Functional C# 10

Putting C# Into Functional Perspective

Zoran Horvat

CEO at Coding Helmet

@zoranh75 https://codinghelmet.com

Version Check

This version was created by using:

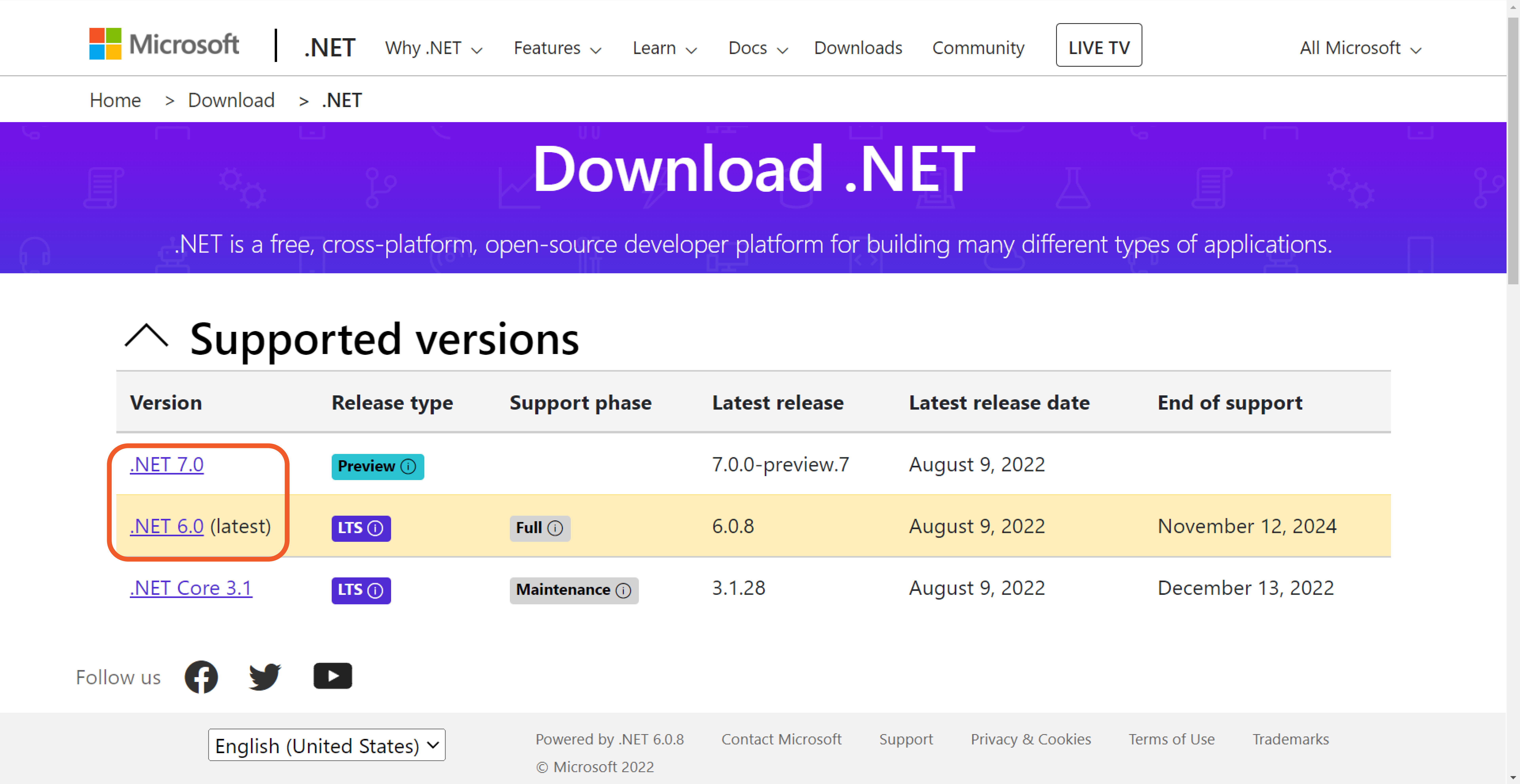
- .NET 6 & .NET 7
- C# 10 & C# 11
- Visual Studio Code 1.7

Relevant Notes

A note on frameworks and libraries:

- The `SkiaSharp` library used in this course is v2.88.0
- For latest information, refer to https://github.com/mono/SkiaSharp

