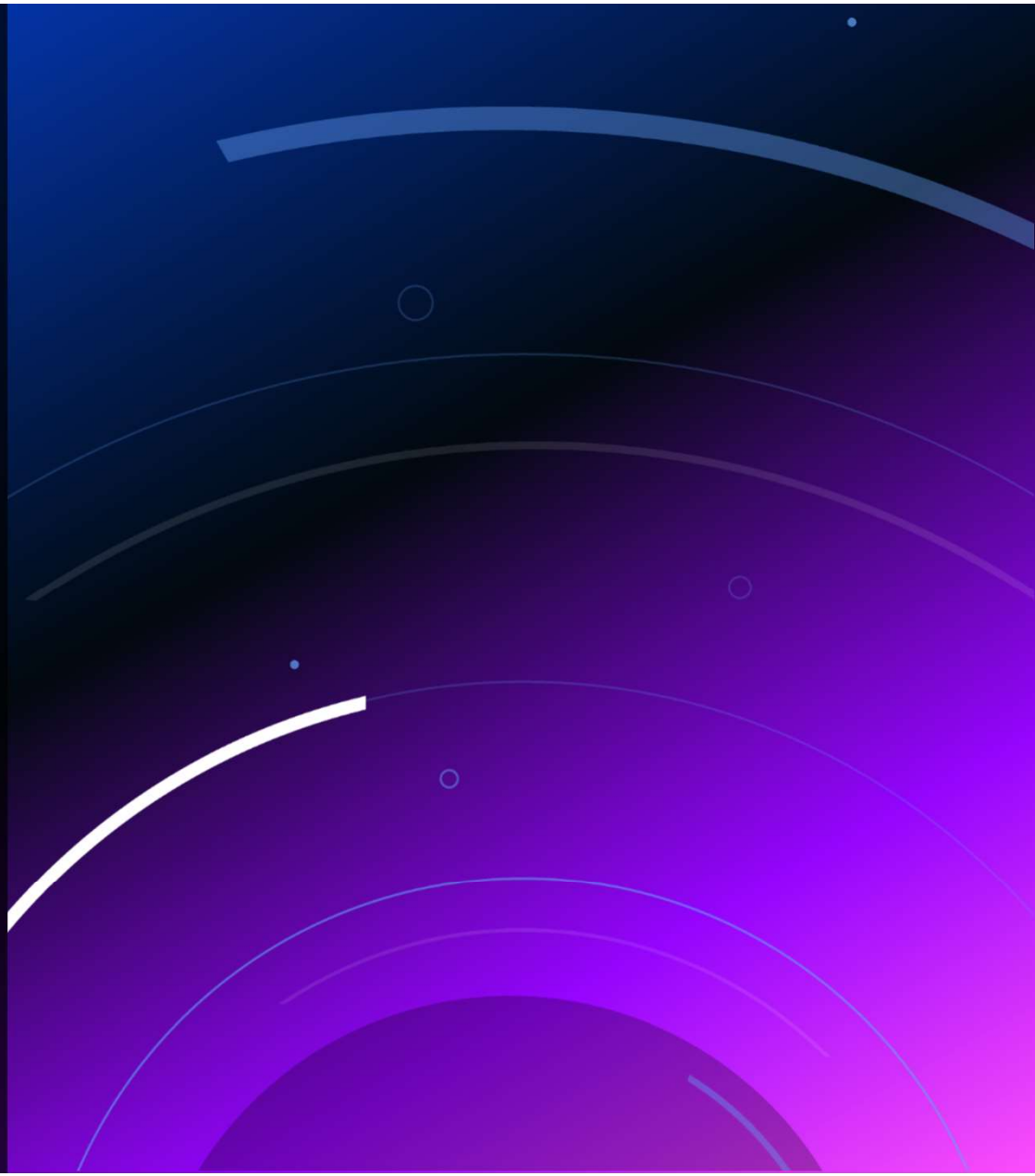

AVALONIA
MOBILE
DESKTOP
WEB

СОДЕРЖАНИЕ



ВВЕДЕНИЕ В AVALONIA

Avalonia UI — это кроссплатфор-менный фреймворк для создания современных графических приложений на C# и XAML. Он позволяет запускать одно и то же приложение на:

- Windows
- Linux
- macOS
- Android и iOS

КРАТКАЯ ИСТОРИЯ

Год	Событие
2013	Первый коммит — **Perspex** (эксперимент)
2015	Добавлен XAML → первая альфа
2016	Переименован в **Avalonia**
2019–2022	Версия **0.10** — стабильный десктоп
2023	**11.0** — мобильные + веб + AOT
2024–2025	**11.1+** — TV, производительность, инструменты

Что дальше?

Rich Text Editor

AI-инструменты для UI

Ещё лучше мобильный UX

Глубокая интеграция с .NET 9+

КРАТКАЯ ИСТОРИЯ

Почему выбирают Avalonia?

Один код → 7+ платформ

Полный .NET

Open Source (MIT)

Активное сообщество

Hot Reload, производительность

Avalonia = WPF 2.0 для всего мира

Почему не просто улучшить WPF?

Microsoft не может — WPF привязан к Windows API

Архитектура устарела — DirectX, Win32, COM

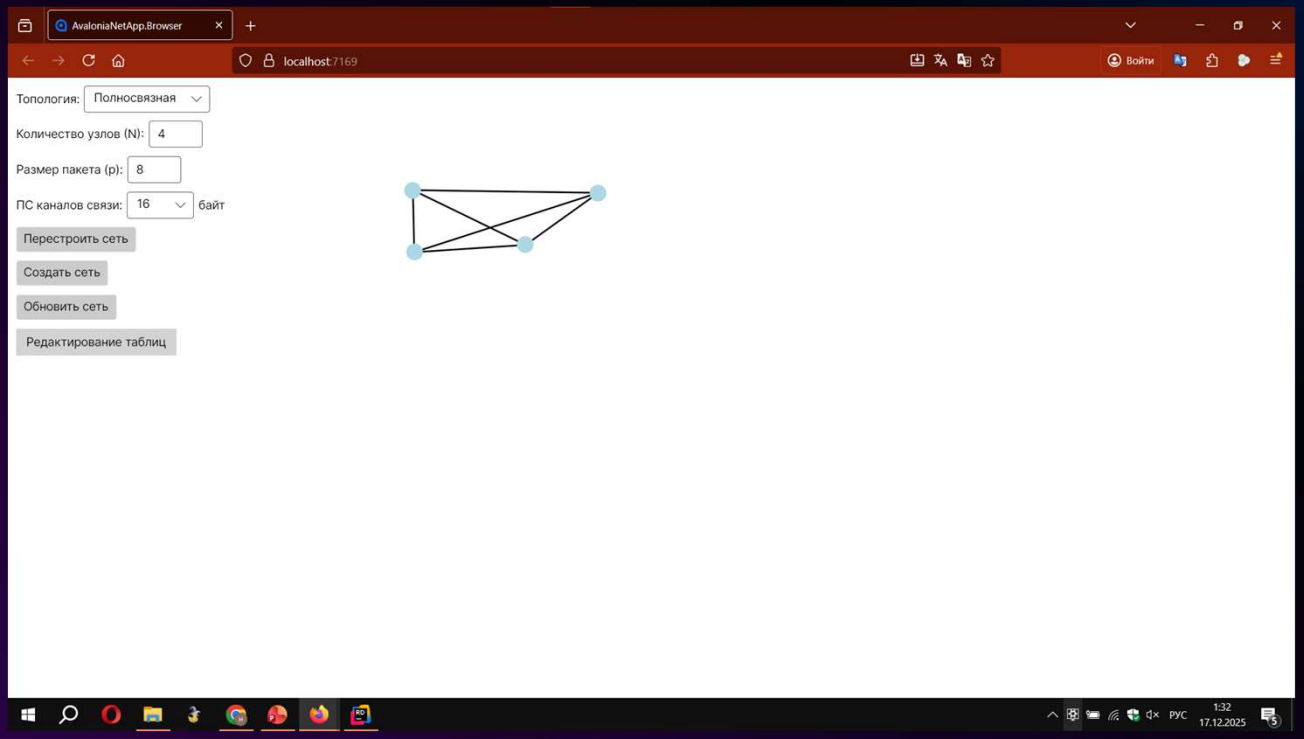
Нет планов на кросс-платформу — фокус на WinUI / MAUI

Сообщество хотело свободу → родился **Perspex** → **Avalonia**

ОСНОВНЫЕ ВОЗМОЖНОСТИ

Поддержка всех популярных платформ. Как пример:

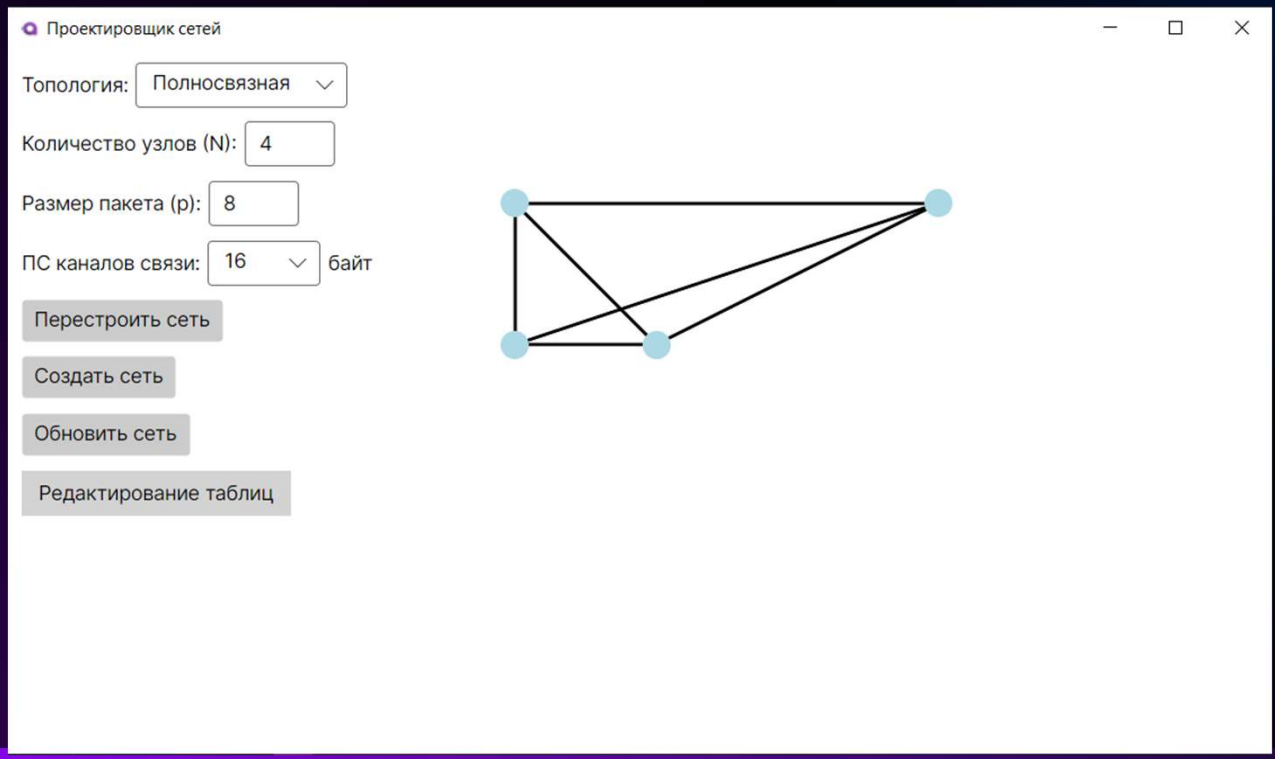
Браузер



ОСНОВНЫЕ ВОЗМОЖНОСТИ

Поддержка всех популярных платформ. Как пример:

