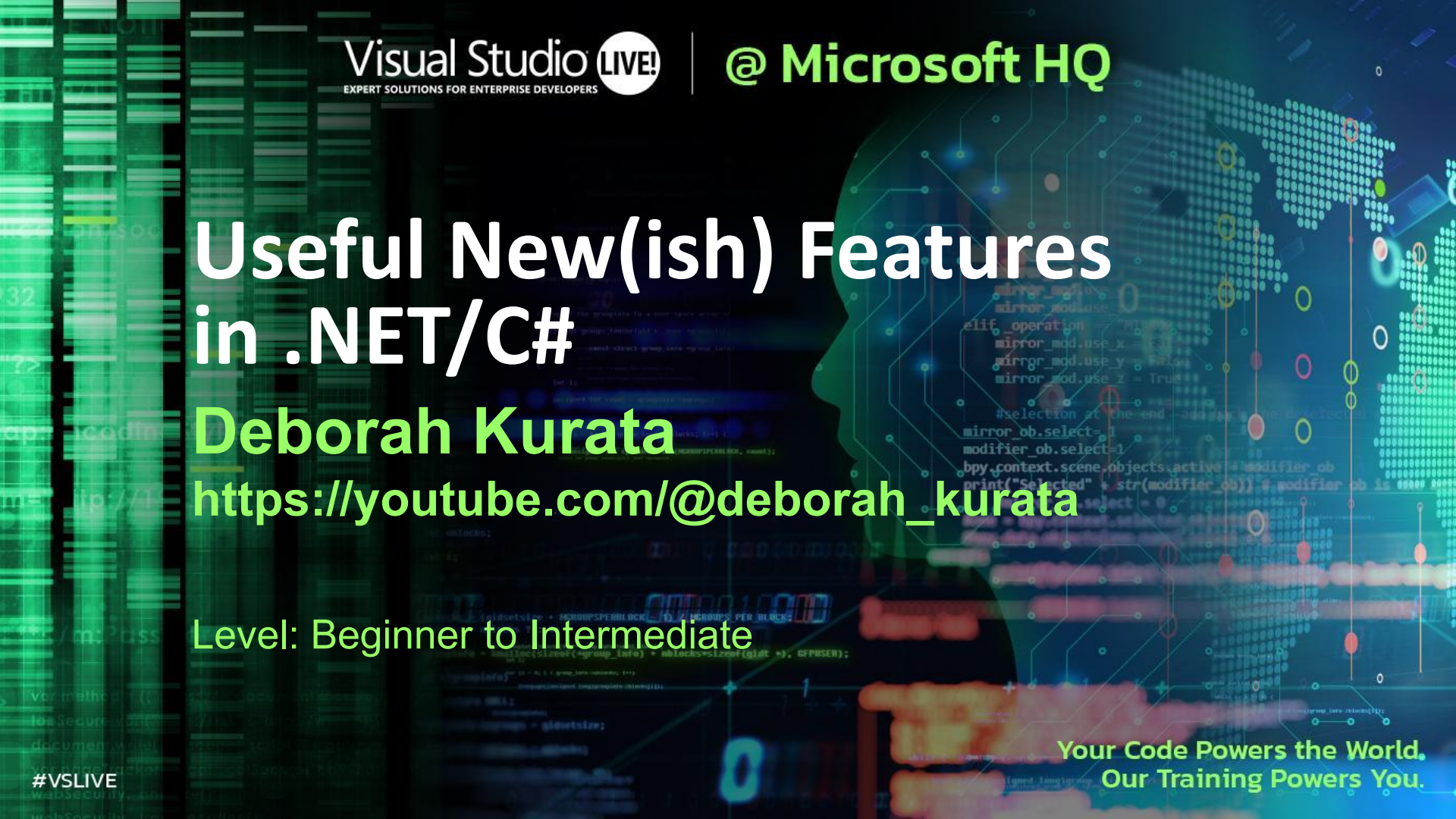

Visual Studio LIVE!
EXPERT SOLUTIONS FOR ENTERPRISE DEVELOPERS
@ Microsoft HQ
Useful New(ish) Features in .NET/C#
Deborah Kurata
https://youtube.com/@deborah_kurata
Level: Beginner to Intermediate
#VSLIVE
Your Code Powers the World.
Our Training Powers You.

