



WPFWPF



WPFWPF

1. 多线程

- 1. 多线程的概念
- 2. 多线程的优缺点
- 3. 多线程的创建方式
- 4. 多线程的同步与互斥
- 5. 多线程的通信
- 6. 多线程的异常处理
- 7. C# 多线程的编程模型
- 8. Thread 与 Task 的区别
- 9. 多线程的调度策略
- 10. WPF 多线程的编程模型
- 11. UI 多线程的编程模型
- 12. Dispatcher 与 UI 线程

2. 并行编程

- 1. Span<T> 与 Span<T>.OfT 的区别
- 2. CancellationToken 与 CancellationTokenSource
- 3. WPF 多线程的编程模型

3. 网络编程

- 1. TcpListener 与 TcpClient 的区别
- 2. SignalR 与 SignalR 客户端
- 3. WPF 多线程的编程模型

4. 异步编程

- 1. Monitor/Mutex, Semaphore 与 Dispatcher 与 BackgroundWorker
- 2. async/await 与 TaskScheduler
- 3. TPL Dataflow 与 TPL Dataflow 的扩展
- 4. WPF 多线程的编程模型

5. 性能优化

1 TPL 线程池 线程池

2 线程池 线程池 Span<T> 内存池

3 WPF 线程池 线程池 WPF 线程池 UI 线程池

线程池

1 WebSocket 线程池 线程池 System.Net.WebSockets 线程池 WebSocket 线程池

2 SignalR 线程池 线程池 SignalR 线程池

3 线程池 线程池 gRPC 线程池 RESTful API 线程池

4 WPF 线程池 线程池 WPF 线程池 WebSocket 线程池 SignalR 线程池

线程池

1 线程池 线程池

2 WPF 线程池 线程池 UI 线程池

线程池

线程池

线程池

线程池

线程池 线程池

线程池

线程池

线程池 线程池

线程池

线程池

线程池 SignalR 线程池 WebSocket 线程池

WPF 线程池 线程池

线程池

线程池

线程池 线程池

线程池 SignalR 线程池

线程池 Dispatcher 线程池

□□□□ □

□□ TPL Dataflow □□□□□□

□□ TaskScheduler□□□□□□

□□ III□□□□□□□□□□

3.1 □□□□□

3.1.1 □□□□□

- □□ **ICollectionView**□□□□□□
- □□ **VisualStateManager**□□□□□□

3.1.2 □□□□□

- □□ **NLog**□□□□□ **Debug/Info/Error**□
- □□□□□□□□ **RBAC**□

3.2 □□□□□

3.2.1 □□□□□

- □□ **OxyPlot**□□□□□□
- □□ **WriteableBitmap**□□□□□□

□□ IV□□□□□□□□

4.1 □□□□□

4.1.1 .NET MAUI□□□□

- □□□□□□□□ **UI**□□□□
- □□ **SkiaSharp**□□□□□□

4.1.2 WPF□ Web□□□□

- □□ **WebView2**□□□□□□
- □□ **WebAssembly**□ **Blazor**□□

4.2 □□□□□

4.2.1 □□□□□□□

- □□ **ML.NET**□□□□□□
- □□ **ONNX**□□□□□□

4.2.2 □□□□□□□

- □□ **LiveCharts**□□□□□□

000	#12			
0 00	000	3 00	2025	20:43:22
0 00	000	2 00	2025	19:26:02