Immutability and equality in C#

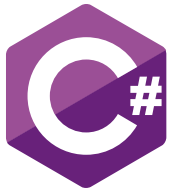
Agenda

Immutability in C#

Init properties

Equality in C#

Immutability



Immutable

An object that does not change after its creation

Immutability in C#

Immutability is new in C# 9 with records

`readonly` doesn't make a field of a class immutable (reference types)

Why ?

Mutable objects can cause some bugs (wrong data)

Immutable objects are thread-safe

Immutable objects can save memory

Immutability when used appropriately can improve the maintainability of your applications

When ?

When an objects should not change

Examples

Objects that are used to pass data like configuration data, library parameters and return objects, API call parameters and return objects, Data transfer objects (DTO)

C# immutable types

`string`

Primitive types (`int, double, decimal…`)

`DateTime`

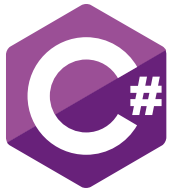
Enumerations

Immutable collections

Demo

Creating an immutable object without records

Init property accessor



Init only accessor

An init-only setter assigns a value to a property only during object construction

Init only accessor

Init properties accessible in object initializers

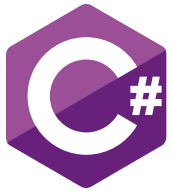
Init property accessor

```
public string Name { get; init; }
```

Demo

Creating an immutable thanks to init properties

Equality in C#



How to compare for equality

== operator

`Object.Equals`

`Object.ReferenceEquals`

Different implementations

Each type has its implementation of these methods

It can be overriden

GetHashCode

`Object.GetHashCode` method calculates a hash code for a quick equality comparison

The hash code is used to identify an object in a hash-based collection such as dictionaries

IEqualityComparer

Allows to implement a customized equality comparison for collections

A default implementation is provided by the `Default` property of the `EqualityComparer<T>` generic class

IEquatable

Allows to implement a type-specific method for equality comparison

When you have to override `Object.Equals()`

Allows to improve the comparison method performances

Demo

Compare instances of different types

Summary

Records allow to implement immutable objects

With an init-only setter you can assign a value to a property only during object construction

Each type has its own implementation of equality and comparison operators and methods

The GetHashCode method calculates a hash code for a quick equality comparison