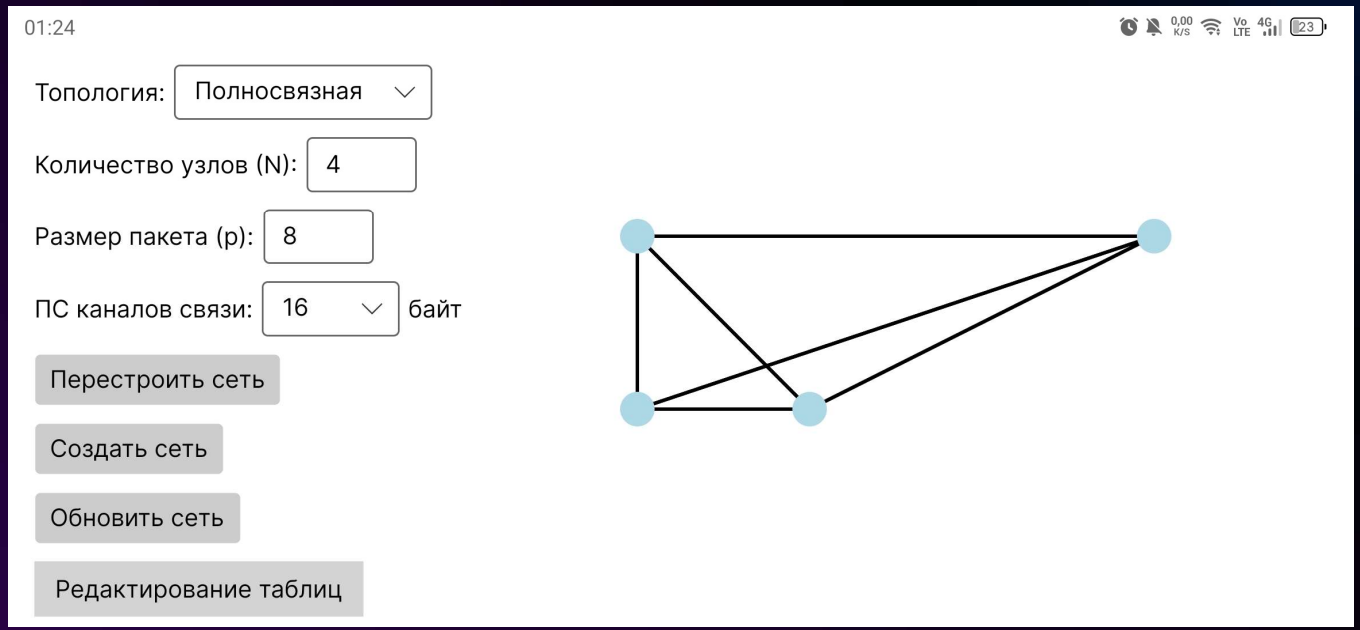
Десктоп



ОСНОВНЫЕ ВОЗМОЖНОСТИ

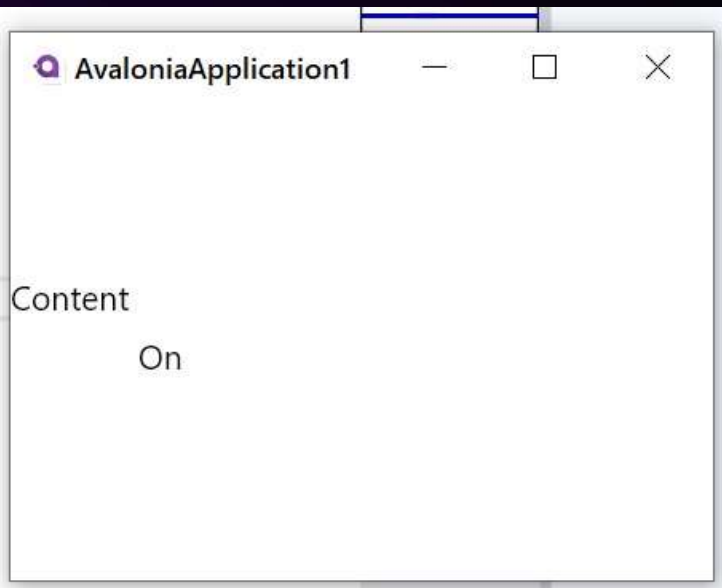
Поддержка всех популярных платформ. Как пример:

Мобильный



ОСНОВНЫЕ ВОЗМОЖНОСТИ

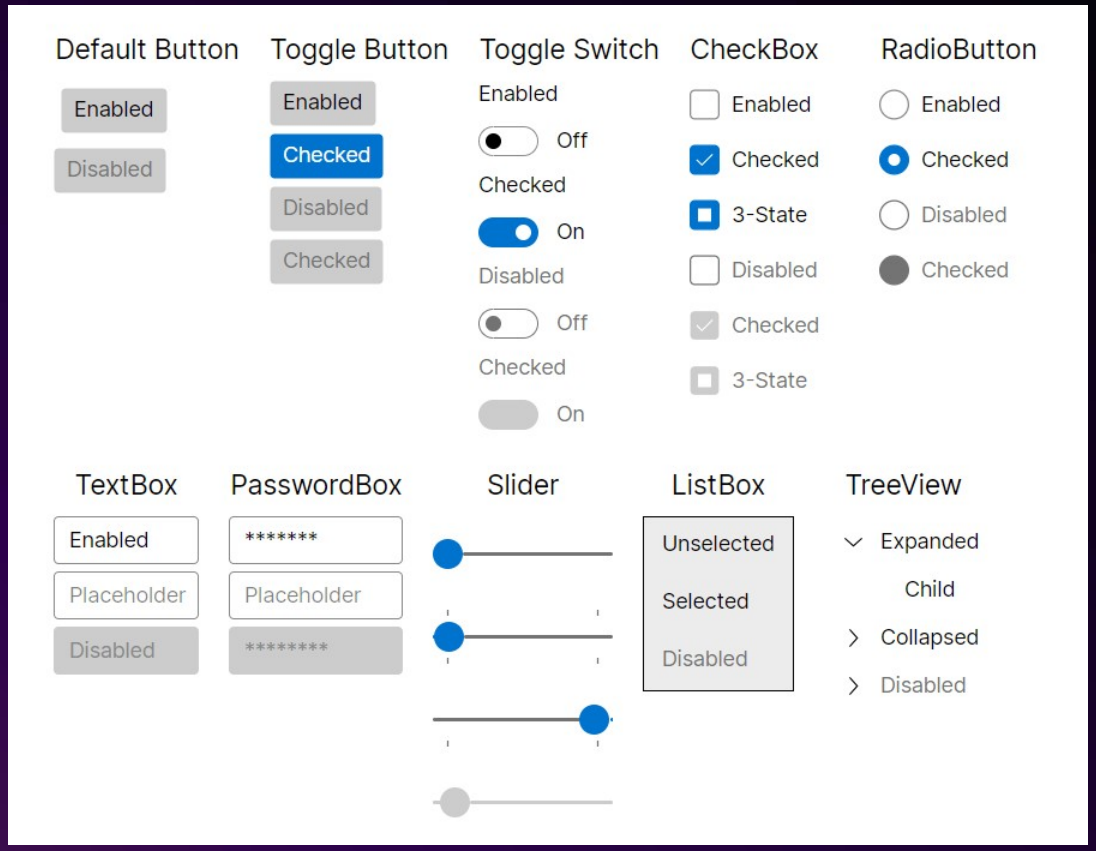
```
    d:DesignHeight="450"
    d:DesignWidth="800"
    Icon="/Assets/avalonia-logo.ico"
    mc:Ignorable="d">
    <Grid>
        <ToggleSwitch
            Content="Content"
            OffContent="Off"
            OnContent="On" />
    </Grid>
</Window>
```



Использование знакомого XAML-синтаксиса (как в WPF и UWP).

ОСНОВНЫЕ ВОЗМОЖНОСТИ

Fluent

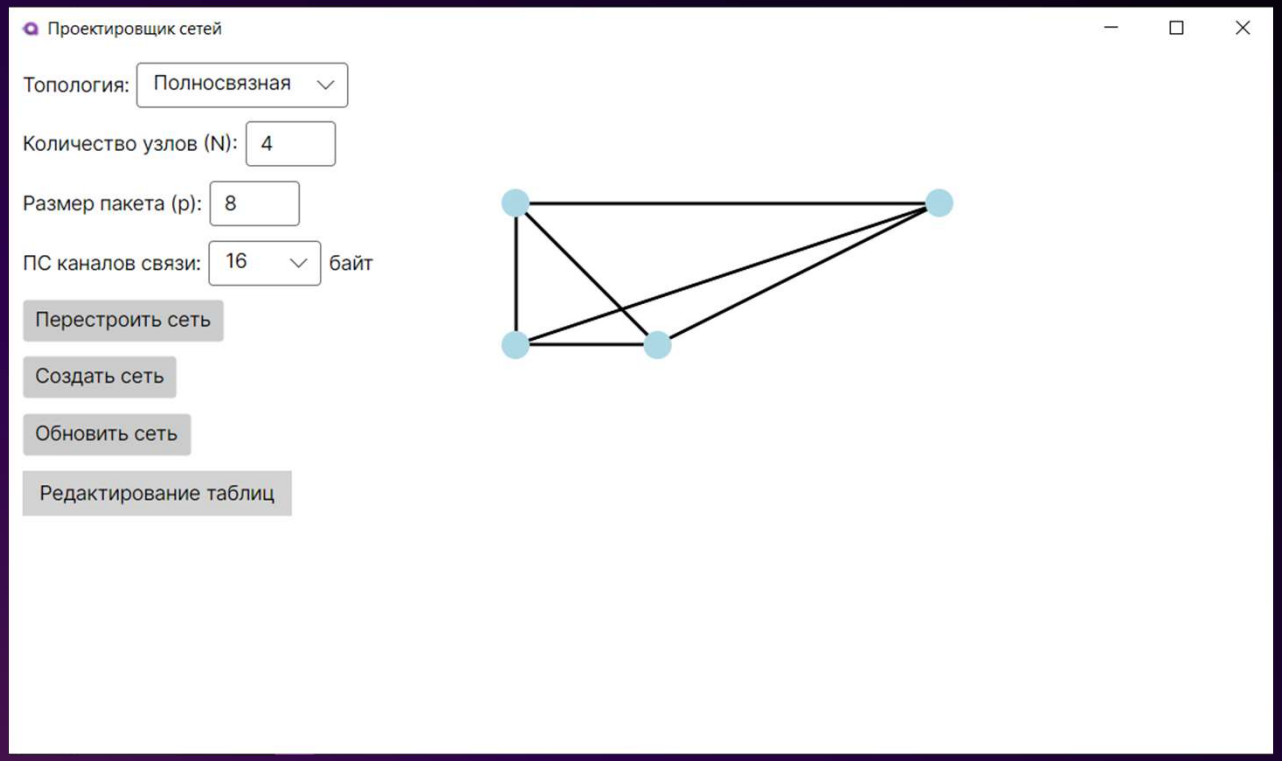


Мощная система стилей и тем (Fluent, Simple, Material).