Deborah Kurata

Developer

Pluralsight Author

YouTube content creator

youtube.com/@deborah_kurata

Microsoft Most Valuable Professional (MVP)

Google Developer Expert (GDE)

@deborahkurata

You?

- New to C#?
- Number of years using C#
  - 1-4 yr
  - 5-9 yrs
  - 10+ yrs

Session Survey

- Your feedback is very important to us
- Please take a moment to complete the session survey found in the mobile app
- Use the QR code or search for “Converge360 Events” in your app store
- Find this session on the Agenda tab
- Click “Session Evaluation”
- Thank you!

Links

@deborahkurata

https://bit.ly/deborahk-vslive-redmond-csharp-24

https://github.com/DeborahK/CSharp-Examples

https://www.youtube.com/@deborah_kurata

Things about strings

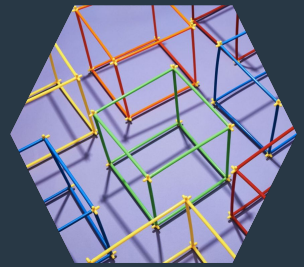
Switches are great

Patterns match state

Indices + ranges navigate

Dates commemorate

Records conceptualize



And code structure is realized

@deborahkurata

Most Recent C# Versions

- C# 7 - 2017 (.NET Framework 4.7)
- C# 8 - 2019 (.NET Core 3)
- C# 9 - 2020 (.NET 5)
- C# 10 - 2021 (.NET 6)
- C# 11 - 2022 (.NET 7)
- C# 12 - 2023 (.NET 8)

Strings

DEMO:
String Literals
String Interpolation

C# 11

Raw String Literals

- New format for string literals
- Allow for whitespace, new lines, embedded quotes, other special characters, etc
- Starts with at least three double-quote characters
- Ends with the same number of double-quotes

C# 11

Raw String Literals

```
string header = """
  <div class="card">
    <div class="card-header">
      Vehicle Detail
    </div>
  </div>
""";
```

Triple double quote

Matching ending

Aligns to the ending quotes

@deborahkurata

C# 11

Tips

- Use for multiple lines, quotes, or other characters requiring escape sequences
- Single line raw string literal requires the quotes on the same line
- Multiple line raw string literals require:
  - Opening quotes on a line above the raw string
  - Closing quotes on their own line below the raw string
  - Text must not be outdented from the closing quotes

C# 11

Verbatim vs Raw String Literals

Is there still a use case for a verbatim string literal?

```
string vehicleJSON = @"
  {
    ""id"": 1,
    ""name:"": ""AT-AT"",
    ""price"": 1999.99
  }";
```

```
string vehicleJSON = """
  {
    "id": 1,
    "name:": "AT-AT",
    "price": 1999.99
  }
  """;
```

@deborahkurata

C# 11 Verbatim vs Raw String Interpolation

Which would you prefer?

```
string vehicleJSON = $@"
  {{
    ""id"": 1,
    ""name:"": {vehicleName},
    ""price"": {price}
  }}";
```

```
string vehicleJSON = $$"""
{
  "id": 1,
  "name:": {{vehicleName}},
  "price": {{price}}
}
""";
```

Switch

C# 8

Switch Statement vs Expression

```
int quantity = 5;
string itemText = "";
switch (quantity)
{
  case 0:
   itemText = "No items";
   break;
  case 1:
   itemText = "One item";
   break;
  default:
   itemText = ${quantity} items";
   break;
 }
```

```
int quantity = 5;
string itemText = quantity switch
{
  0 => "No items",
  1 => "One item",
  _ => ${quantity} items"
};
```

pattern matching

Which would you prefer?

