GOPAS
POCITACOVA.SKOLA.CZ

PLATINUM PARTNER
Microsoft DAQUAS

Visual Studio a C# vNext

Robert Haken
MVP, MCT, MCSD, MCSE
software architect, HAVIT, s.r.o.
haken@havit.cz,
@RobertHaken

G2B·TechEd

Visual Studio 2017 vNext - 15.6 Preview 3

C# 7.0

C# 7.1

C# 7.2

C# vNext - 7.3

C# vNext - 8.0

Visual Studio 2017 vNext – 15.6 (Preview 3)

Improved Solution Load Performance
Visual Studio Build Tools (incl. TypeScript + Node.js)
Snapshot Debugging from Debugging Target
VS Installer – Pause, Progress Details
Updated F# tooling
CPU Profiling - Async Call Stack Stitching (post-mortem only)
Navigate To Decompiled Sources
Test Explorer Hierarchy View (& Real-Time Tests Discovery)
GIT tooling - Tags

Visual Studio 2017 vNext

Side-by-side Installation

Feature Flags Extension

https://marketplace.visualstudio.com/items?itemName=PaulHarrington.FeatureFlagsExtension

DEMO

Visual Studio 2017 15.6 Preview 3, C# 7.1, 7.2, ...

C# vNext (7.3): Equality operators on tuples

```
// C# 7.3?
myTuple == (5, "test")
(count, label) == (5, "test")


// C# now
(myTuple.count == 5) && (myTuple.label == "test")
```

https://github.com/dotnet/csharplang/issues/190

C# vNext (7.3): Native-Sized Number Types

```
// now
IntPtr ptr;
UIntPtr uptr;
IntPtr.Size

// vNext
IntN i;
UIntN u;
FloatN f;
```

https://github.com/dotnet/corefxlab/blob/master/docs/specs/nativesized.md

C# vNext (7.3): Enum + Delegate constraints

```
public void DoSomething<T>(T data)
       where T: System.Delegate
```

```
public void  DoSomething<T>(T data)
       where T: System.Enum
```

https://github.com/dotnet/csharplang/issues/104
https://github.com/dotnet/csharplang/issues/103

C# vNext (8.0): Async streams and disposables

```
IAsyncEnumerable<Person> people = database.GetPeopleAsync();

foreach await (var p in people) { … }


using await (IAsyncDisposable resource = await store.GetRecordAsync(…)) { … }
```

https://github.com/dotnet/csharplang/blob/master/proposals/async-streams.md

C# vNext (8.0): Records

```
class Person(string First, string Last);

class Person : IEquatable<Person>
{
    public string First { get; }
    public string Last { get; }

    public Person(string First, string Last) => (this.First, this.Last) = (First, Last);

    public void Deconstruct(out string First, out string Last)
        => (First, Last) = (this.First, this.Last);

    public bool Equals(Person other)
        => other != null && First == other.First && Last == other.Last;

    public override bool Equals(object obj) => obj is Person other ? Equals(other) : false;
    public override int GetHashCode() => GreatHashFunction(First, Last);
    …
}
```

https://github.com/dotnet/csharplang/blob/master/proposals/records.md

C# vNext (8.0): Nullable reference types

```
string x = null; // compiler warning
Console.Write(x);
```

```
string? x = null; // compiler OK
Console.Write(x); // compiler warning
```

```
public void DoSomething(string! myNonNullableParameter)
// auto-generates a runtime null check
```

https://github.com/dotnet/csharplang/blob/master/proposals/nullable-reference-types.md

C# vNext (8.0): Default interface methods

```
interface IEnumerable<T>
{
  int Count()
  {
     int count = 0;
     foreach (var x in this)
        count++; return count;
  }
}

interface IList<T> : IEnumerable<T>
{
  int Count { get; }
  override int IEnumerable<T>.Count() => this.Count;
}
```

https://github.com/dotnet/csharplang/blob/master/proposals/default-interface-methods.md

C# vNext (??): Extension everything

```
extension Enrollee extends Person
{
    // static field
    static Dictionary<Person, Professor> enrollees = new Dictionary<Person, Professor>();

    // instance method
    public void Enroll(Professor supervisor) { enrollees[this] = supervisor; }

    // instance property
    public Professor Supervisor => enrollees.TryGetValue(this, out var supervisor) ? supervisor : null;

    // static property
    public static ICollection<Person> Students => enrollees.Keys;

    // instance constructor
    public Person(string name, Professor supervisor) : this(name) { this.Enroll(supervisor); }
}
```

Reference

Demo – C# 7.1+

https://github.com/hakenr/CSharp71Plus

Demo – C# 7.0

https://github.com/hakenr/CSharp7Demo

Blog – HAVIT Knowledge Base

http://knowledge-base.havit.cz/

Děkuji za pozornost.

www.G2BTeched.cz

www.TechEd.cz

www.gopas.cz