



- ling□ lamda
- C#□□
- EF□□□□□□□□□□
- Maui□□□□

linq lamda

```
// 0 < -1  
"".IndexOf("", StringComparison.OrdinalIgnoreCase)
```

C#

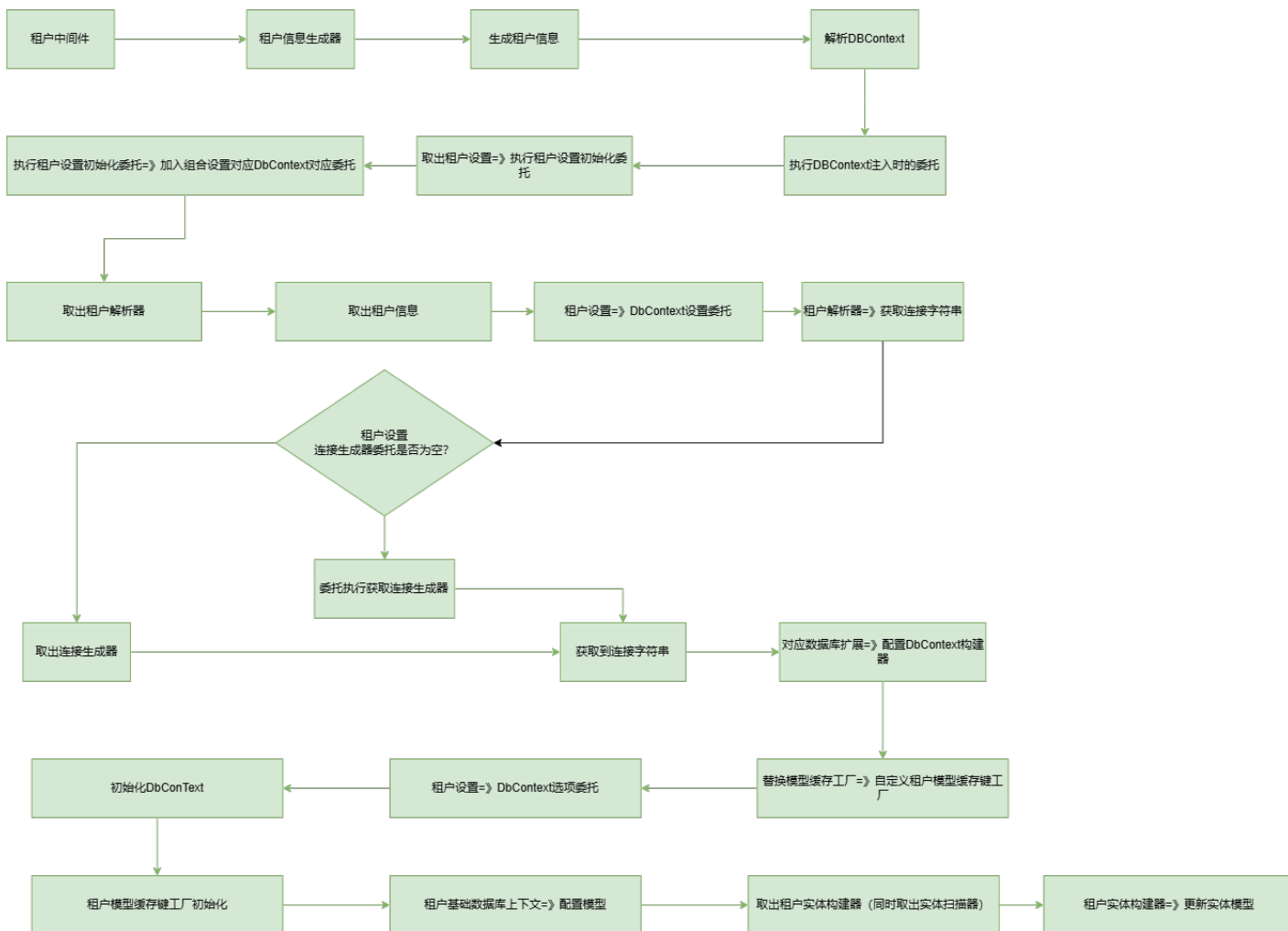
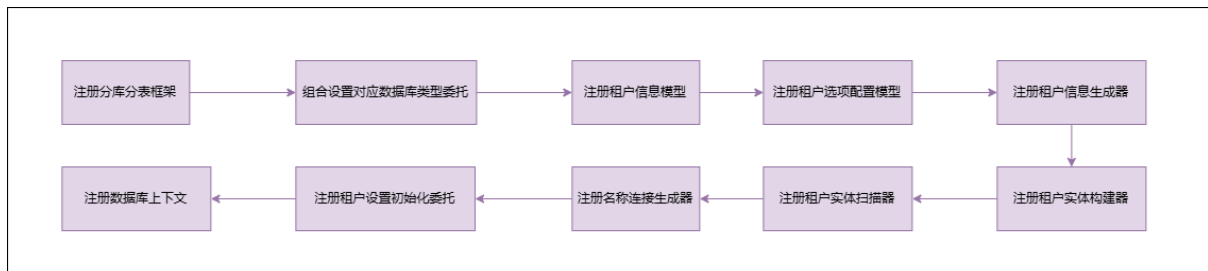
□□□□□□□□

```
public class FileStatus {
[DllImport("kernel32.dll")]
private static extern IntPtr _lopen(string lpPathName, int iReadWrite);
[DllImport("kernel32.dll")]
private static extern bool CloseHandle(IntPtr hObject);
private const int OF_READWRITE = 2;
private const int OF_SHARE_DENY_NONE = 0x40;
private static readonly IntPtr HFILE_ERROR = new IntPtr(-1);
public static int FileIsOpen(string fileFullName)
{
if (!File.Exists(fileFullName))
{
return -1; // □□□□
}

IntPtr handle = _lopen(fileFullName, OF_READWRITE | OF_SHARE_DENY_NONE);
if (handle == HFILE_ERROR)
{
return 1; // □□□□
}

CloseHandle(handle);
return 0;
}
}
```