https://dotnet.microsoft.com/en-us/download/dotnet



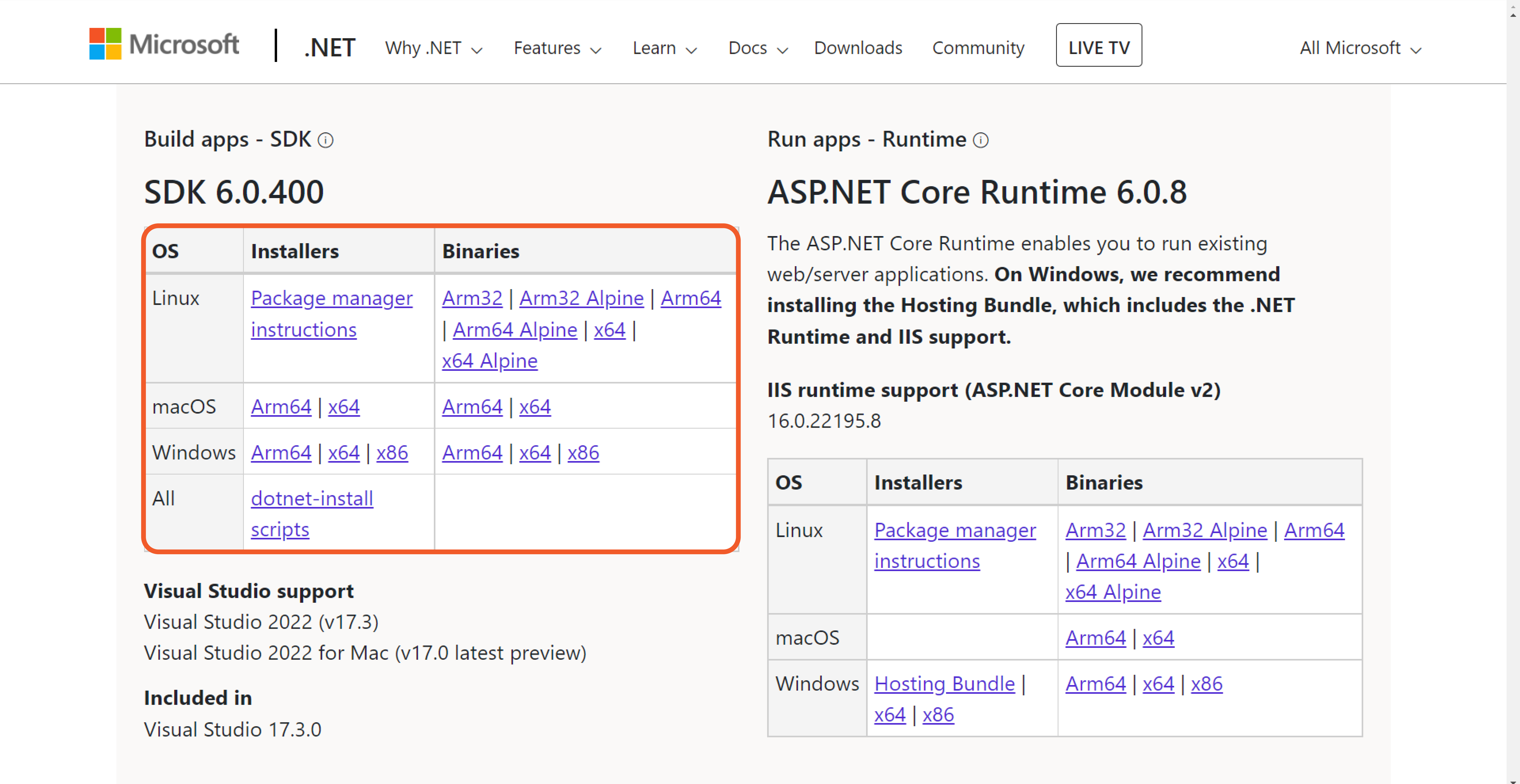
Download .NET

.NET is a free, cross-platform, open-source developer platform for building many different types of applications.

Supported versions

Version	Release type	Support phase	Latest release	Latest release date	End of support
.NET 7.0	Preview ⓘ		7.0.0-preview.7	August 9, 2022	
.NET 6.0 (latest)	LTS ⓘ	Full ⓘ	6.0.8	August 9, 2022	November 12, 2024
.NET Core 3.1	LTS ⓘ	Maintenance ⓘ	3.1.28	August 9, 2022	December 13, 2022

https://dotnet.microsoft.com/en-us/download/dotnet



Build apps - SDK ⓘ

SDK 6.0.400

OS	Installers	Binaries
Linux	Package manager instructions	Arm32 | Arm32 Alpine | Arm64 | Arm64 Alpine | x64 | x64 Alpine
macOS	Arm64 | x64	Arm64 | x64
Windows	Arm64 | x64 | x86	Arm64 | x64 | x86
All	dotnet-install scripts	

Visual Studio support
Visual Studio 2022 (v17.3)
Visual Studio 2022 for Mac (v17.0 latest preview)

