C# 11 and .NET 7

Christian Nagel

https://csharp.christiannagel.com

Christian Nagel

- Training
- Coaching
- Consulting
- Development

- Microsoft MVP
- www.cninnovation.com
- csharp.christiannagel.com
- @christiannagel

Professional
C# and .NET
2021 Edition

Christian Nagel

※BASTA!

Topics

- Focus on new C# features

- .NET updates related to C#

- Subset of all the new features released in November-2022

Small but still nice...

- Extended nameof scope

- Generic attributes

- Unsigned right shift >>>

```
[MyAttr(nameof(p1))]
public void Foo(int p1,
                [MyAttr(nameof(p1))] int p2)
{
}
```

Records with Classes and Structs

- C# 9: records with classes

- C# 10: records with structs

- Positional records (syntax with primary constructors)
- Nominal records

Do you prefer object initializers?

- **required** keyword
- SetsRequiredMembers attribute

Static virtual Members in Interfaces

Static Virtual Members in Interfaces

static abstract or static virtual members
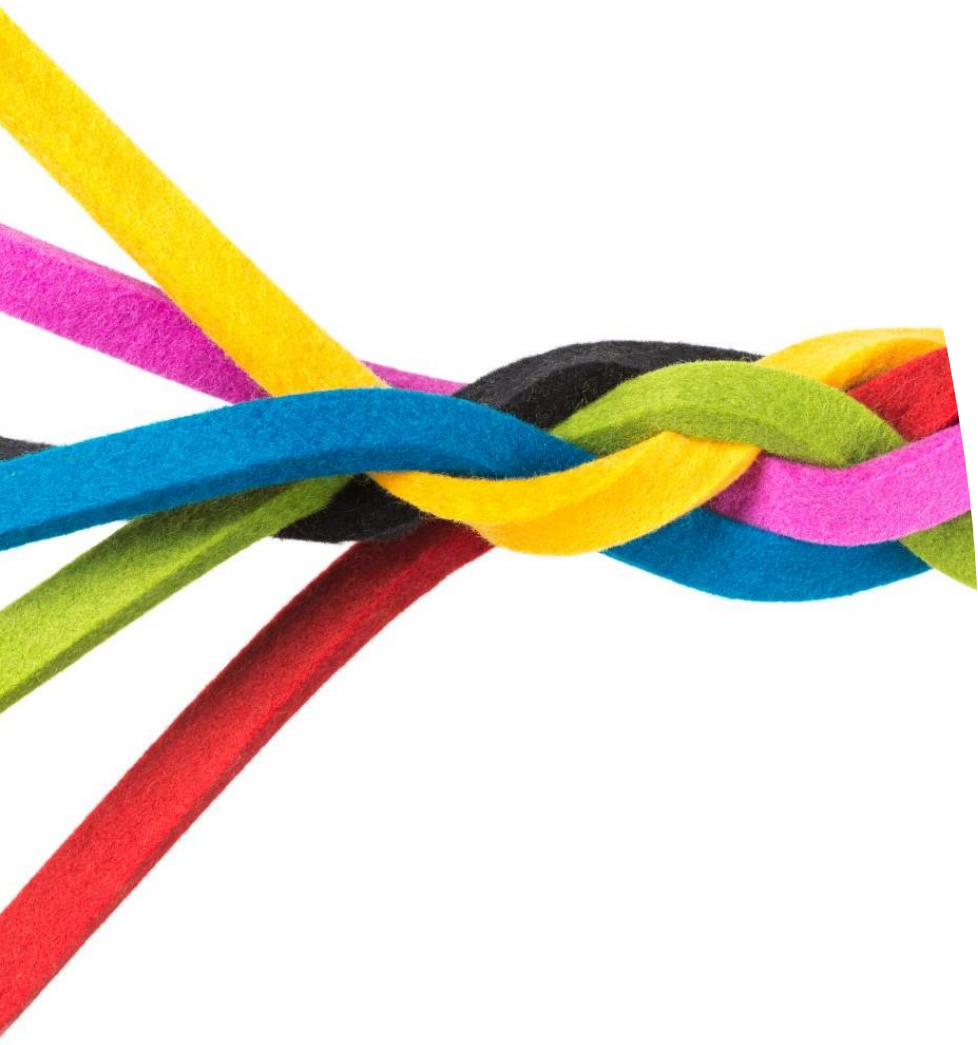
Use operators with generic types

Generic math: Number types implement INumber<T>

- IAdditionOperators, IDecrementOperators, IDivisionOperators…
- IAdditiveIdentity, IMultiplicativeIdentity, ISpanFormattable, ISpanParsable…

List Patterns

- Sequence of elements using [ ]
- Match elements with type, property, relational patterns
- _ discard: match a single element
- … range: match a sequence of zero or more

UTF-8 String Literals

- String format used on the Internet

- u8 suffix on strings

- Native type. ReadOnlySpan<byte>

- Convert to byte array with Span - ToArray

Raw String Literals

- Write code in a string?
- New format of strings literals
- Arbitrary text, whitespace, newlines, embedded quotes
- JSON, XML, C#...