Данная тема основана на рекомендациях от Microsoft по созданию интерактивного и привлекательного интерфейса.

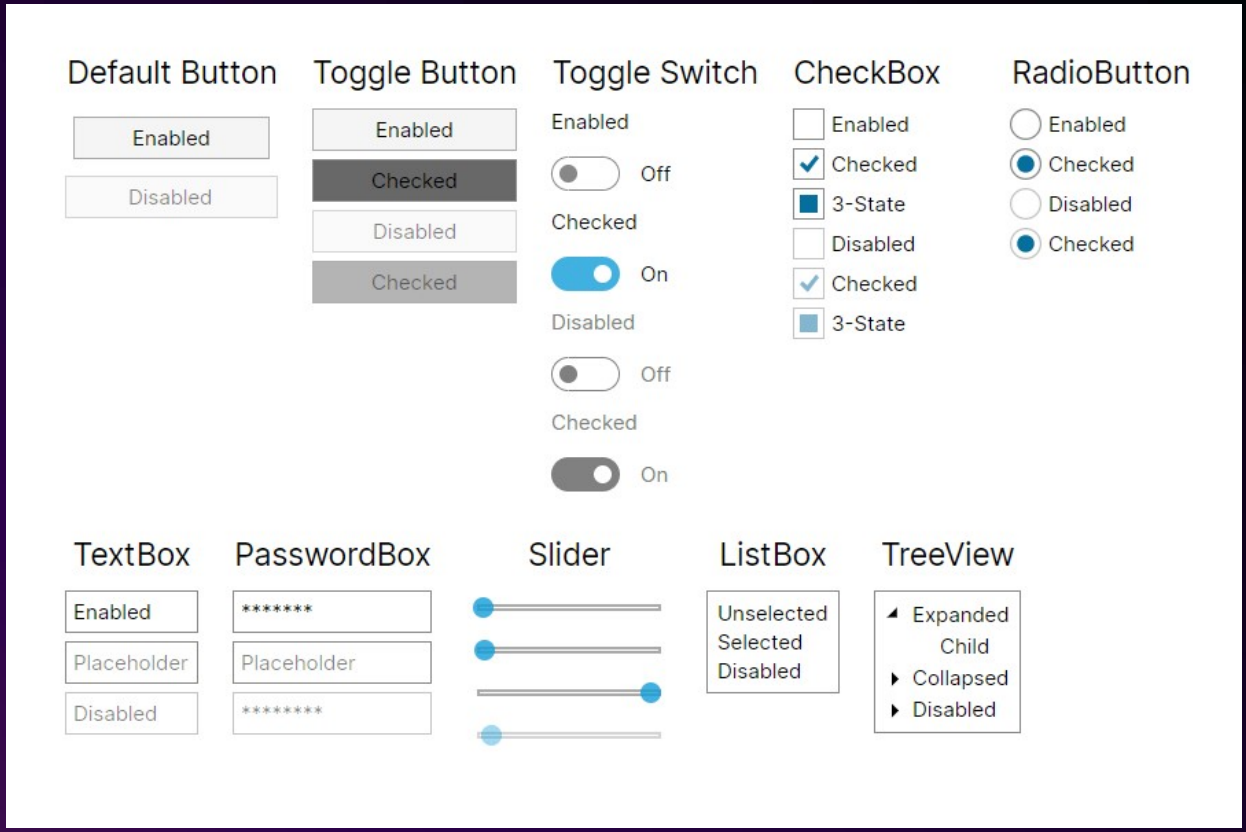
ОСНОВНЫЕ ВОЗМОЖНОСТИ

Fluent на примере приложения:



ОСНОВНЫЕ ВОЗМОЖНОСТИ

Simple

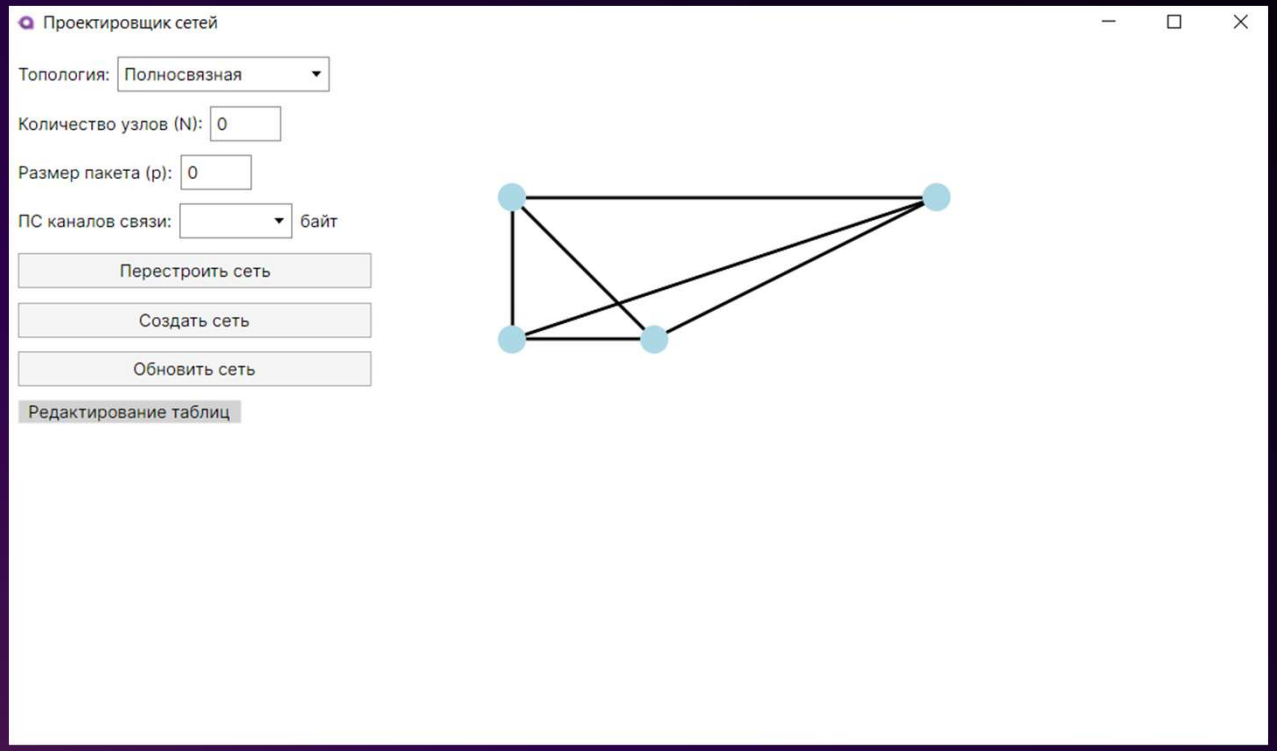


Мощная система стилей и тем (Fluent, Simple, Material).

Это самая простая и минималистичная тема.

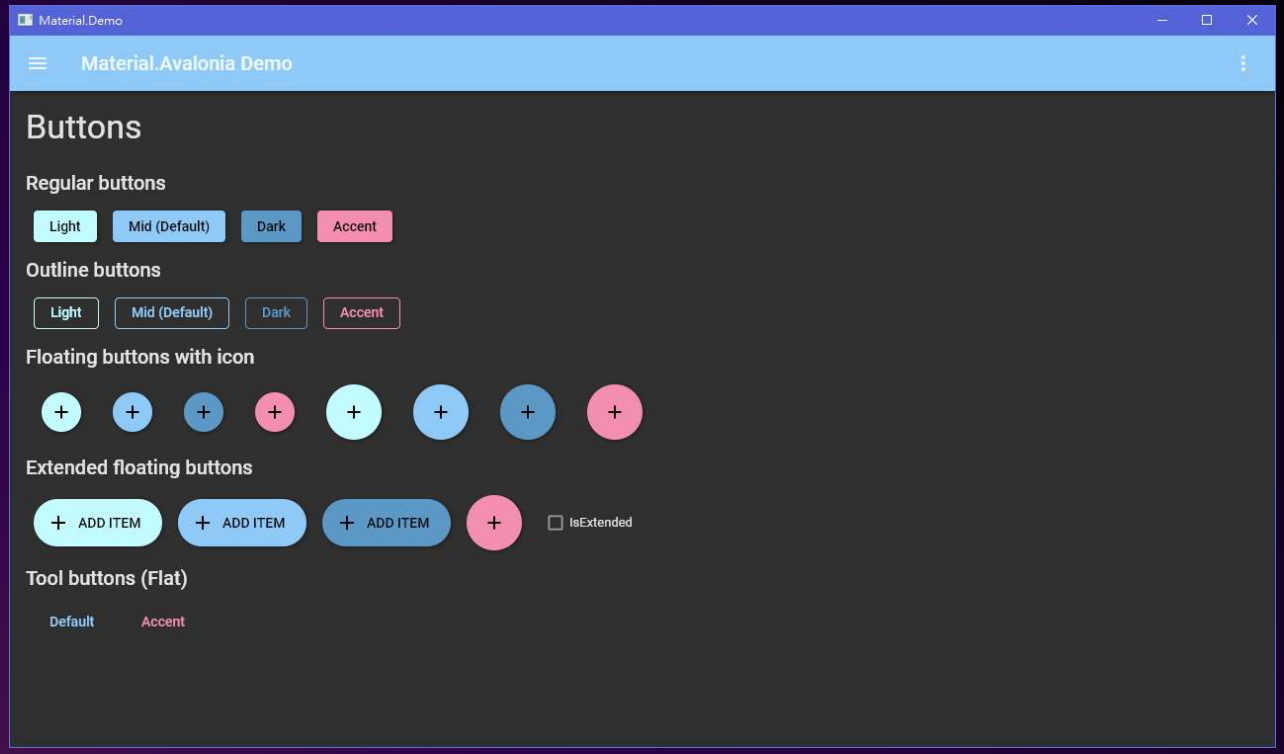
ОСНОВНЫЕ ВОЗМОЖНОСТИ

Simple на примере приложения:



ОСНОВНЫЕ ВОЗМОЖНОСТИ

Material

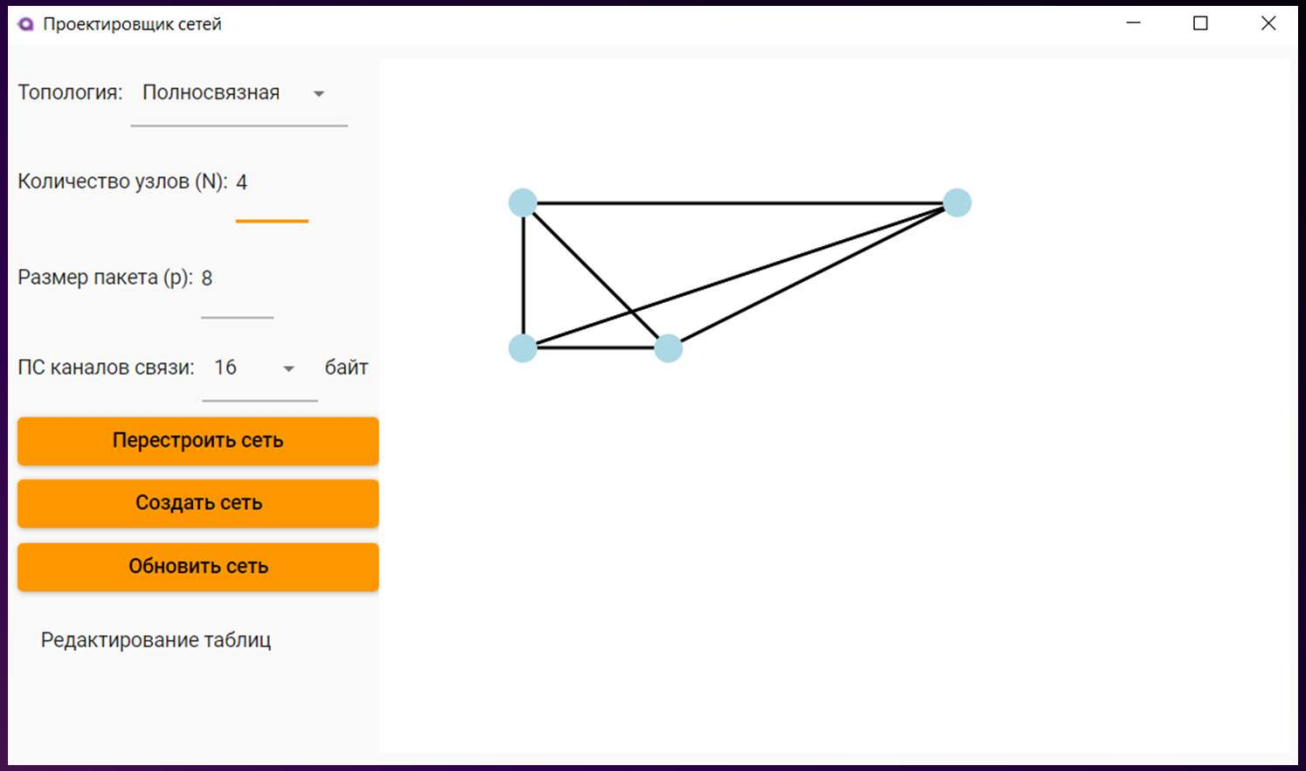


Мощная система стилей и тем (Fluent, Simple, Material).

Material основан на открытых стилях Гугл для создания удобных и красивых интерфейсов.

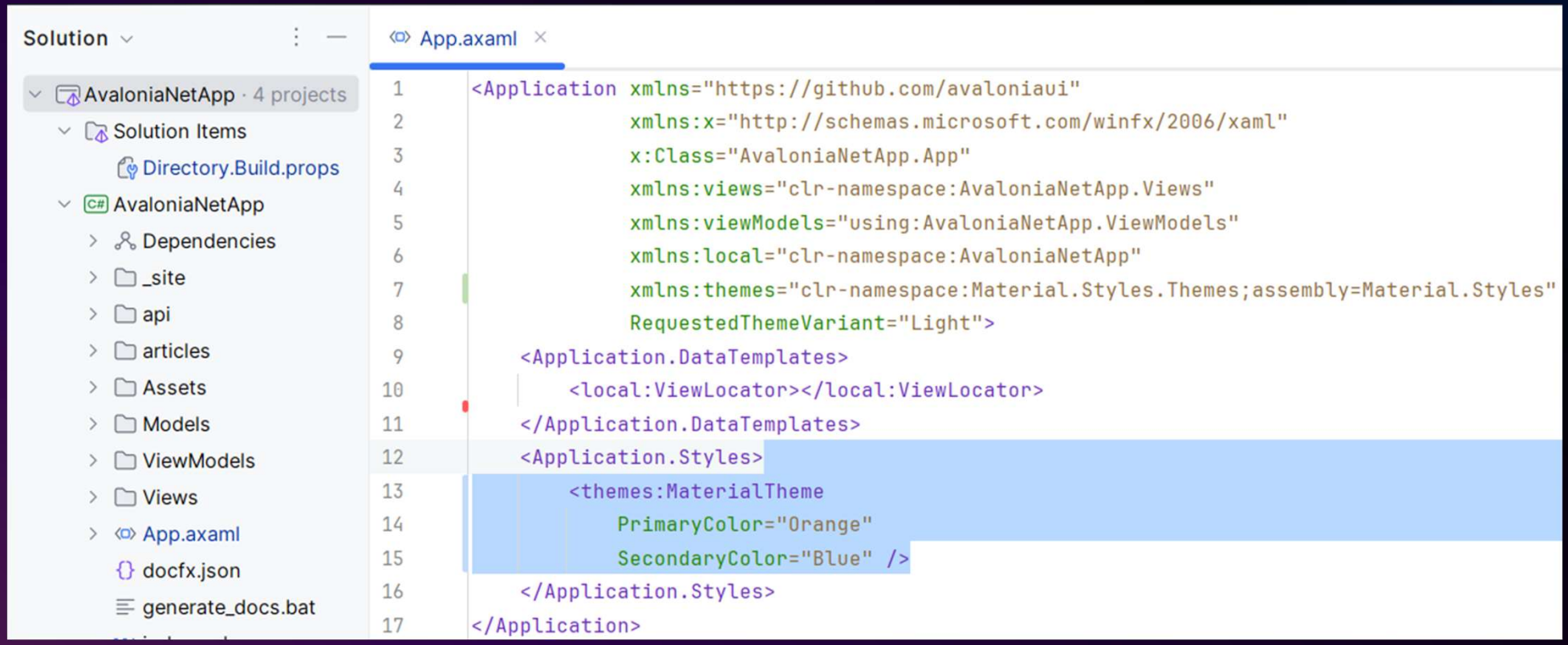
ОСНОВНЫЕ ВОЗМОЖНОСТИ

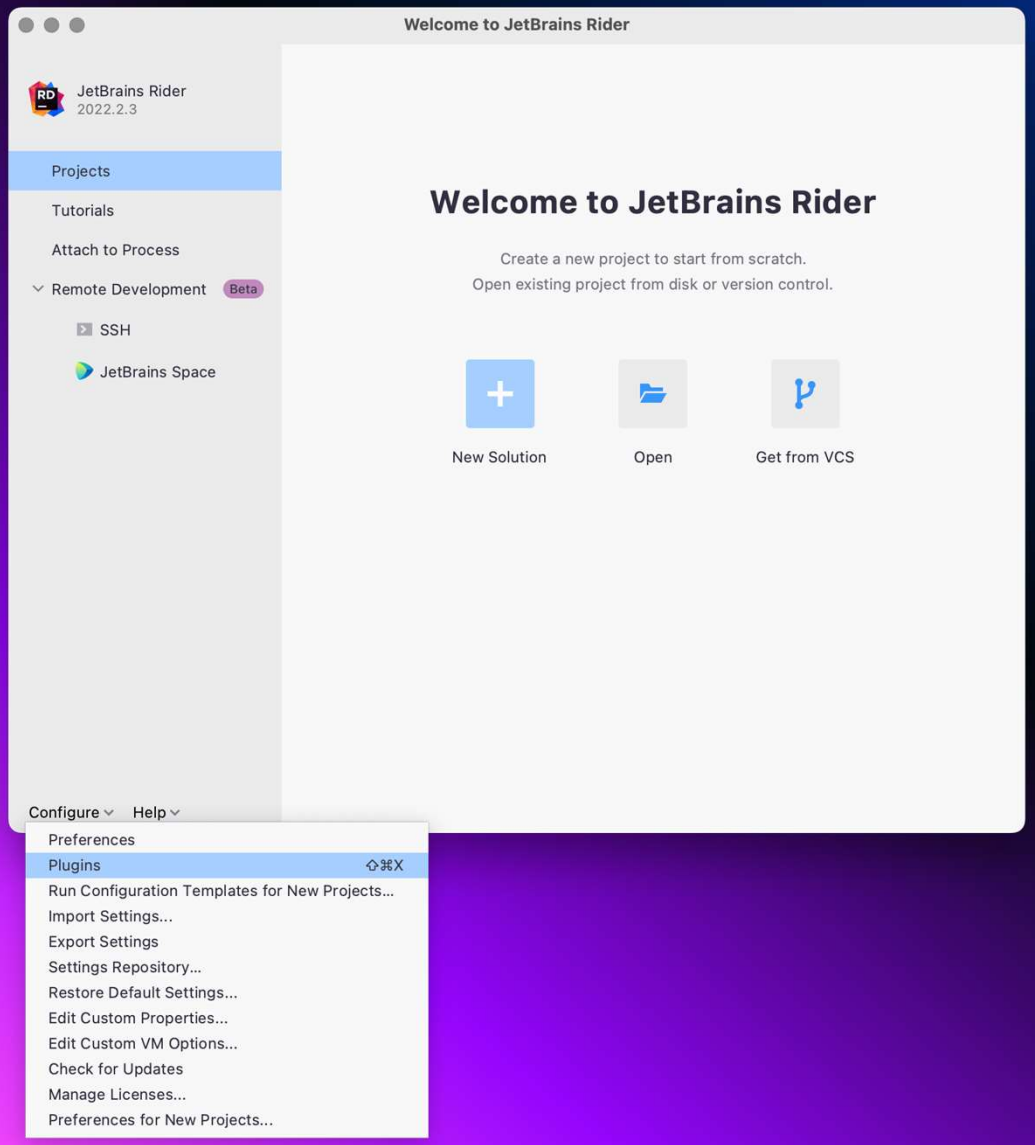
Material на примере приложения:



ОСНОВНЫЕ ВОЗМОЖНОСТИ

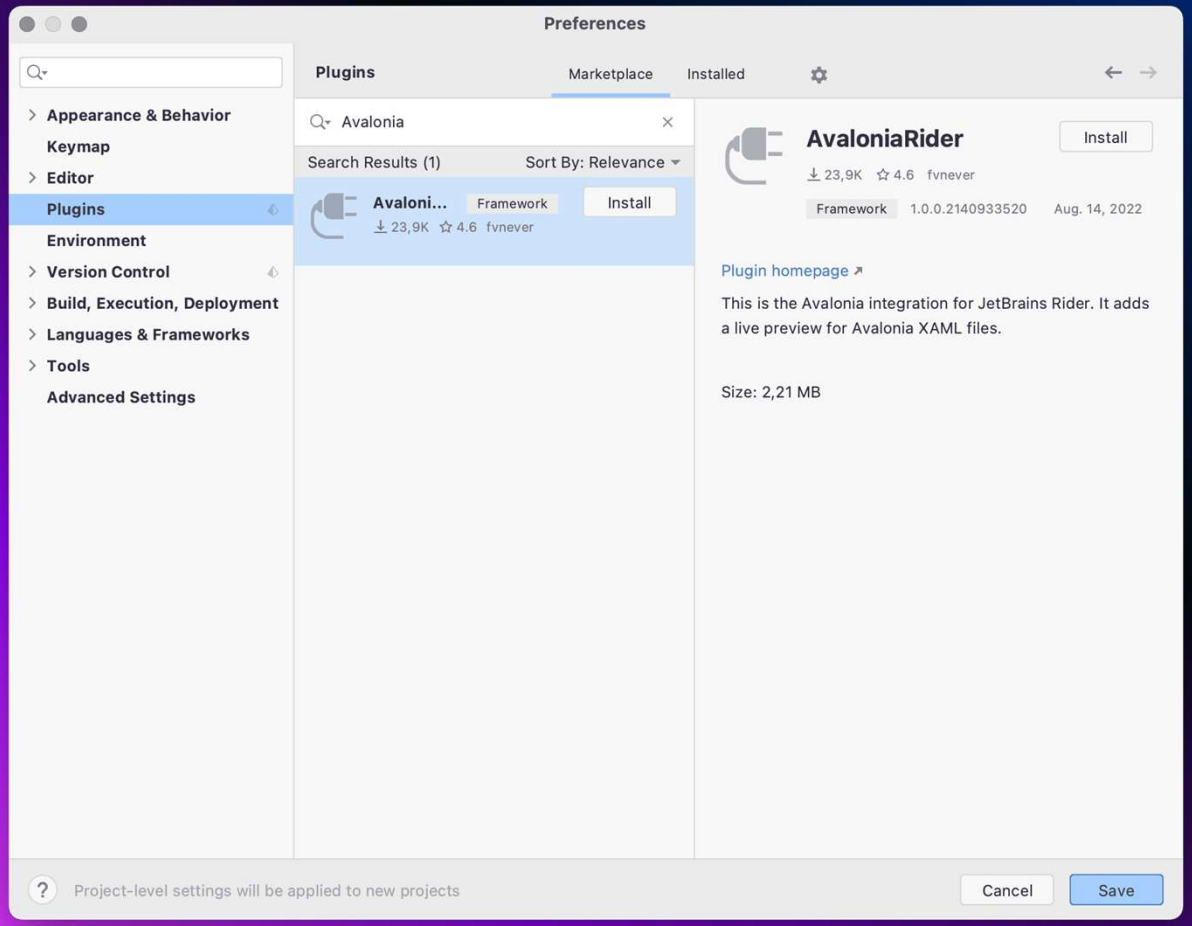
Цвет темы Material задан через специальные параметры





ОСНОВНЫЕ ВОЗМОЖНОСТИ

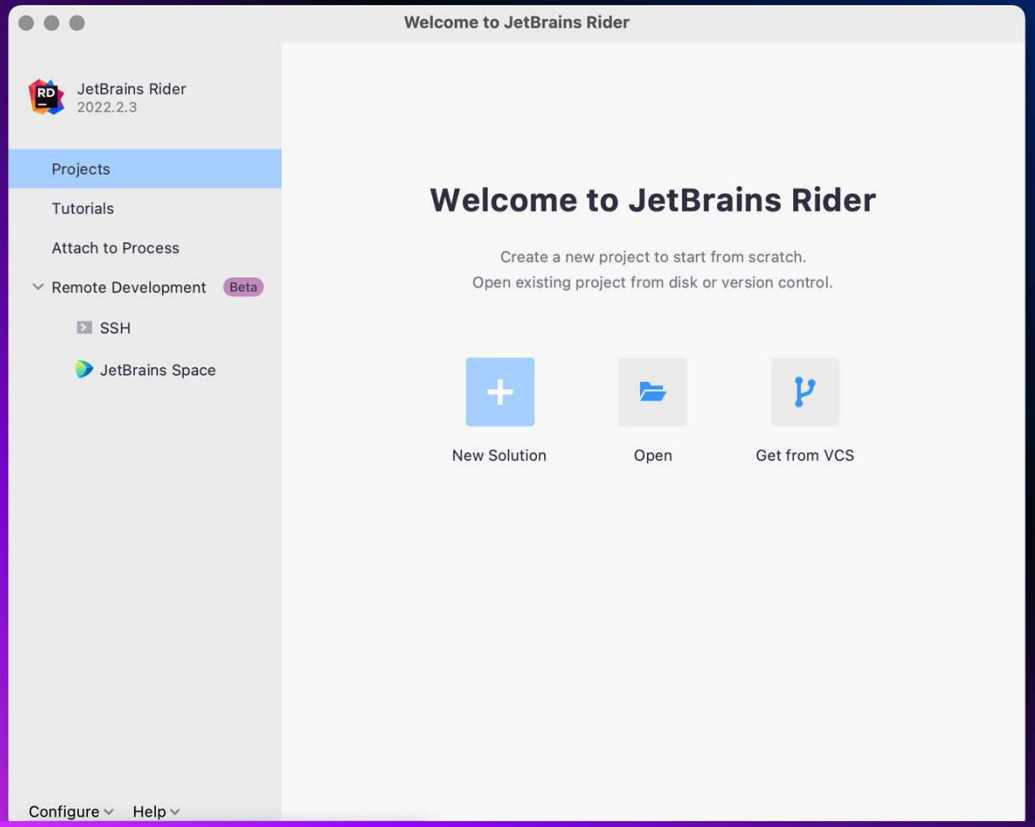
Интеграция с Rider.
Для создания проекта в данной IDE надо открыть Конфигурация>Плагины



ОСНОВНЫЕ ВОЗМОЖНОСТИ

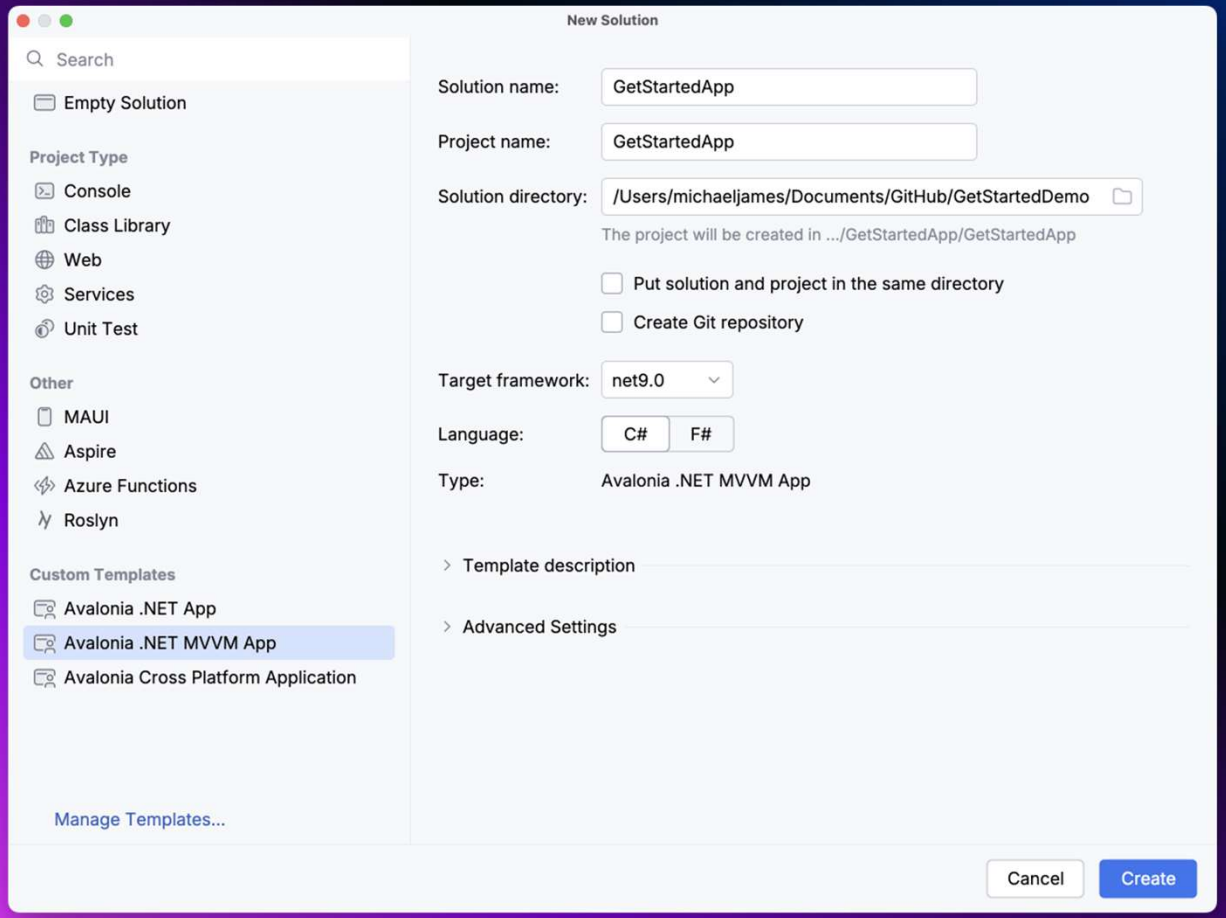
Интеграция с Rider.
Написать в поиск AvaloniaRider и скачать плагин.