Included in
Visual Studio 17.3.0

Run apps - Runtime ⓘ

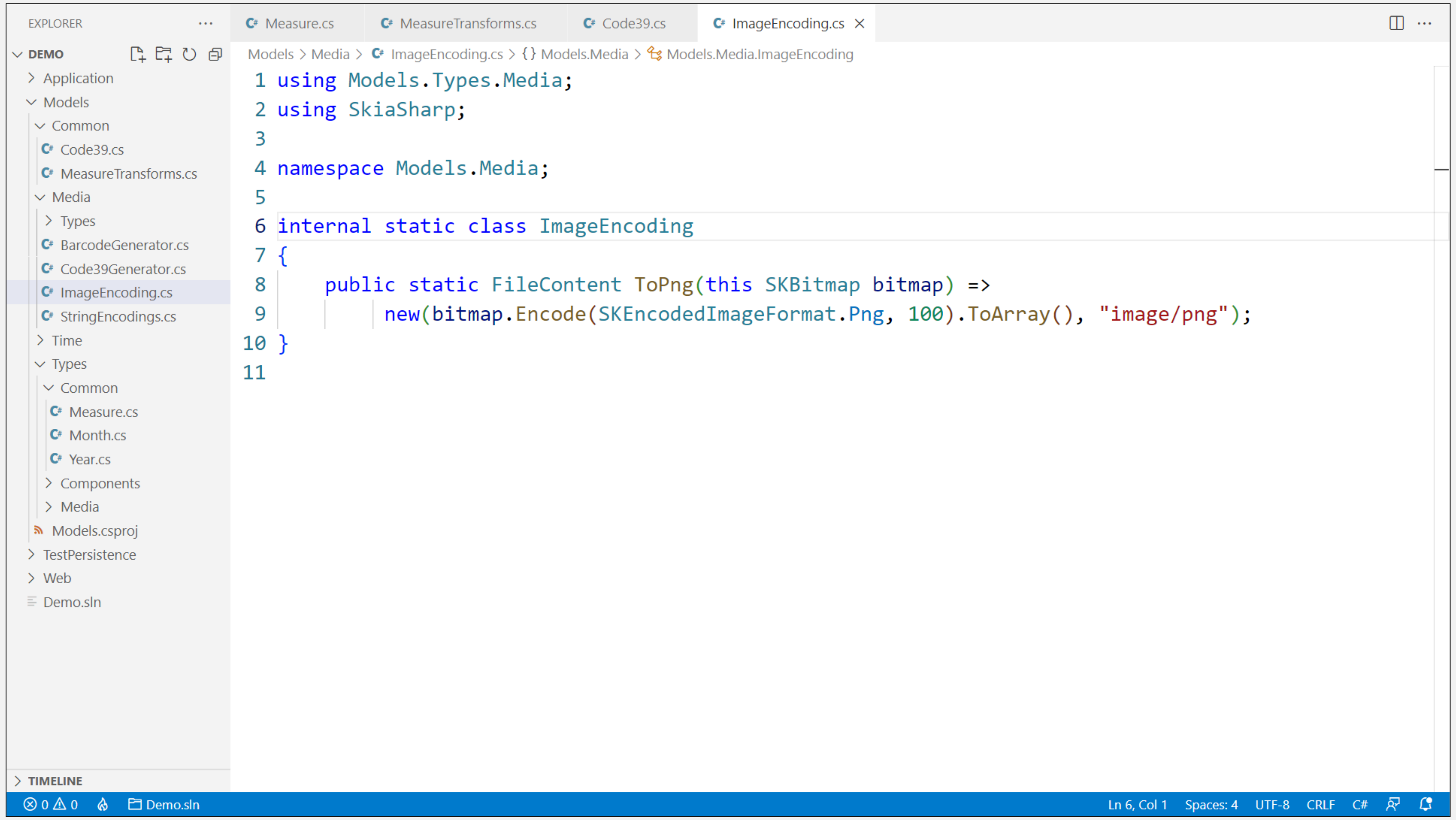
ASP.NET Core Runtime 6.0.8

The ASP.NET Core Runtime enables you to run existing web/server applications. **On Windows, we recommend installing the Hosting Bundle, which includes the .NET Runtime and IIS support.**

IIS runtime support (ASP.NET Core Module v2)
16.0.22195.8

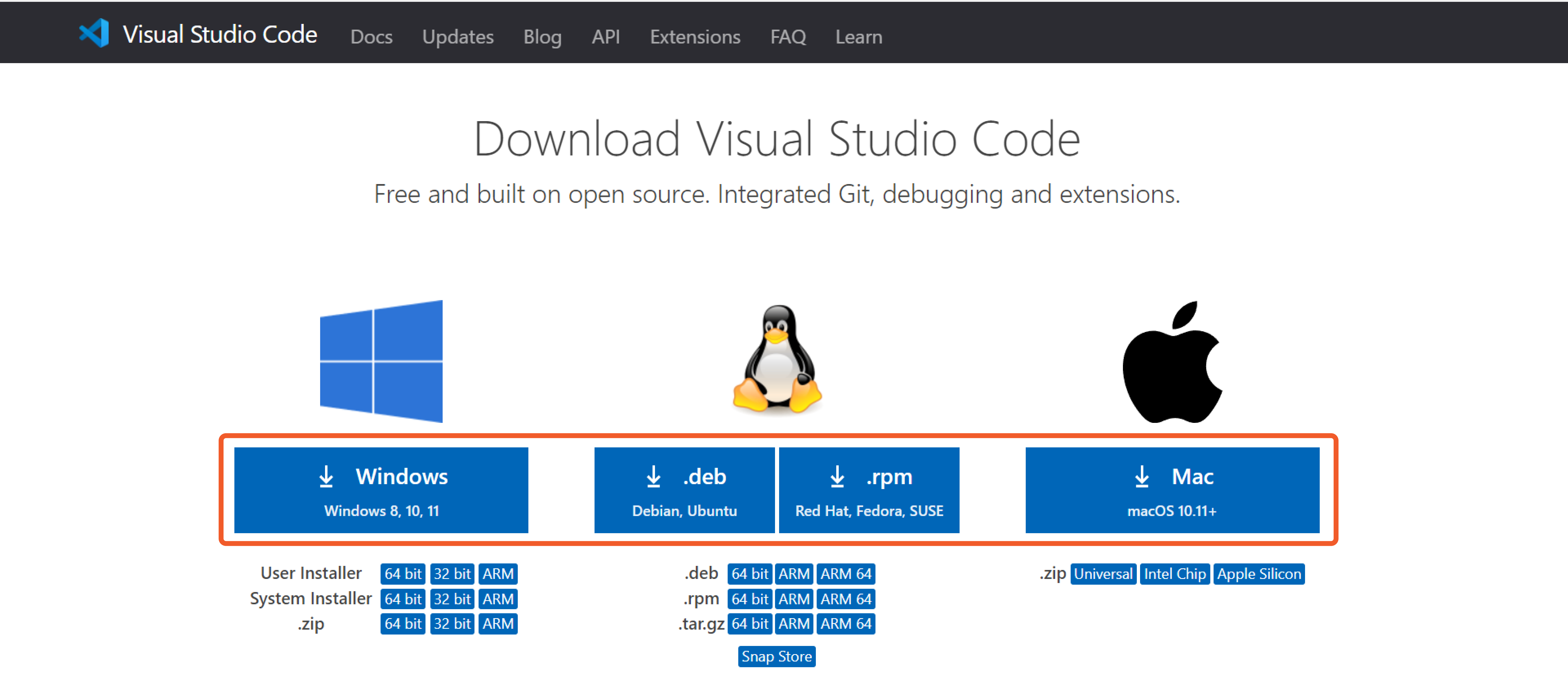
OS	Installers	Binaries
Linux	Package manager instructions	Arm32 | Arm32 Alpine | Arm64 | Arm64 Alpine | x64 | x64 Alpine
macOS		Arm64 | x64
Windows	Hosting Bundle | x64 | x86	Arm64 | x64 | x86