EF



Maui

MauiProgram

```
public static class MauiProgram
{
    public static MauiApp CreateMauiApp()
    {
        var builder = MauiApp.CreateBuilder();
        builder.UseMauiApp<App>();
        builder.ConfigureFonts(fonts =>
        {
            fonts.AddFont("OpenSans-Regular.ttf", "OpenSansRegular");
            fonts.AddFont("OpenSans-Semibold.ttf", "OpenSansSemibold");
        });
        builder.UsePrism(prism =>
        {
            prism.ConfigureServices(service =>
            {
                RestClientOptions restClientOptions = new()
                {
                    //restClientOptions.RemoteCertificateValidationCallback += (sender, cert, chain, error) => true;
                    BaseUrl = new Uri("https://dynamicapi.lingyanspace.com"),
                    MaxRedirects = 5,
                    Timeout = TimeSpan.FromMinutes(3),
                    ThrowOnAnyError = true
                };

                RestClient restClient = new(restClientOptions);
                service.AddSingleton(restClient);
                service.AddSingleton<ICourseService, CourseService>();
            });
            prism.RegisterTypes(container =>
            {
                //  MainPage
                container.RegisterForNavigation<MainPage>();
                container.RegisterForRegionNavigation<HomeView, HomeViewModel>();
            });
        });
    }
}
```

```

});

prism.OnInitialized((container) =>
{
    var regionManager = container.Resolve<IRegionManager>();
    regionManager.RegisterViewWithRegion("MainRegion", nameof(HomeView));
});

//□□□□□

prism.CreateWindow(navigationService =>
navigationService.NavigateAsync($"/{nameof(MainPage)}"));

});

#if DEBUG
    builder.Logging.AddDebug();
#endif

return builder.Build();
}
}

```



```

<?xml version="1.0" encoding="utf-8" ?>
<ContentPage
    x:Class="LearnMirro.MainPage"
    xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    xmlns:prism="http://prismlibrary.com"
    Title="MainPage"
    Background="#eee">
    <ContentView prism:RegionManager.RegionName="MainRegion" />
</ContentPage>

```