После завершения установки нажмите кнопку «Перезапустить IDE».



ОСНОВНЫЕ ВОЗМОЖНОСТИ

Интеграция с Rider.
После перезагрузки можно нажать на “+” чтобы создать новый проект.

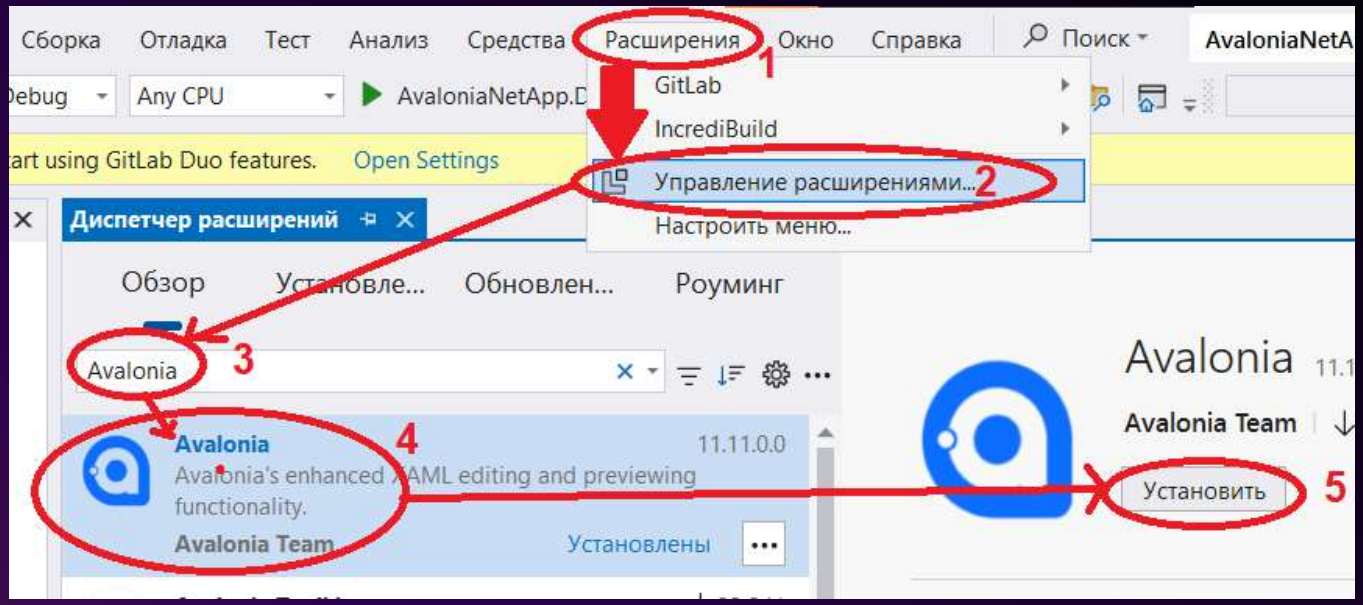


ОСНОВНЫЕ ВОЗМОЖНОСТИ

Интеграция с Rider.
В списке выбрать нужный вариант архитектуры Авалонии в списке слева и создать проект.

(Про Visual Studio аналогично)

ОСНОВНЫЕ ВОЗМОЖНОСТИ



Интеграция с Visual Studio.
Для создания проекта в данной IDE надо:

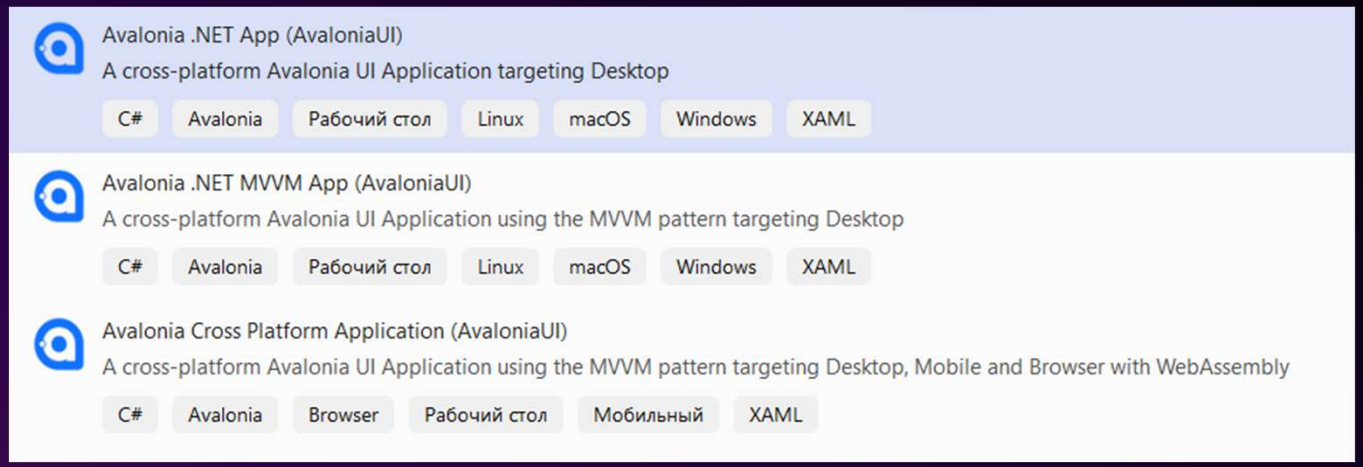
1-2. открыть Расширения>Управление расширениями

3. В поиске ввести “Avalonia”

4. Выбрать первый вариант

5. Установить

ОСНОВНЫЕ ВОЗМОЖНОСТИ



Интеграция с Visual Studio.
Теперь можно просто создать проект Avalonia с одним из трёх шаблонов.

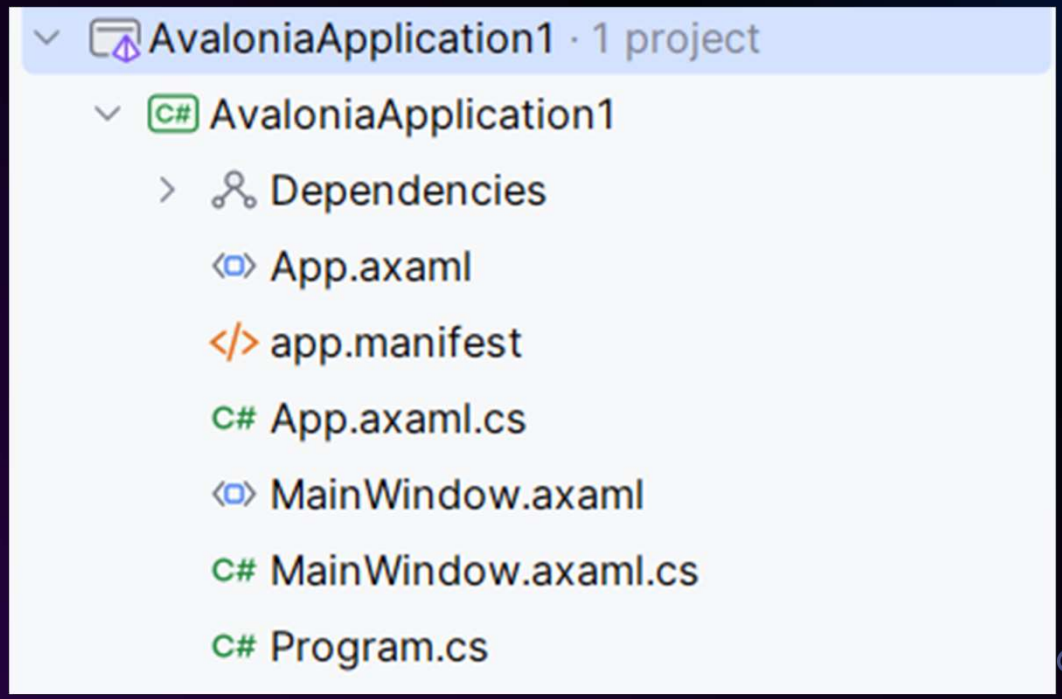
АРХИТЕКТУРА: ШАБЛОНЫ

Avalonia .NET App

Минимальный базовый шаблон.
Подходит, когда нужно «чистое» приложение без заранее навязанной архитектуры.

Особенности:

1. Только инфраструктура Avalonia (App.axaml, MainWindow.axaml).
2. **Нет MVVM**, нет дополнительных библиотек.
3. Создаётся как стартовая точка для ручной сборки архитектуры.
4. Максимальная гибкость — всё пишешь сам.



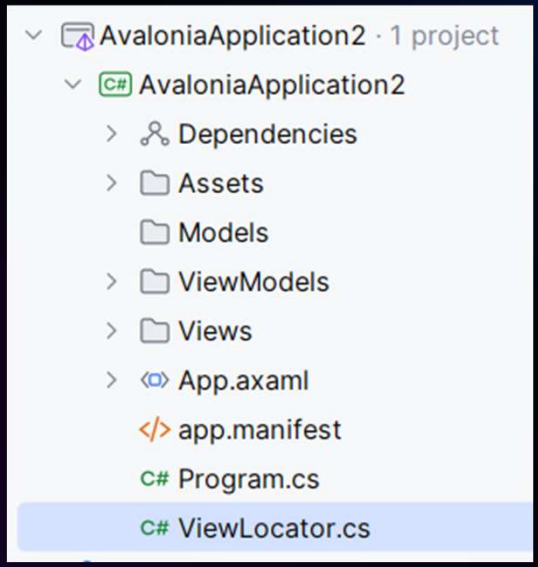
АРХИТЕКТУРА: ШАБЛОНЫ

Avalonia .NET MVVM App

Готовый шаблон с архитектурой MVVM.
Подходит для типичных приложений сразу с разделением View / ViewModel.

Особенности:

- Реализован MVVM «из коробки».
- Добавлены базовые ViewModels, привязки, команды.
- Использует **один из двух официальных MVVM-подходов Avalonia** (обычно CommunityToolkit.Mvvm).
- Логика и представление изначально разделены.