Raw String Literal Syntax

- Three or more quotes
- Interpolation with $ or multiple $
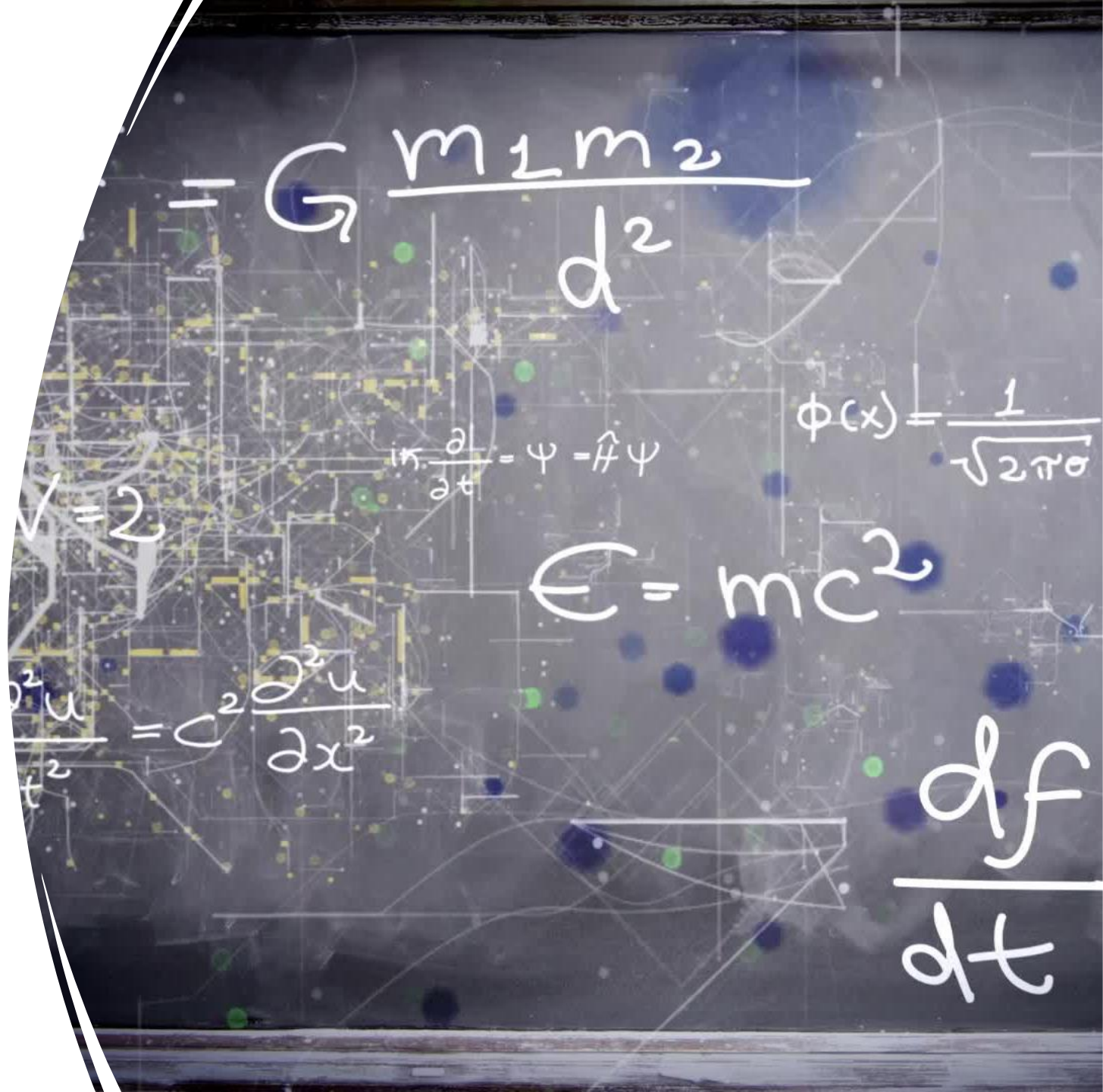
Access Modifier

- Access modifiers:
  - public
  - protected
  - private
  - internal
  - protected internal
  - private protected

- New access modifier: **file**
- Class only visible within the file, scope to the file

Source Generators

- LibraryImport instead of DllImport
  - Partial instead of extern
  - Generate code
- GeneratedRegex
  - Performance of RegexOptions.Compiled

Ahead of Time (AOT) Compilation

- Native AOT Deployment
- Start quickly, less memory consumption
- Restrictions
  - No runtime-code generation
  - No reflection
  - Limited number of libraries
- Self-contained apps
- Trimming

Summary – C# 11 Highlights

- Required modifier
- Raw string literals
- List Patterns
- Static virtual members in interfaces
- Kick-off AOT

Questions?

More Information

- https://github.com/cnilearn/bastamainz2022
- https://csharp.christiannagel.com
- https://learn.microsoft.com/dotnet/csharp

BASTA!

Thank you !