https://code.visualstudio.com/download



```csharp
using Models.Types.Media;
using SkiaSharp;

namespace Models.Media;

internal static class ImageEncoding
{
    public static FileContent ToPng(this SKBitmap bitmap) =>
        new(bitmap.Encode(SKEncodedImageFormat.Png, 100).ToArray(), "image/png");
}

```

https://code.visualstudio.com/download



Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.

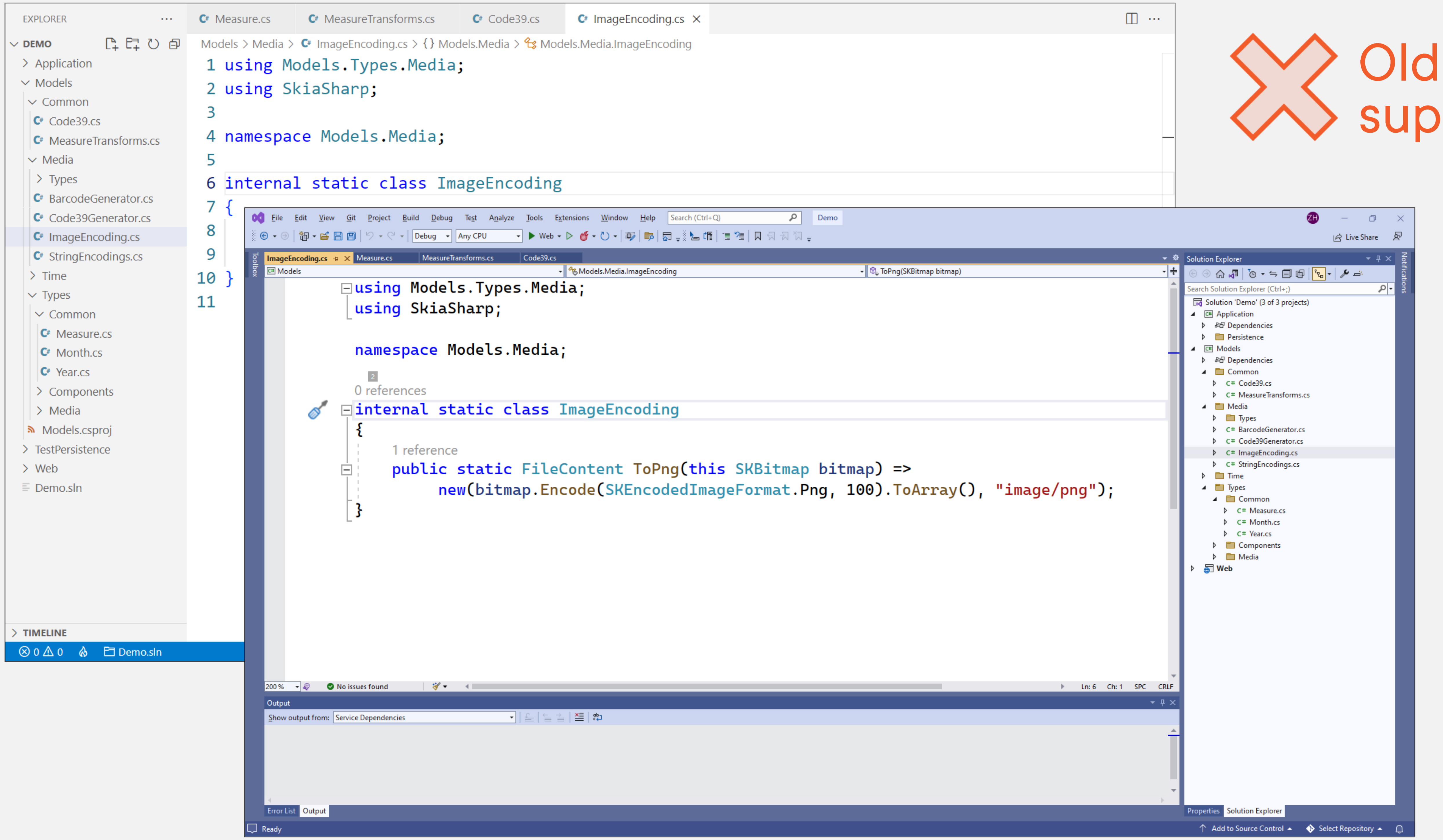
User Installer 64 bit 32 bit ARM
System Installer 64 bit 32 bit ARM
.zip 64 bit 32 bit ARM