Pattern Matching

Complex if conditions

```
foreach (var vehicle in vehicles)
{
  if (goodCredit)
  {
    if (newVehicle)
    {
      message = """
        Congratulations on your new vehicle!
        We hope you enjoy driving it as much as we enjoyed building it.
        """;
    }
```

Business logic

Validation logic

Display logic

Do you have complex if logic in your projects?

@deborahkurata

C# 7

Pattern Matching

- A technique for checking an expression against a set of patterns
- Used in:
  - switch statement
  - switch expression (C# 8)
  - is expression

Pattern

```
if (quantity is null)
{
  return;
}
```

C# 8 Switch Expression Pattern Matching

```
bool isItemInStock = true;
string itemText = isItemInStock switch
{
    true => "Item is in stock",
    false => "Item is not in stock",
};
```

Assign to a variable

input value

patterns (conditions)

execution expression

Compares the input value to the patterns

Evaluated top to bottom
First match wins

DEMO:
Pattern Matching

C# 7+

Patterns

- Patterns provide concise and expressive code
- Replace complex if logic with patterns
- Combine patterns as needed
- Introduced in v7 and improved upon in later versions

C# 7+

Pattern Matching

```
Vehicle vehicle = new Vehicle { Id = 1,
    Name = "Model X", ZeroTo60 = 2.7M, Passengers = 6 };

decimal price = vehicle switch
{
  { ZeroTo60: < 3, Passengers: 7 } => 120_000,
  { ZeroTo60: < 3, Passengers: 6 } => 110_000,
  { ZeroTo60: < 4, Passengers: 7 } => 105_000,
  { ZeroTo60: < 4, Passengers: 6 } => 99_990,
  _ => 90_000
};
```

Do you have complex if's that could be patterns?

@deborahkurata

Indices and Ranges

DEMO:
Indices
Ranges

.NET 8

Index

```
string text = "This is some text!";
Index bang = ^1;
Console.WriteLine(text[bang]);          // Last character
Console.WriteLine(text[^text.Length]);  // First character
```

Works with any type that is "countable"

Would you use this?

@deborahkurata

.NET 8

Range

```
string text = "This is some text!";
Range some = 8..12;
Console.WriteLine(text[some]);    // Just some
Console.WriteLine(text[..4]);     // First word
```

Works with array and string types (copied not referenced)

Would you use this?

@deborahkurata

C# 12

Collection Expression

```
// List initialization
List<string> items = new() { "ring", "sword", "bow", "axe"};

// Or use the new C#12 Collection Expressions
List<string> items = ["ring", "sword", "bow", "axe"];
```

Dates

DEMO:
DateOnly
TimeOnly

.NET 6

DateOnly

```
DateOnly hireDate = new DateOnly(2024, 3, 14);
DateOnly current = DateOnly.FromDateTime(DateTime.Today);
```

DateOnly:
year, month, day

- Better type safety
- Better serialization
- Better match to database date type

.NET 6

TimeOnly

```
DateOnly hireDate = new DateOnly(2024, 3, 14);
DateOnly current = DateOnly.FromDateTime(DateTime.Today);
```

TimeOnly:
hour, minute, second

- TimeSpan was intended for **elapsed** time
- Better handles 24 hour clock
- Better serialization

Records

Records

Are you using records?

