

Programação Assíncrona - Exercícios

- 1- O que é programação assíncrona e qual é a sua importância na programação moderna?
- 2- Como a programação assíncrona é implementada em C# usando as classes Task e Task<T>?
- 3- Como você pode usar a palavra-chave "async" para criar um método assíncrono em C# e a palavra-chave "await" para aguardar a conclusão de uma operação assíncrona?
- 4- Quais são as diferenças entre o modelo de programação síncrona e assíncrona em C#?
- 5- Qual é a diferença entre a classe Task e a classe ValueTask em C# ? Em quais cenários você pode preferir usar uma em vez da outra?
- 6- Como você pode usar a classe ValueTask para criar operações assíncronas eficientes em C#? Quais são as melhores práticas para o uso do ValueTask em código assíncrono?
- 7- Como você pode lidar com exceções em operações assíncronas em C#? Quais são as melhores práticas para lidar com exceções em código assíncrono?
- 8- Como você pode cancelar uma tarefa assíncrona em C#? Quais são as etapas necessárias para realizar o cancelamento de forma eficaz?
- 9 - Quais são as melhores práticas para o cancelamento de tarefas assíncronas em C#? Como você pode lidar com a liberação de recursos e a reversão de alterações feitas antes do cancelamento?
- 10- Como você implementaria o cancelamento de múltiplas exceções em uma operação assíncrona em C# usando AggregateException e InnerExceptions ?
- 11- Quais são as diferenças entre o cancelamento de exceções em uma operação assíncrona utilizando a classe CancellationToken e a utilização da exceção OperationCanceledException?