AULA #01

Desvendando o C# Moderno

.DOT LIVE

Bruno Roberto Olympio Ferreira - bruno.olympio.ferreira@gmail.com - IP: 187.107.71.41

Sobre a DotLive

Sobre a DotLive

- Série de aulas semanais sobre carreira e programação com .NET, com conteúdo de qualidade e atualizado
- Sempre nas quartas-feiras, às 20h, aqui no canal
- **As aulas ficam gravadas? Sim e não.**
  - Sim, amanhã a aula estará no canal ainda.
  - Porém, ficará disponibilizada por uma semana, e então ficará exclusiva a alunos do Método .NET e Imersão .NET Expert, e mentorados

Sobre o LuisDev

Luis Felipe (LuisDev)

Programador, Mentor, Instrutor

- Experiência como Desenvolvedor .NET Sênior para Brasil, Estados Unidos e Europa
- 2x Microsoft MVP (Developer Technologies)
- 9x Microsoft Certified
- Especialização em Arquitetura de Soluções
- Mais de 700 alunos e dezenas de mentorados

O que vamos ver hoje

O que veremos hoje?

- Conhecer alguns dos recursos apresentados desde o C# 7.0 até o C# 11.0
  - Tuples (C# 7)
  - Expressões Switch (C# 8)
  - Declarações using (C# 8)
  - Nullable Reference Types (C# 8)
  - Records (C# 9)
  - Modificador required (C# 11)

Tuples

Tuples

- As tuplas são um recurso do C# que permite retornar múltiplos valores de um método ou armazenar múltiplos valores em uma variável. Elas foram introduzidas no C# 7.0 e aprimoradas em versões posteriores.
- As tuplas são compostas por um conjunto ordenado de elementos, cada um com um nome opcional. É possível acessar os elementos através de seus índices ou de seus nomes, o que pode melhorar a legibilidade do código.
- As tuplas são mais leves do que as classes, o que significa que elas podem fornecer um melhor desempenho em cenários onde o desempenho é crítico. Elas também podem ajudar a melhorar a legibilidade do código em alguns casos, especialmente quando estamos trabalhando com um pequeno número de valores heterogêneos.

Tuples

- Outra vantagem das tuplas é que elas podem ser desestruturadas, ou seja, seus elementos podem ser atribuídos a variáveis separadas. Isso pode tornar o código mais limpo e legível, além de simplificar a lógica do programa.
- As tuplas também suportam valores nulos, o que é útil para lidar com situações em que um ou mais elementos podem estar ausentes. Para isso, é necessário usar os operadores de coalescência nula ou o operador de verificação de null.
- Por fim, as tuplas também podem ser usadas em expressões LINQ, o que permite agrupar e projetar dados com mais facilidade.

Demo: Tuples

Expressões switch

Expressões switch

- Expressões switch são uma forma simplificada e poderosa de expressar lógicas condicionais em C#.
- Introduzido no C# 8, permite que você escreva expressões condicionais com menos código e mais legibilidade.
- As expressões switch são uma alternativa às declarações switch.
    - As declarações switch têm um formato mais antigo e rígido, que exige o uso de **break** para cada case, tornando o código mais verboso e propenso a erros.
    - As expressões switch são mais flexíveis, pois permitem que você defina o valor de retorno diretamente em cada case, o que pode simplificar e agilizar o código.

Demo: Expressões switch

Declarações using

Declarações using

- O comando "using" pode ser utilizado para declarar variáveis e objetos que serão automaticamente descartados ao final do escopo em que foram criados. Antes, o "using" era utilizado somente para descartar objetos após o término de seu uso dentro do escopo.
- Agora, o escopo das variáveis declaradas com o "using" é limitado ao bloco em que foram declaradas. Logo, assim que o bloco de código é encerrado, a variável é automaticamente descartada.

Demo: Declarações using

Nullable Reference Types

Nullable Reference Types

- Nullable Reference Types é um recurso do C# 8.0 que permite que os desenvolvedores adicionem mais segurança e confiabilidade aos seus códigos ao identificar e evitar a possibilidade de exceções de referência nula.
- O recurso faz com que todas as referências de objeto sejam consideradas como não nulas, a menos que explicitamente marcadas como nulas. Isso significa que o compilador agora pode apontar potenciais problemas de referência nula durante a compilação, em vez de esperar até que a exceção ocorra em tempo de execução.

Nullable Reference Types

- Isso nos ajuda a detectar erros em nossos códigos antes mesmo de serem executados.
- Por exemplo, imagine um método que espera receber um objeto, mas a chamada do método passa um valor nulo. Antes do recurso Nullable Reference Types, isso só seria descoberto em tempo de execução, resultando em uma exceção de referência nula. Com o recurso, o compilador detectaria isso e apresentaria um aviso, permitindo que o desenvolvedor corrija o problema antes de executar o código.
- Outro exemplo: métodos que poderiam retornar um objeto nulo, como o SingleOrDefault, agora retornam um tipo de referência nulo, já avisando de potenciais exceções de referência nula

Demo: Nullable Reference Types

Records

Records

- Records é um recurso do C# 9 que oferece uma maneira mais fácil e concisa de definir tipos de dados imutáveis.

- Records são definidos usando a palavra-chave "record" e possuem uma sintaxe simplificada, onde os campos e propriedades podem ser definidos em uma única declaração. Além disso, records possuem construtores e métodos.

Records

- Uma das principais vantagens de usar records é a facilidade de compará-los uns com os outros. Isso é feito usando o operador "==", que compara os valores dos campos e propriedades, em vez de comparar as referências de objeto.
- Outra vantagem, derivada do ponto anterior, é a capacidade de utilizar records como Value Objects. Como eles são imutáveis e comparáveis, podem ser facilmente utilizados como chaves em dicionários ou como elementos em conjuntos, o que pode melhorar muito o desempenho e a eficiência do código.

Demo: Records

Modificador required

Modificador required

- Disponível no C# 11, indica que uma propriedade deve ser inicializada através de um inicializador de objetos
- Com isso, podemos criar requisitos de inicialização de propriedades usando inicializador de objetos, ao invés de construtores
- Construtores podem ser usados, mas precisam do atributo *[SetsRequiredMembers]*, que basicamente diz ao compilador que o construtor inicializa os membros necessários

Demo: Modificador required

Vamos concluir?

Lembretes

- Sempre nas quartas-feiras, às 20h
- **As aulas ficam gravadas? Sim e não.**
  - Sim, amanhã a aula estará no canal ainda.
  - Porém, ficará disponibilizada por uma semana, e então ficará exclusiva a alunos do Método .NET e Imersão .NET Expert, e mentorados
- Compartilhe seu feedback sobre as aulas nas redes sociais com a hashtag *#dotlive-luisdev*, me marcando
- Já deixe agendado o horário das 20h de suas quartas-feiras
- Vamos que vamos!