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

Задание 1. Электронный дневник студента

Листинг задачи:

namespace StudentDiary

{

public partial class MainWindow : Window, INotifyPropertyChanged

{

private const string SaveFilePath = "grades.xml";

private string \_newSubject;

private CollectionViewSource \_filteredGrades;

public string NewSubject

{

get => \_newSubject;

set

{

\_newSubject = value;

OnPropertyChanged(nameof(NewSubject));

}

}

public ObservableCollection<GradeEntry> Grades { get; } = new ObservableCollection<GradeEntry>();

public ObservableCollection<string> AllSubjects { get; } = new ObservableCollection<string>();

public MainWindow()

{

InitializeComponent();

DataContext = this;

\_filteredGrades = new CollectionViewSource { Source = Grades };

GradesGrid.ItemsSource = \_filteredGrades.View;

LoadData();

Grades.CollectionChanged += (s, e) => UpdateSubjects();

Closing += (s, e) => SaveData();

}

private void LoadData()

{

try

{

if (File.Exists(SaveFilePath))

{

var serializer = new XmlSerializer(typeof(ObservableCollection<GradeEntry>));

using (var stream = new FileStream(SaveFilePath, FileMode.Open))

{

var savedData = (ObservableCollection<GradeEntry>)serializer.Deserialize(stream);

Grades.Clear();

foreach (var item in savedData) Grades.Add(item);

UpdateSubjects();

}

}

}

catch (Exception ex)

{

MessageBox.Show($"Ошибка загрузки: {ex.Message}");

}

}

private void SaveData()

{

try

{

var serializer = new XmlSerializer(typeof(ObservableCollection<GradeEntry>));

using (var stream = new FileStream(SaveFilePath, FileMode.Create))

{

serializer.Serialize(stream, Grades);

}

}

catch (Exception ex)

{

MessageBox.Show($"Ошибка сохранения: {ex.Message}");

}

}

private void UpdateSubjects()

{

AllSubjects.Clear();

foreach (var subject in Grades.Select(g => g.Subject).Distinct().OrderBy(s => s))

{

AllSubjects.Add(subject);

}

}

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

{

if (!ValidateInputs()) return;

Grades.Add(new GradeEntry

{

Subject = NewSubject?.Trim(),

Date = DateInput.SelectedDate.Value,

Grade = int.Parse(GradeInput.Text)

});

ClearForm();

}

private bool ValidateInputs()

{

if (string.IsNullOrWhiteSpace(NewSubject))

{

MessageBox.Show("Введите название предмета!", "Ошибка",

MessageBoxButton.OK, MessageBoxImage.Warning);

return false;

}

if (!DateInput.SelectedDate.HasValue)

{

MessageBox.Show("Выберите дату!", "Ошибка",

MessageBoxButton.OK, MessageBoxImage.Warning);

return false;

}

if (!int.TryParse(GradeInput.Text, out int grade) || grade < 1 || grade > 10)

{

MessageBox.Show("Оценка должна быть от 1 до 10!", "Ошибка",

MessageBoxButton.OK, MessageBoxImage.Warning);

return false;

}

return true;

}

private void ClearForm()

{

NewSubject = string.Empty;

DateInput.SelectedDate = null;

GradeInput.Clear();

SubjectInput.Focus();

}

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

{

\_filteredGrades.View.Filter = item =>

{

var entry = (GradeEntry)item;

bool valid = true;

if (StartDate.SelectedDate.HasValue)

valid &= entry.Date.Date >= StartDate.SelectedDate.Value.Date;

if (EndDate.SelectedDate.HasValue)

valid &= entry.Date.Date <= EndDate.SelectedDate.Value.Date;

return valid;

};

}

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:

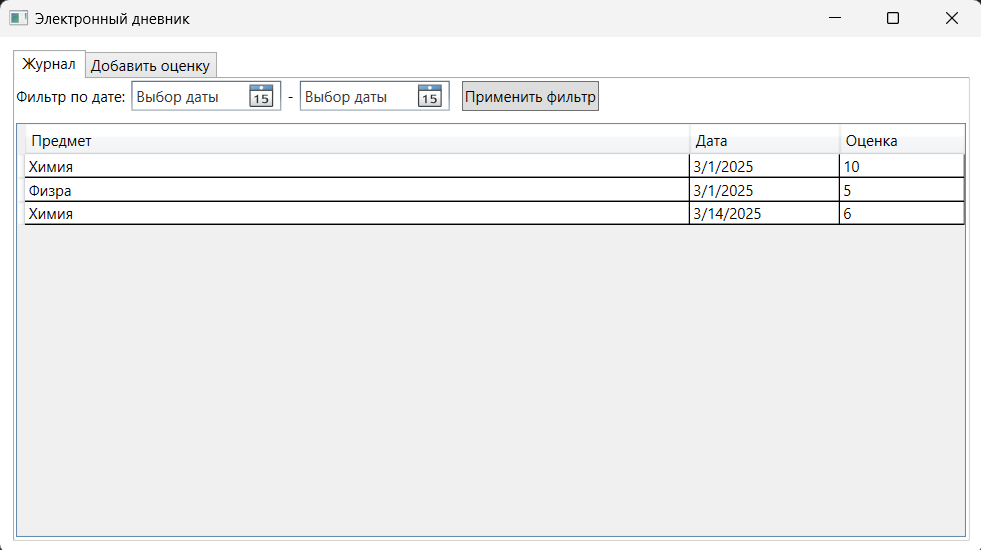


Рисунок 1.1 – Результат работы программы