C# 9

Record Types

- Concise syntax for creating a reference type
- Primarily intended for immutable properties
- Provides:
  - Value equality (compares all property values)
  - Built-in formatting for display (ToString())
  - Built-in deconstruct

Immutable: Not changed after initialization

Instead, create a new one from a copy

Classes

```
public class Vehicle
{
   public required int Id { get; init;}
   public string? Name { get; set; }
   public decimal? Price { get; set; }
   public int Passengers { get; set; }
   public decimal ZeroTo60 { get; set; }

   public Vehicle() { }

   public decimal CalculateMilesPerCharge() { }
   public decimal CalculateNextPayment() { }
}
```

Properties

Methods

C# 9

Record Types

Positional parameters

```
public record Vehicle(int Id, string Name,
  decimal Price, int Passengers, decimal ZeroTo60);
```

Standard property syntax

```
public record Vehicle
{
  public required int Id { get; init;}
  public required string Name { get; init; }
  ...
}
```

C# 9

When Using Positional Syntax

```
public record Vehicle(int Id, string Name,
  decimal Price, int Passengers, decimal ZeroTo60);
```

- Creates public init-only auto-implemented properties
- Creates a primary parameterized constructor
- Creates `Equals()`, `ToString()`, and `Deconstruct()` methods

C# 9

Create Object vs Record

```
var vehicle = new Vehicle
{
   Id = 1,
   Name = "Model X Plaid",
   Price = 120_000M,
   Passengers = 6,
   ZeroTo60 = 2.7M
};
```

Positional syntax

```
var vehicle = new Vehicle(2, "Model Y", 99_990M, 5, 3.5M);
```

C# 9

Class Equality

```
var vehicle1 = new Vehicle
{
   Id = 1,
   Name = "Model X Plaid",
   Price = 120_000M,
   Passengers = 6,
   ZeroTo60 = 2.7M
};
```

```
var vehicle2 = new Vehicle
{
   Id = 1,
   Name = "Model X Plaid",
   Price = 120_000M,
   Passengers = 6,
   ZeroTo60 = 2.7M
};
```

FALSE!

```
Console.WriteLine($"Equal: {vehicle1 == vehicle2}");
```

Must manually override Equals()

Are they equal?

C# 9

Record Value Equality

```
var vehicle = new Vehicle(2, "Model Y", 99_990M, 5, 3.5M);
```

```
var vehicle2 = new Vehicle(2, "Model Y", 99_990M, 5, 3.5M);
```

TRUE!

```
Console.WriteLine($"Equal: {vehicle1 == vehicle2}");
```

Automatically overrides Equals()

Are they equal?

C# 9

Class Formatting for Display

```
var vehicle = new Vehicle
{
   Id = 1,
   Name = "Model X Plaid",
   Price = 120_000M,
   Passengers = 6,
   ZeroTo60 = 2.7M
};
```

```
Console.WriteLine(vehicle);
```

Displays the class name: Vehicle

Must manually override ToString()

What does this display?

C# 9

Record Formatting for Display

```
var vehicle = new Vehicle(2, "Model Y", 99_990M, 5, 3.5M);
```

```
Console.WriteLine(vehicle);
```

Vehicle { Id = 2, Name = Model Y, Price = 99990,
Passengers = 5, ZeroTo60 = 3.5 }

Automatically overrides ToString()

What does this display?

@deborahkurata

C# 9

Nondestructive Mutation

```
var vehicle = new Vehicle(2, "Model Y", 99_990M, 5, 3.5M);
```

with expression

```
var newVehicle = vehicle with { Passengers = 6 };
```

Support `with` expression to enable non-destructive mutation (changes)

Creating a new record from a copy and changing that copy

@deborahkurata

C# 9

Use Record Instead of Class:

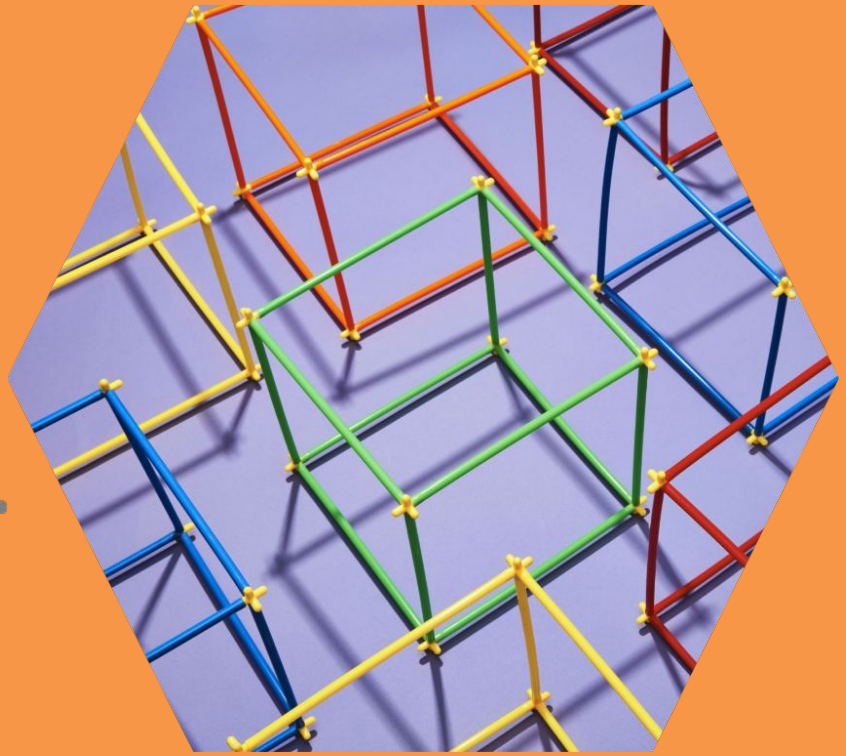
- When working with immutable (unmodified) data
- To reduce "boilerplate" (overrides)
- When code needs to compare value equality

C# 9

Use Class Instead of Record:

- When working with data that has behavior (methods)
- When working with mutable (modifiable) data

Code Structure



using Directive

```
using System;
using System.Text;
using System.Diagnostics.CodeAnalysis;
```

```
using System;

Console.WriteLine("Hello World!");
```

Use types defined in a namespace

Without specifying the fully qualified namespace

C# 10

Global Using Directives

```
global using System;
```

Effectively adds the using to every file in the project

Must be before any "normal" using

Consider adding to a separate GlobalUsings file

Visual Studio enables implicit usings

.NET 6

Implicit Usings (.csproj)

```
<Project Sdk="Microsoft.NET.Sdk">

  <PropertyGroup>
    <OutputType>Exe</OutputType>
    <TargetFramework>net7.0</TargetFramework>
    <ImplicitUsings>enable</ImplicitUsings>
    <Nullable>enable</Nullable>
  </PropertyGroup>

</Project>
```

ImplicitUsings

C# 10

Auto-generated Usings

```
// <auto-generated/>
global using global::System;
global using global::System.Collections.Generic;
global using global::System.IO;
global using global::System.Linq;
global using global::System.Net.Http;
global using global::System.Threading;
global using global::System.Threading.Tasks;
```

obj\debug\net7.0\VehicleSales.GlobalUsings.g.cs

C# 10

Auto-generated Usings

```
// <auto-generated/>
global using global::System;
global using global::System.Collections.Generic;
global using global::System.IO;
global using global::System.Linq;
global using global::System.Net.Http;
global using global::System.Threading;
global using global::System.Threading.Tasks;
```

Still need your own GlobalUsings for all else

Namespace

Namespace

```
namespace VehicleSales
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");
        }
    }
}
```

C# 10

File-scoped Namespace

File-scoped namespace

```
namespace VehicleSales;

class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Hello World!");
    }
}
```

Saves vertical and horizontal space

@deborahkurata

Useful New-ish Features in .NET/C#

Did you see something new or useful for your projects?

Links

@deborahkurata

https://bit.ly/deborahk-vslive-redmond-csharp-24

https://github.com/DeborahK/CSharp-Examples

https://www.youtube.com/@deborah_kurata

Session Survey

- Your feedback is very important to us
- Please take a moment to complete the session survey found in the mobile app
- Use the QR code or search for “Converge360 Events” in your app store
- Find this session on the Agenda tab
- Click “Session Evaluation”
- Thank you!