.deb 64 bit ARM ARM 64
.rpm 64 bit ARM ARM 64
.tar.gz 64 bit ARM ARM 64
Snap Store

.zip Universal Intel Chip Apple Silicon

✓ Demos should run fine with future C# and .NET

✗ Older C# and .NET do not support most of the demos



```
using Models.Types.Media;
using SkiaSharp;

namespace Models.Media;

internal static class ImageEncoding
{


}
```

```
using Models.Types.Media;
using SkiaSharp;

namespace Models.Media;

internal static class ImageEncoding
{
    public static FileContent ToPng(this SKBitmap bitmap) =>
        new(bitmap.Encode(SKEncodedImageFormat.Png, 100).ToArray(), "image/png");
}
```

Language Features Used in This Course

Language feature/syntax	Version it was introduced in
Func/Action delegates	C# 2
Sequences (IEnumerable<T>)	
yield return/break	
LINQ, lambda expressions	C# 3
Extension methods	
Expression-bodied members	C# 6
Tuples, deconstruction	C# 7
Pattern matching	(and later)
switch expressions	C# 8
Ranges	
Nullable reference types	
Record types, init-only setters	C# 9
Record structs	C# 10

Assumptions

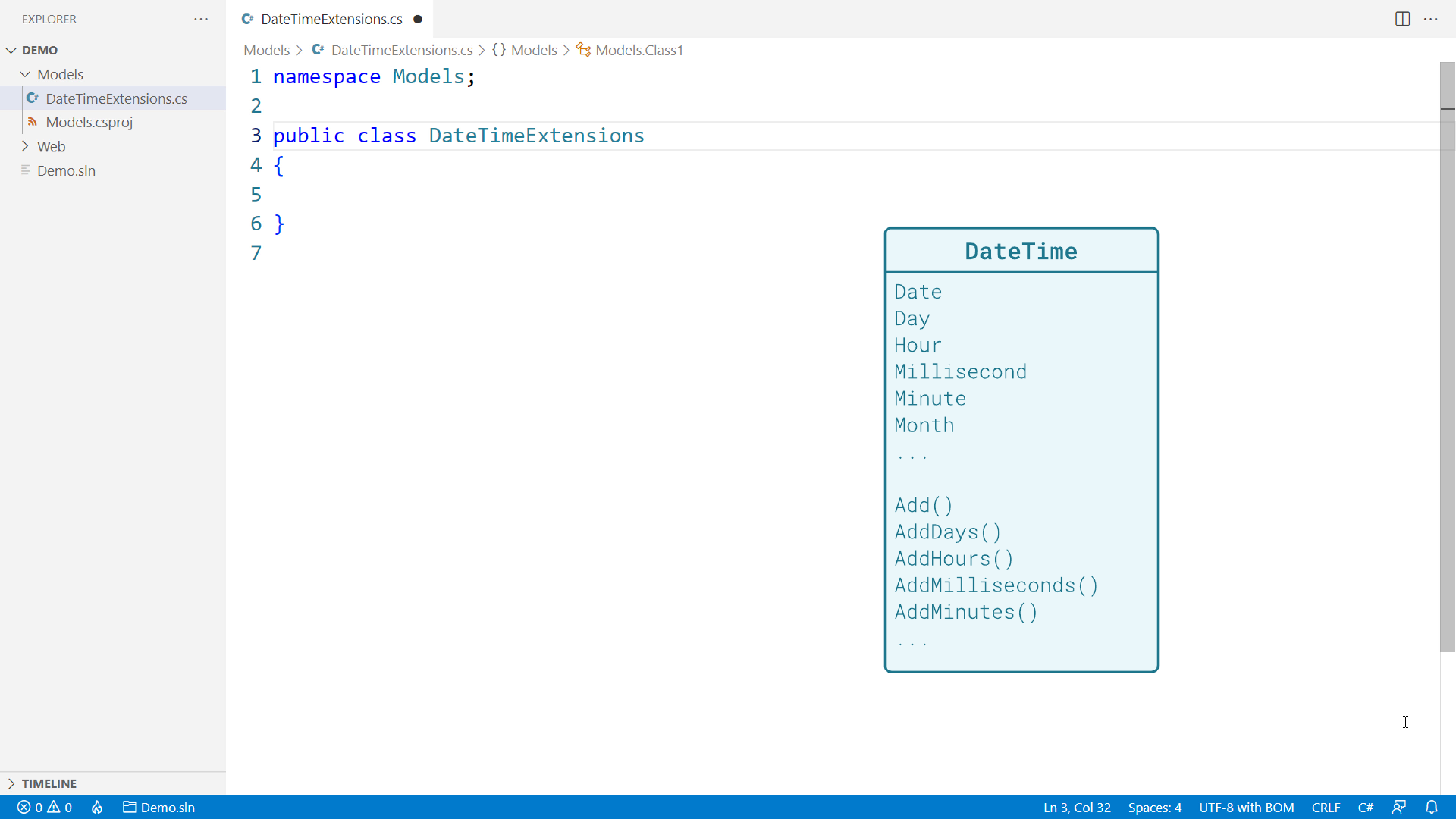
C# programming language
required

Functional programming
welcome

LINQ
required

Persistence & network
not covered

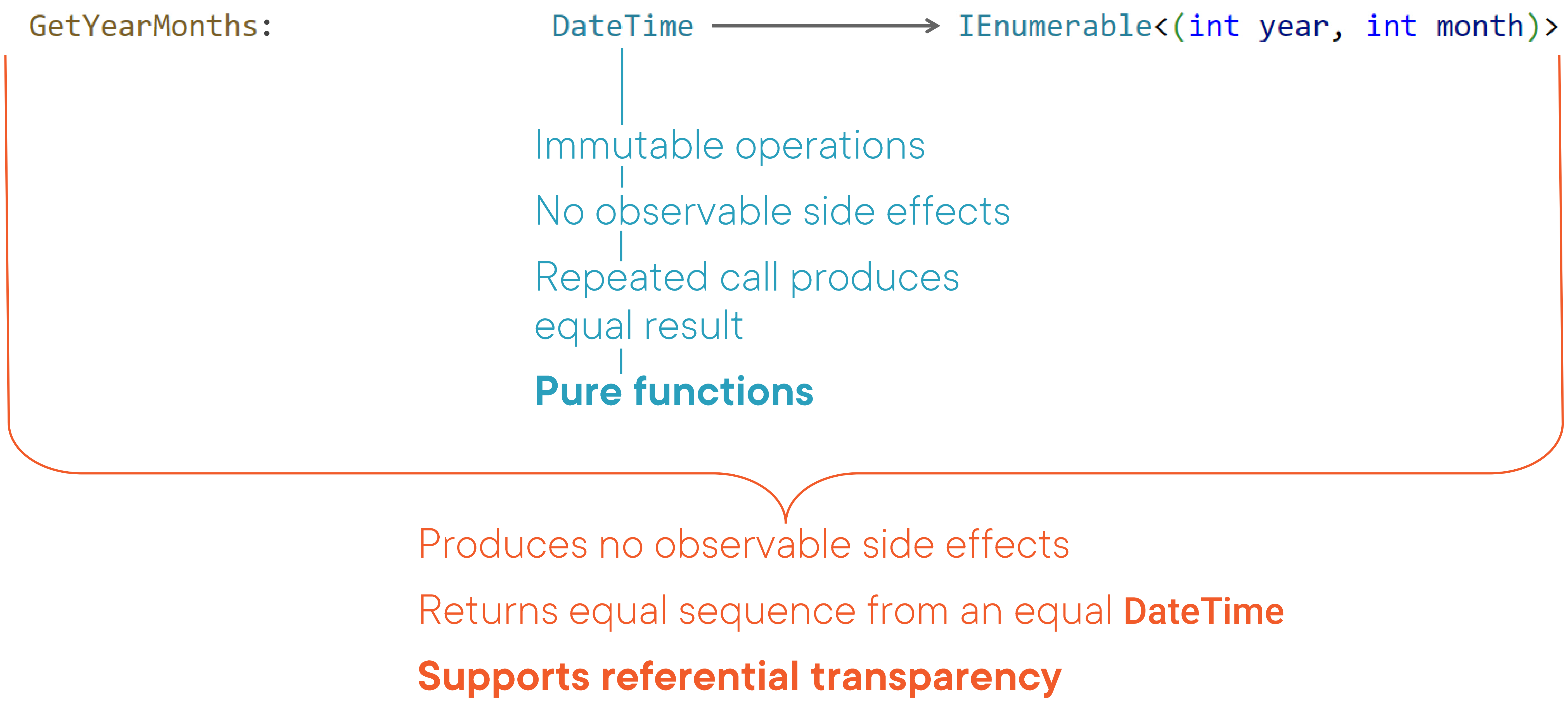
Immutable modeling
covered in depth



```
namespace Models;

public class DateTimeExtensions
{

}

```

DateTime

```
Date
Day
Hour
Millisecond
Minute
Month
...

Add()
AddDays()
AddHours()
AddMilliseconds()
AddMinutes()
...
```



