nagarro

C# & .NET

Agenda

C# & .NET

C# conventions

Naming conventions, coding guidelines

Useful C# elements

.NET in the industry

nagarro

C# and .NET

Intro

C# & .NET

In the current IT ecosystem, several .NET versions are still under support and used: .NET Framework 4.8, .NET 6.0, .NET 8.0

Each version of .NET supports up to a certain C# language version:

- .NET Framework 4.8 -> C# 7.0-7.3
- .NET 6.0 -> C# 10.0
- .NET 8.0 -> C# 12.0

Let's have a look over these versions and see what some of the updates are there from the C# 7.3 version and what contains in the .NET versions

C# 7.0-7.3

C# & .NET

- **Improved constraint for Generics, added ‘unmanaged’, ‘System.Enum’, ‘System.Delegate’ constraint**
- **Improvements to the ‘ref’ variables and parameters**
  - Example: using ‘in’ to ensure reference passing but no changes in the method
- **You can now compare Tuples**
- **You can now add ‘_’ to separate thousands, e.g. int number = 1_000_000**

C# 10.0

C# & .NET

- Global using directives, example: global using System;
- File-scoped namespace, example: namespace MyNamespace;
- ‘With’ expression, example: var x = obj1 with { Property1 = 5 }
- Property matching
  - Example: if (obj is { Property1: { NestedProperty: 42}}) { some code}
- Implicit using in SDK Projects

C# 12.0

C# & .NET

- Using Declarations in Blocks, example: using var resource = new Resource();
- Simplified literals collections, example: var strings = ["apple", "banana", "cherry"];

nagarro

Useful C# elements

nagarro

Demo

nagarro

C# conventions

C# Naming conventions

C# conventions

- Class names, method names, namespaces, and public properties – Pascal Case
  - Example: CustomOrder, CalculateTotal, NoOfEntries
- Private or internal fields, method parameters, and local variables – Camel Case
  - Example: totalAmount, orderNumber, tempValue
- Constants and readonly static fields – Uppercase with underscores
  - Example: MAX_RETRIES, DEFAULT_PATH
- Additional special suffix are added to classes performing important roles, like ‘Controller’, ‘EventHandler’, ‘Context’ and many others

C# layout conventions

C# conventions

- Indentation is recommended at 4 spaces per level (avoid using tabs)
- Use blank lines to separate method definitions, properties and regions within a class to enhance readability
- Files should contain only 1 class, interface, enum or struct
- The Namespace should reflect the folder structure
- Place ‘using’ directives outside of namespace declaration

C# coding conventions

C# conventions

- Null checks to be performed (where valid), using the conditional operators
  - Example: var result = data?.Length ?? 0;
- Try…Catch using specific exceptions instead of generic exception
- Prefer string interpolation over concatenation.
- Use Linq for simple querying and collection manipulation
- Use ‘async’ and ‘await’ for asynchronous actions
- Avoid extension methods as much as possible
- Avoid Magic Numbers

nagarro

.NET in the industry

.NET in the industry

- **Web programming**
  - REST API (.NET WEB API)
  - API + FullStack (React / Angular)
  - Entity Framework
  - Self-Hosted web services (kestrel)
- **Desktop programming**
  - WFP / MAUI / Windows Forms
  - Entity Framework
  - Windows Services

Thank you!