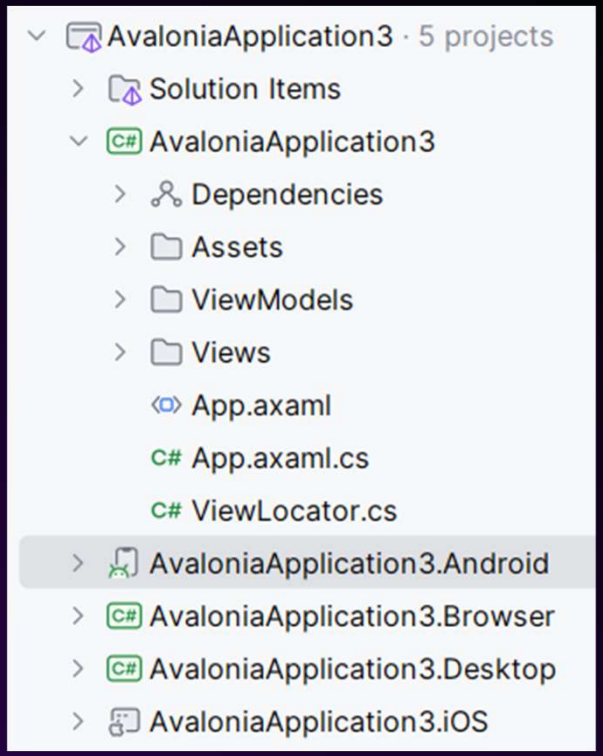
АРХИТЕКТУРА: ШАБЛОНЫ

Avalonia Cross Platform Application

Расширенная версия базового шаблона с фокусом на мультиплатформенности.

Особенности:

1. Более подробная структура проекта под Windows, Linux, macOS, Android, iOS.
2. Уже настроены платформенные стартовые точки (например, AppBuilder для разных ОС).
3. Может содержать заготовки под MVVM (но не всегда включает полноценный MVVM-набросок как MVVM-App).
4. Оптимальный выбор, если цель — выпуск на разные платформы.



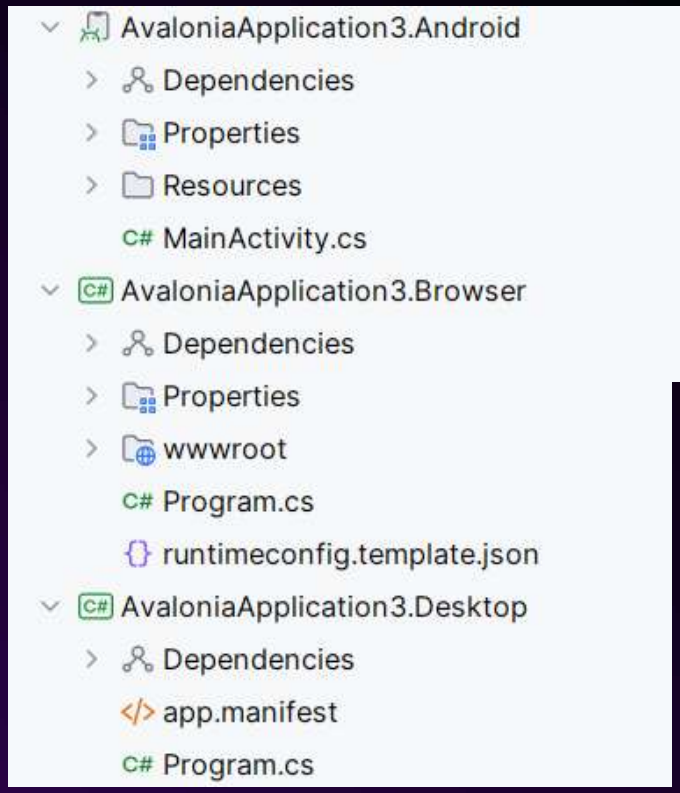
АРХИТЕКТУРА: ШАБЛОНЫ

Avalonia Cross Platform Application

Расширенная версия базового шаблона с фокусом на мультиплатформенности.

Особенности:

1. Более подробная структура проекта под Windows, Linux, macOS, Android, iOS.
2. Уже настроены платформенные стартовые точки (например, AppBuilder для разных ОС).
3. Может содержать заготовки под MVVM (но не всегда включает полноценный MVVM-набросок как MVVM-App).
4. Оптимальный выбор, если цель — выпуск на разные платформы.



АРХИТЕКТУРА: ИНСТРУМЕНТЫ

Инструмент	Простыми словами	Для кого	Сложность
CommunityToolkit.Mvvm	«Пиши как обычно, а мы сделаем за тебя рутину»	Новичков и обычных приложений	Лёгкий
ReactiveUI	«Всё — поток событий, UI сам реагирует»	Сложные, динамичные приложения	Сложнее

АРХИТЕКТУРА: ИНСТРУМЕНТЫ

ReactiveUI

```
using ReactiveUI;
using System.Reactive.Linq;

public class PersonViewModel : ReactiveObject
{
    private int _age;
    public int Age
    {
        get => _age;
        set => this.RaiseAndSetIfChanged(ref _age, value);
    }

    public bool IsAdult => _isAdult.Value;
    private readonly ObservableAsPropertyHelper<bool> _isAdult;

    public PersonViewModel()
    {
        // Реактивный поток: IsAdult обновляется автоматически
        this.WhenAnyValue(x => x.Age)
            .Select(age => age >= 18)
            .ToProperty(this, x => x.IsAdult, out _isAdult);
    }
}
```

CommunityToolkit

```
using CommunityToolkit.Mvvm.ComponentModel;

public partial class PersonViewModel : ObservableObject
{
    [ObservableProperty]
    private int age;

    public bool IsAdult => Age >= 18;
}
```

АРХИТЕКТУРА: MVVM

MVVM — Model–View–ViewModel

Архитектурный паттерн для разделения логики и интерфейса.

View

UI — окна, кнопки, разметка.
Ничего не знает о логике.

ViewModel

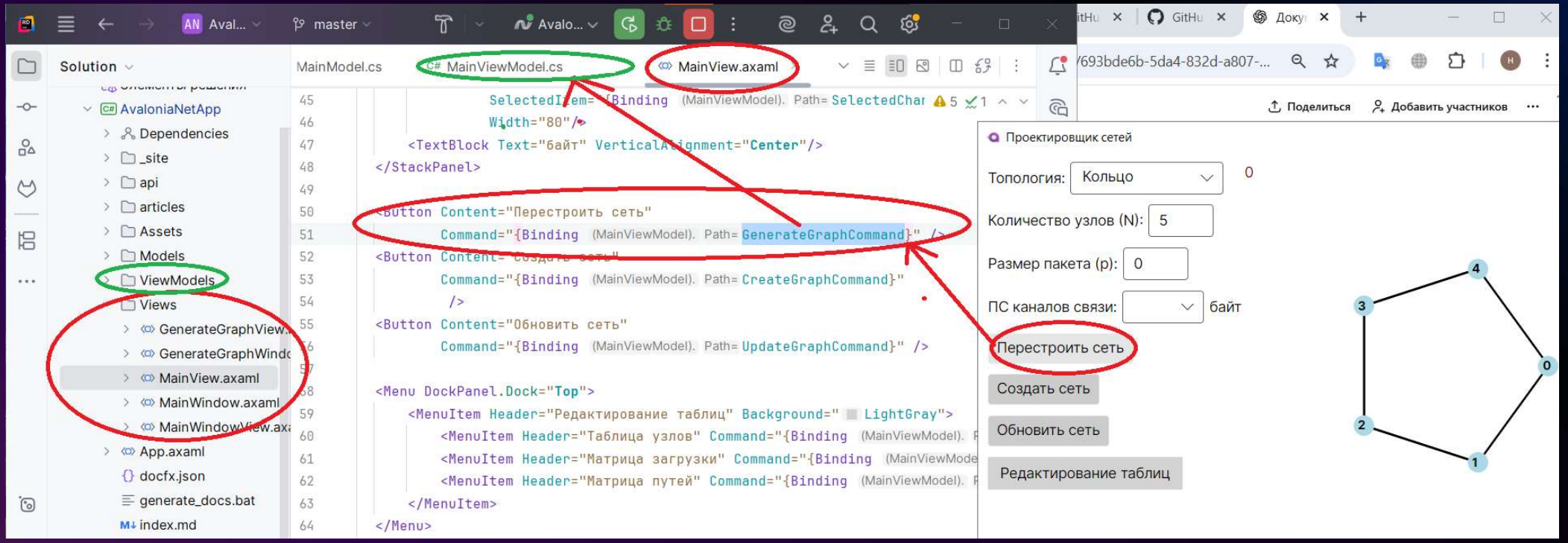
Логика: данные, команды, состояния.
Содержит свойства и команды, ккоторым привязывается View (binding).
Не содержит UI-кода.

