**14** **Технологии создания графического пользовательского интерфейса**

Задание 1. Электронный дневник студента

Листинг задачи:

namespace StudentDiary

{

public class MainViewModel : INotifyPropertyChanged

{

public Schedule CurrentSchedule { get; } = new Schedule();

private Grade? \_selectedGrade;

private DateTime? \_filterDate;

private readonly ObservableCollection<Grade> \_grades = new();

private ObservableCollection<Grade> \_filteredGrades = new();

public ObservableCollection<Grade> FilteredGrades

{

get => \_filteredGrades;

set

{

\_filteredGrades = value;

OnPropertyChanged(nameof(FilteredGrades));

}

}

public DateTime? FilterDate

{

get => \_filterDate;

set

{

\_filterDate = value;

OnPropertyChanged(nameof(FilterDate));

FilterGrades();

}

}

public Grade? SelectedGrade

{

get => \_selectedGrade;

set

{

\_selectedGrade = value;

OnPropertyChanged(nameof(SelectedGrade));

CommandManager.InvalidateRequerySuggested();

}

}

public ICommand AddGradeCommand { get; }

public ICommand EditGradeCommand { get; }

public ICommand DeleteGradeCommand { get; }

public ICommand FilterGradesCommand { get; }

public MainViewModel()

{

AddGradeCommand = new RelayCommand(\_ => AddGrade());

EditGradeCommand = new RelayCommand(\_ => EditGrade(), \_ => CanEditDelete());

DeleteGradeCommand = new RelayCommand(\_ => DeleteGrade(), \_ => CanEditDelete());

FilterGradesCommand = new RelayCommand(\_ => FilterGrades());

LoadSampleGrades();

FilterGrades();

}

private void LoadSampleGrades()

{

\_grades.Add(new Grade { Subject = "Математика", Date = DateTime.Now.AddDays(-3), Score = 5 });

\_grades.Add(new Grade { Subject = "Физика", Date = DateTime.Now.AddDays(-1), Score = 4 });

}

private bool CanEditDelete() => SelectedGrade != null;

private void AddGrade()

{

var dialog = new GradeEditWindow();

if (dialog.ShowDialog() == true)

{

\_grades.Add(dialog.Grade);

FilterGrades();

}

}

private void EditGrade()

{

if (SelectedGrade == null) return;

var dialog = new GradeEditWindow(SelectedGrade);

if (dialog.ShowDialog() == true)

{

var index = \_grades.IndexOf(SelectedGrade);

\_grades[index] = dialog.Grade;

SelectedGrade = dialog.Grade;

FilterGrades();

}

}

private void DeleteGrade()

{

if (MessageBox.Show("Удалить оценку?", "Подтверждение", MessageBoxButton.YesNo) == MessageBoxResult.Yes)

{

\_grades.Remove(SelectedGrade!);

FilterGrades();

}

}

private void FilterGrades()

{

if (FilterDate == null)

{

FilteredGrades = new ObservableCollection<Grade>(\_grades);

return;

}

var filtered = \_grades.Where(g => g.Date.Date == FilterDate.Value.Date).ToList();

FilteredGrades = new ObservableCollection<Grade>(filtered);

}

public event PropertyChangedEventHandler? PropertyChanged;

protected virtual void OnPropertyChanged(string propertyName)

{

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

}

}

}}

Таблица 1.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Предмет, дата, оценка | Добавление предмета, оценки. Сообщение об ошибке |

Анализ результатов:

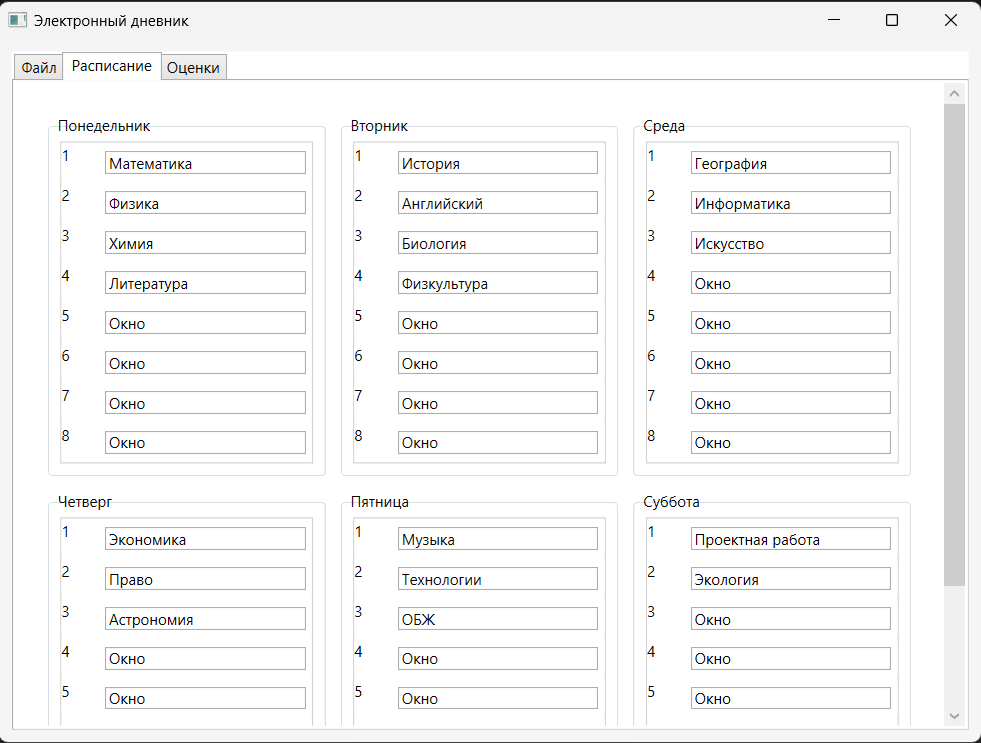


Рисунок 1.1 – Результат работы программы