`GetYearMonths:` `DateTime` → `IEnumerable<(int year, int month)>`

Immutable operations

No observable side effects

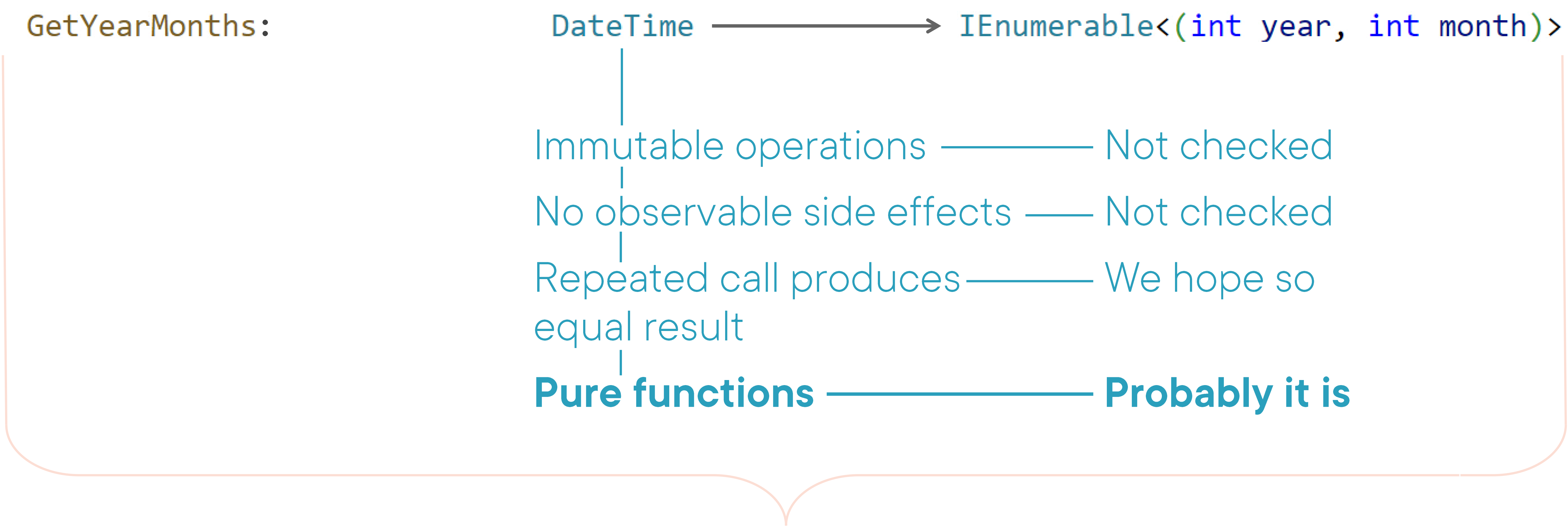
Repeated call produces equal result

Pure functions

Produces no observable side effects

Returns equal sequence from an equal **DateTime**

Supports referential transparency



```
GetYearMonths:          DateTime → IEnumerable<(int year, int month)>
```

Immutable operations — Not checked

No observable side effects — Not checked

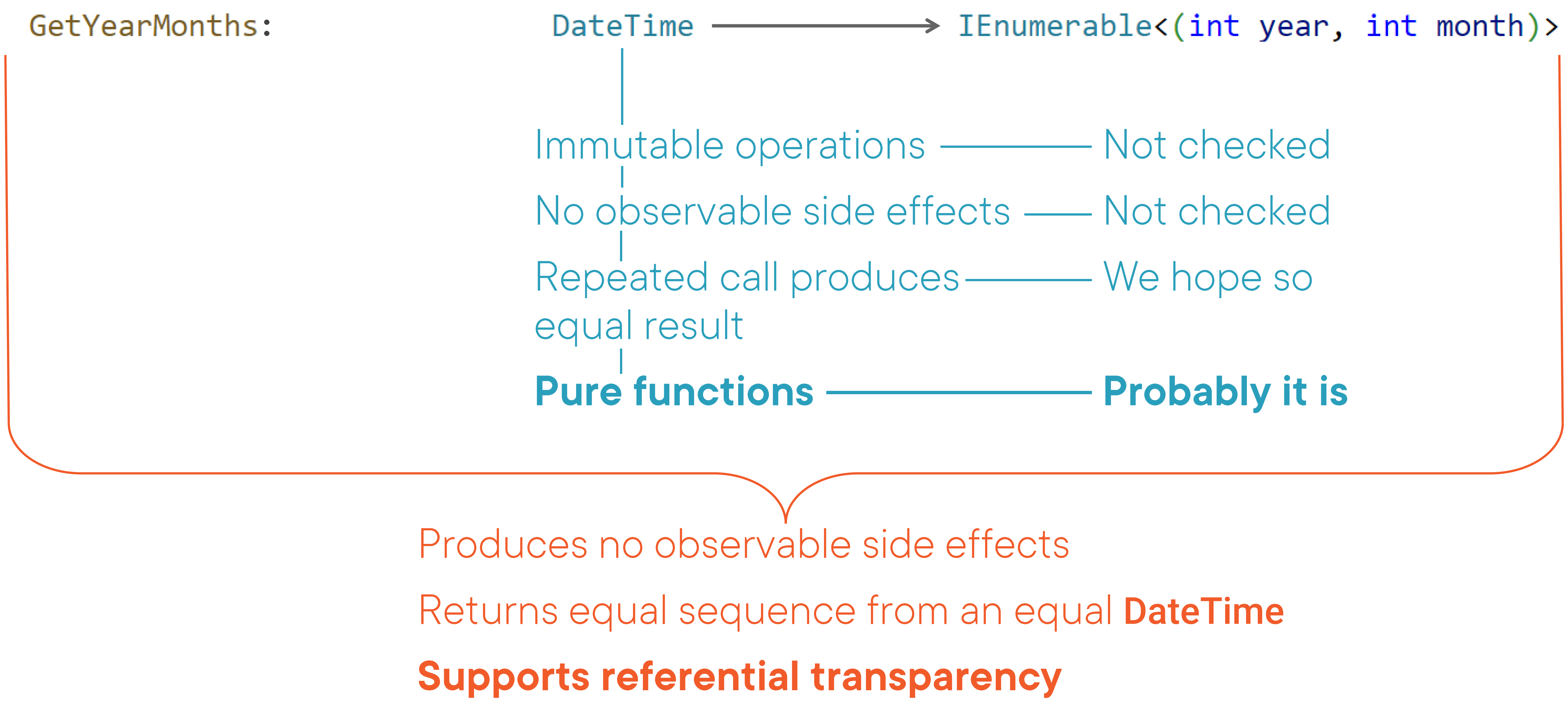
Repeated call produces equal result — We hope so

