

get => \_isLoading;

set

{

\_isLoading = value;

OnPropertyChanged(nameof(IsLoading));

}

}

public ICommand AddGradeCommand { get; }

public StudentViewModel()

{

\_gradeService = new GradeService();

AddGradeCommand = new RelayCommand(AddGrade);

}

public async Task LoadStudentsAsync()

{

IsLoading = true;

Students = new ObservableCollection<StudentModel>(await \_gradeService.GetStudentsAsync());

IsLoading = false;

}

private void AddGrade(object parameter)

{

// Логика для добавления оценки

}

public event PropertyChangedEventHandler PropertyChanged;

protected virtual void OnPropertyChanged(string propertyName)

{

PropertyChanged?.Invoke(this, new PropertyChangedEventArgs(propertyName));

}

}

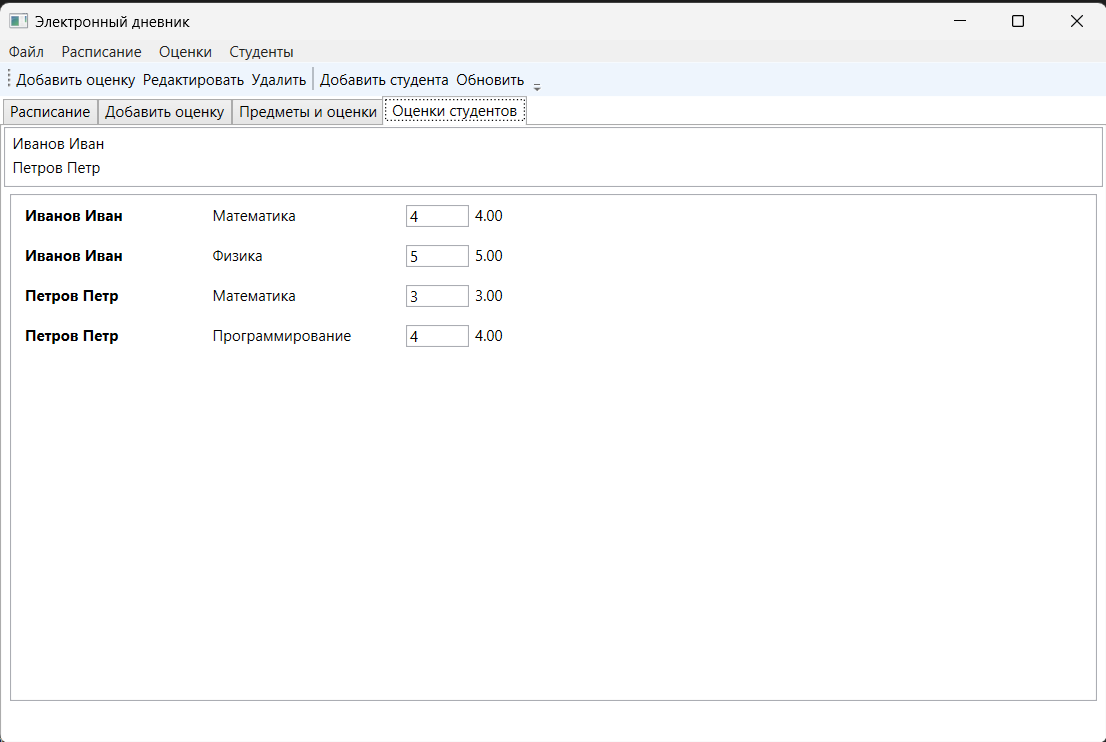


Рисунок 14.1 - Результат работы программы