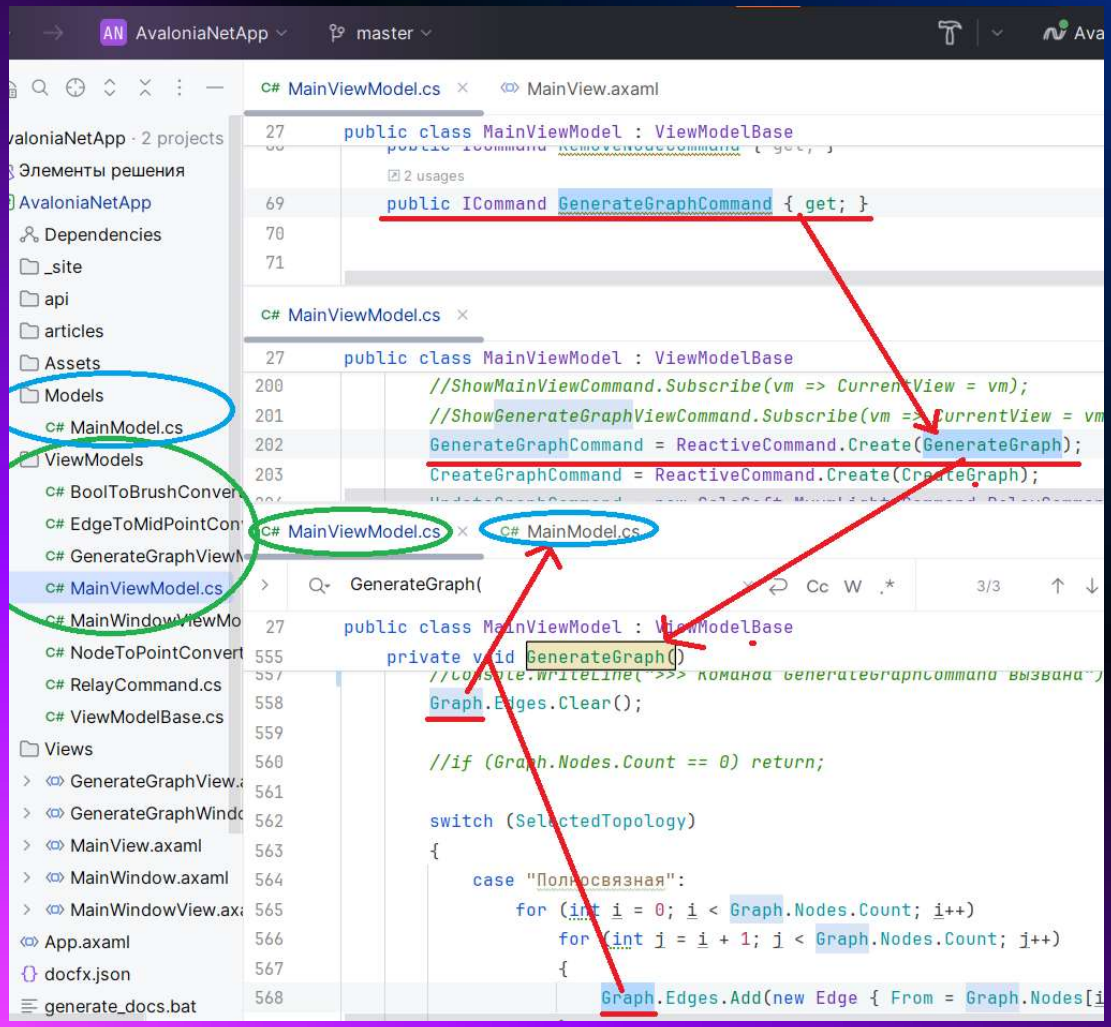
Model

Бизнес-данные и работа с ними: структуры данных, сервисы, доступ к БД, API.

АРХИТЕКТУРА: MVVM

View пример: Кнопка “Перестроить сеть” вызывает команду GenerateGraphCommand



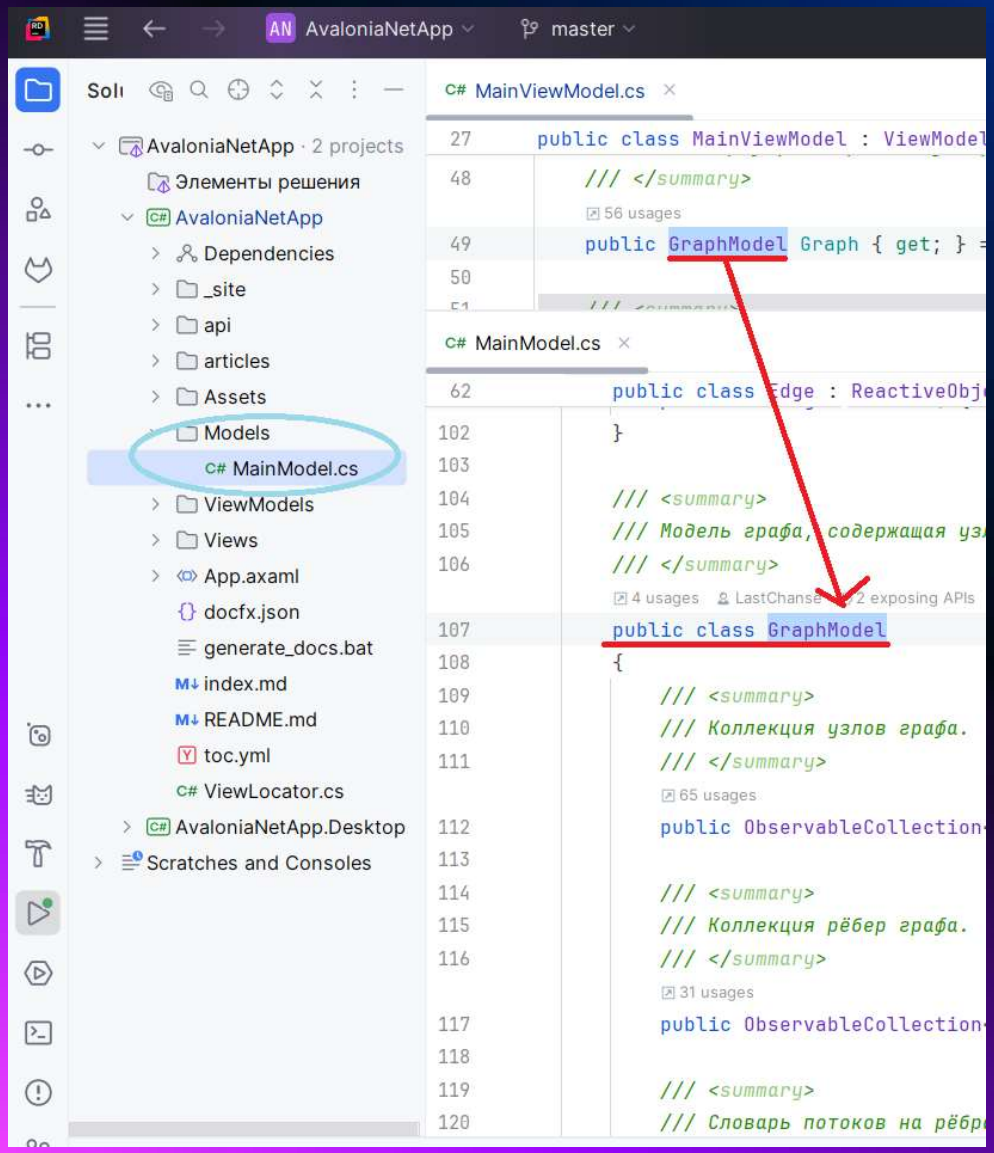


АРХИТЕКТУРА: MVVM

ViewModel пример:

GenerateGraphCommand Задаётся в модели представления для доступа из представления.

К команде привязывается метод GenerateGraph() который использует Graph связанный с моделью графа сети.



АРХИТЕКТУРА: MVVM

Model пример:

Graph связанный с моделью графа сети GraphModel который описывает модель в виде списка узлов и рёбер.

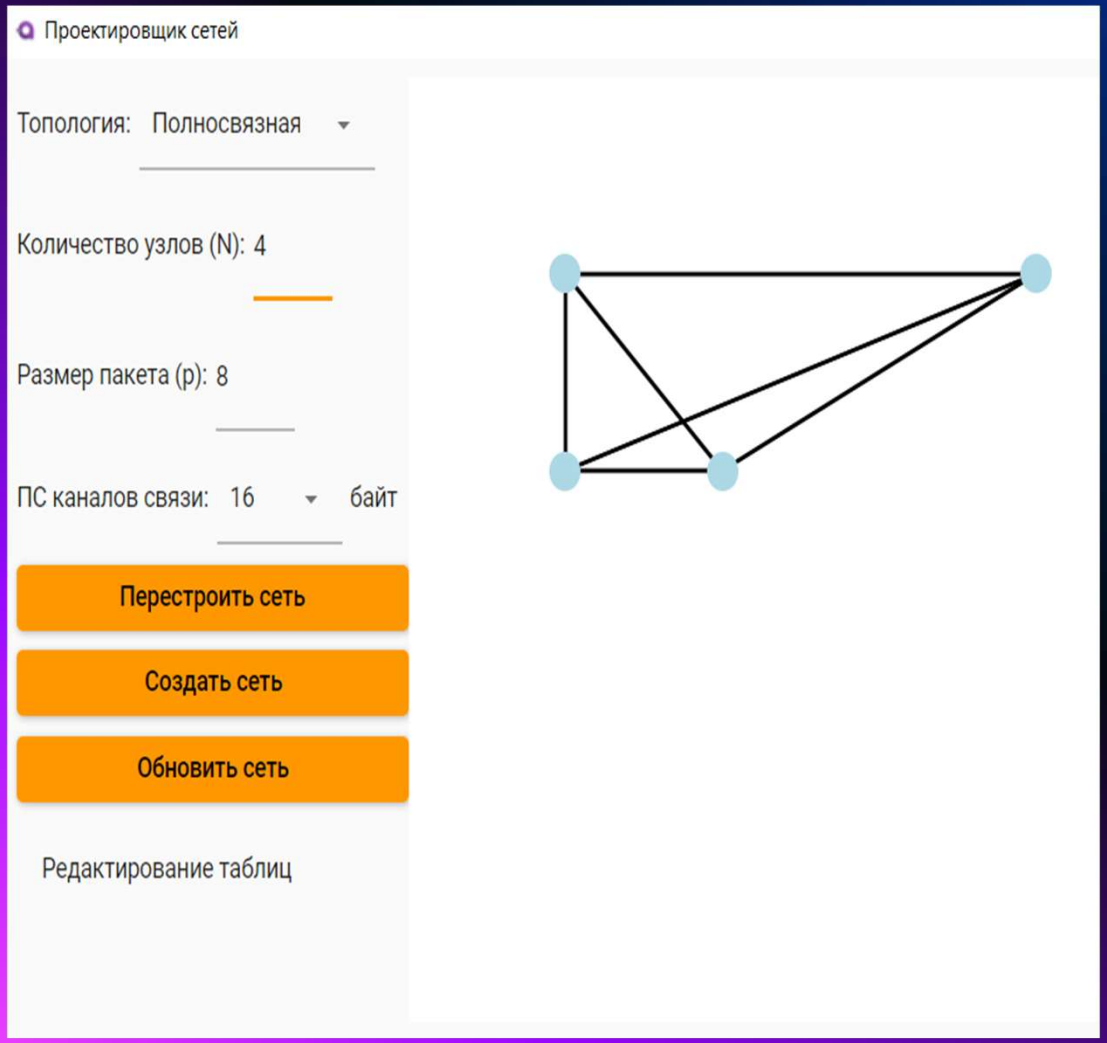
ПРЕИМУЩЕСТВА И НЕДОСТАТКИ

Недостатки:

- Молодая экосистема по сравнению с WPF, и как следствие недостаток обучающих материалов(нет учебников) и форумов для разработчиков авалония.
- Не все компоненты и SDK пока доступны на мобильных платформах.

Преимущества:

- Кроссплатформенность.
- Привычный XAML и MVVM
- Гибкая стилизация и темы
- Открытый исходный код
- Активное сообщество



ЗАКЛЮЧЕНИЕ

Avalonia UI — перспективный фреймворк для создания кроссплатформенных .NET-приложений. Он объединяет знакомый XAML-подход, гибкий дизайн и поддержку всех ОС. Благодаря активному сообществу и открытому коду, Avalonia быстро развивается и становится серьёзной альтернативой WPF и MAUI.

СПАСИБО

Благодарю за внимание!

Источники:

https://docs.avaloniaui.net/

https://github.com/AvaloniaUI/Avalonia.Samples/

https://git.kpfu.ru/GAKutasov/avalonianetapp