Pure functions — **Probably it is**

Produces no observable side effects

Returns equal sequence from an equal **DateTime**

Supports referential transparency



`GetYearMonths:` `DateTime` → `IEnumerable<(int year, int month)>`

Immutable operations	Not checked
No observable side effects	Not checked
Repeated call produces equal result	We hope so
Pure functions	**Probably it is**

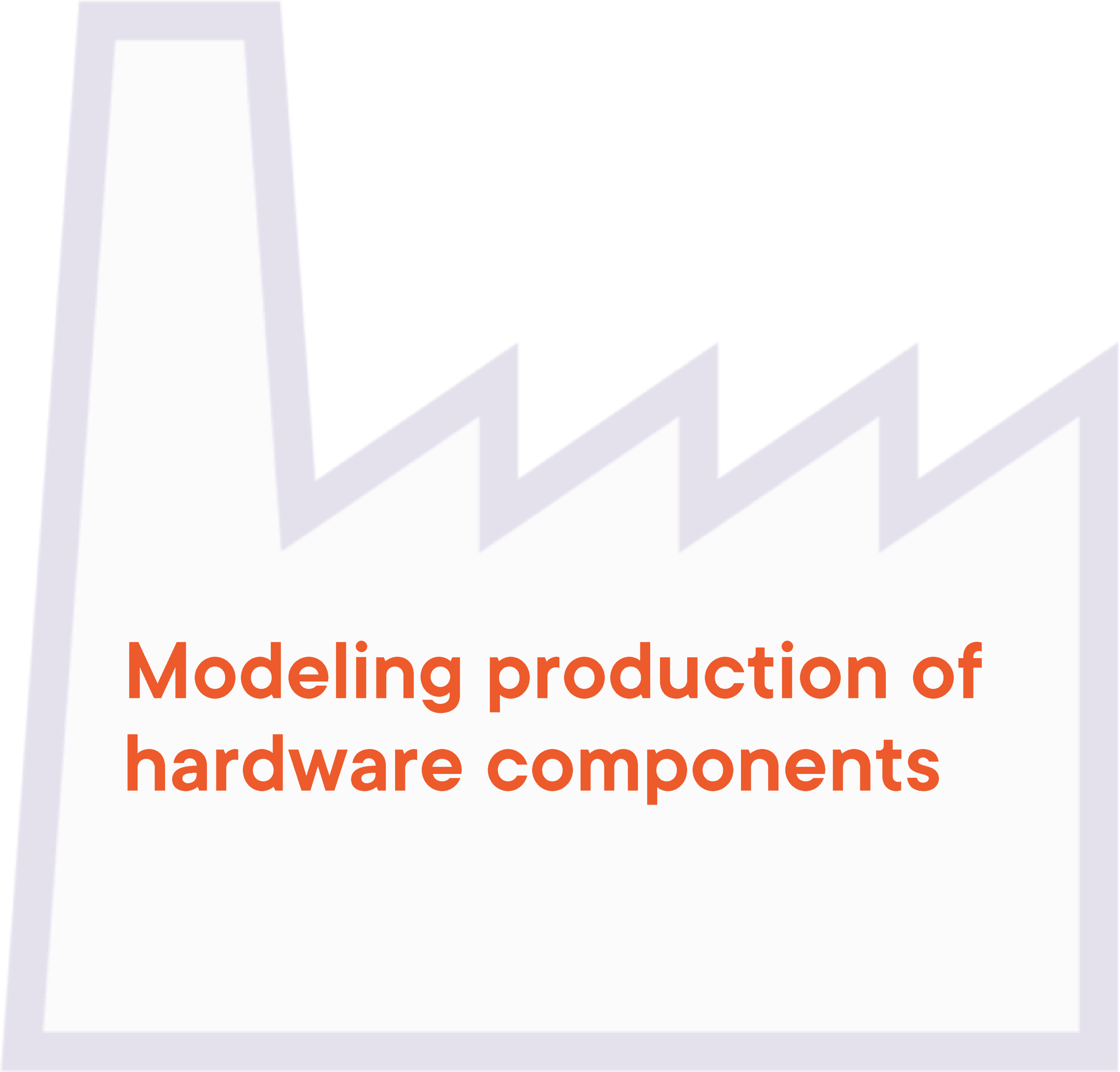
Produces no observable side effects

Returns equal sequence from an equal **DateTime**

Supports referential transparency

Mostly based on discipline

Demo Application



Modeling production of hardware components

Inventory of parts and materials

Record of suppliers

Specifications of products

Demo Application

Only use common C# syntax

Develop a functional domain model

Showcase modern C# functional capabilities

In the Following Modules…

Defining types and functions

Separation of types and functions

Modeling the domain with types

Defining functions on the type system

In the Following Modules…

Partial function application

Injecting parameters and dependencies

Discriminated unions

Increasing expressiveness of types

In the Following Modules…

Representation of missing objects

Wrapping it all together

Summary

Introduced basics of functional programming with C#

Applied modern C# syntax

- Tuples
- Extension methods
- LINQ

Designed a model without adding a single custom class

Functional syntax is native in C#

Up Next:

Introducing Functional Types and Functions