

Curso ATDEV200 - Programação | Orientada a Objectos (OOP) em C#

5,00 Dia(s) - 30,00 Horas

Introdução

Este curso tem como destinatários quem pretenda desenvolver os seus conhecimentos na linguagem de programação C# (CSharp) e na Framework Microsoft .NET.

Neste curso serão apresentados diferentes tipos de aplicações: consola, WPF, ASP.NET MVC e ASP.NET Core. Também será abordado o tema do Entity Framework, para explorar as formas de aceder facilmente a bases de dados.

Público-alvo

Quem pretenda aprender uma nova linguagem de programação ou consolidar alguns conceitos de C#.

Quando completar o curso

Os participantes ficarão familiarizados com a sintaxe da linguagem C# e os conceitos de programação orientada a objetos .NET, como: classes, objetos, herança, polimorfismo, etc. A Entity Framework será também um dos temas abordados na formação.

Pré-requisitos

Os participantes desta ação não necessitam de experiência em C#, no entanto é recomendado um conhecimento de programação em qualquer outra linguagem imperativa.

Exames

(não existem exames)

Conteúdo em detalhe

Visão geral do .NET

O .NET Framework foi projetado para ser usado no desenvolvimento de vários tipos de aplicações. Este módulo proporciona uma visão geral do que é .NET e das aplicações que se podem construir e os vários “flavours” no .NET, como .NET Standard e .NET Core.

- O que é uma aplicação .NET?
- O Common Language Runtime e as bibliotecas de classes .NET
- .NET, .NET Core, .NET Standard, Xamarin / Mono
- Aplicações .NET: MVC, Web API, WPF, UWP

Aplicações .NET

Ao construir aplicações em .NET, é necessário seguir algumas regras. Veremos a estrutura de uma aplicação e quais partes são necessárias durante o tempo de execução.

- A solução e o projeto do Visual Studio
- O que são namespaces e assemblies

Classes e Objetos

Uma das partes essenciais na programação orientada a objetos (OOP) é criar classes e na execução das aplicações criar objetos a partir destas. Neste módulo, falaremos sobre os fundamentos da programação .NET. Criamos variáveis para armazenar números, texto e objetos complexos e tentamos usar operadores. Exploraremos as formas pelas quais se cria uma classe e adiciona estado e comportamento aos objetos.

- O que é uma classe?
- Campos, Métodos e Propriedades
- Namespaces

Construtores de instância

Sempre que instancia um objeto de uma classe, ele é construído. Este módulo irá analisar como o .NET gere esse processo.

- Construtores de instância
- Inicializadores de construtor
- “Overload” de Construtores

Tipos em .NET

Para armazenar números, texto ou objetos, precisamos de variáveis. Essas variáveis serão guardadas memória local. Sabendo onde na memória cada tipo de variável é armazenado, podemos prever o que acontecerá com cada objeto.

- Tipo de categorias em .NET
- Tipos de valor, tipos de referência e tipos de referência imutáveis
- Tipos de valores personalizados: estrutura e enumeração
- Boxing e Unboxing
- Argumentos de funções. Passagem por valor ou por referência; Parâmetros de Saída

Membros estáticos e classes estáticas

Por vezes necessitamos de uma funcionalidade que não faz parte de um objeto, mas sim de uma classe completa. Este processo permite o uso de métodos sem criar um objeto ou armazenar dados que podem ser acedidos por todos os objetos. Em .NET designamos este processo por acesso a membros e a classes estáticas.

- Membros estáticos
- Construtores estáticos
- Classes estáticas

Herança

Um aspeto muito importante da OOP é a capacidade de herdar a funcionalidade de um conceito base. Exploraremos como podemos usar a herança para estender uma classe e adicionar funcionalidades.

- Conceitos gerais de Herança
- Níveis de acessibilidade: Público, Privado, Protegido e Interno

Polimorfismo

Além da herança na OOP, existe outro conceito interessante designado por polimorfismo. Este mecanismo permite criar implementações diferentes para tipos específicos, com base numa definição genérica de uma função.

- Herança e polimorfismo
- Substituição de métodos e propriedades
- Usar a classe base
- “Sealed” Classes, métodos e propriedades
- A classe de objeto
- “Override” dos métodos ToString, Equals e GetHashCode

Classes abstratas e interfaces

Os conceitos de classes abstratas e interfaces são muito importantes .NET.

- Classes abstratas
- Definição de interfaces
- Implementação de interfaces

Coleções

Armazenar elementos numa coleção é crucial em linguagens de programação. Este capítulo descreve o uso de coleções no .NET.

- Arrays em .NET
- Matrizes multidimensionais
- Propriedades do indexador

- Os ciclos: for, foreach e while

Genéricos

Em C#, podem considerar-se dois mecanismos separados ao escrever código reutilizável em diferentes tipos: herança e genéricos. A herança fornece capacidade de reutilização com o conceito de tipos base, os genéricos expressam a capacidade de reutilização fornecendo modelos que contêm “placeholders” dos tipos. Os genéricos, em comparação com a herança, podem aumentar a segurança dos tipos a usar e reduzir a necessidade de “casting” e “boxing” nas aplicações. Este módulo ilustrará como e quando usar genéricos nas aplicações.NET.

- Como usar genéricos
- Criar genéricos
- Restrições

Controlo de Erros e Exceções

Sempre que algo inesperado acontece numa aplicação é necessário que o programa lide com o problema de forma robusta. Com o controlo de Error e Exceções, podemos reagir apropriadamente quando esses problemas ocorrem.

- Exceções em .NET
- Os blocos try-catch-when-finally
- Criação de exceções personalizadas

Delegados e eventos

Um dos aspetos mais difíceis da programação .NET é ser capaz de armazenar funções numa variável. Isso é exatamente o que os delegados permitem realizar. Este conceito é usado com várias tecnologias .NET, como programação assíncrona, manipulação de eventos, LINQ. Os eventos são baseados em delegados, permitindo que a aplicação reaja a eventos, como um clique de botão.

- Criação e uso de delegados
- Delegados multicast
- Criação de eventos
- Usar eventos no UI

Funcionalidades de Idioma

Este módulo apresenta recursos interessantes incluídos no .NET relativos à localização.

- Tipo de inferência
- Métodos de extensão
- Métodos Anónimos

- Tipos anónimos
- Expressões Lambda

Aceder a bases de dados com LINQ e Entity Framework

O LINQ permite usar expressões de pesquisa que são muito semelhantes às pesquisas em SQL, mas que podem ser aplicadas a vários tipos de dados. Isso torna muito fácil certas operações, como a filtragem e classificação de dados. Neste módulo será também apresentado o Entity Framework para acesso a dados relacionais.

- Classe Enumerable
- Delegados Func
- Entity Framework
- Operações de pesquisa: from, where, order, group e select
- Conhecer o Entity Framework
- Conceitos básicos do Entity Framework