Using Interfaces

Gill Cleeren
CTO Xpirit Belgium

@gillcleeren | www.xpirit.com/gill

Agenda

- Understanding interfaces
- Exploring and using built-in interfaces
- Using polymorphism with interfaces

Understanding Interfaces

Recap: The Different Custom Categories of Types

Enum

Struct

Class

Interface

Delegate

Understanding C# Interfaces

Define a contract that must be implemented by classes that use it

C# Interfaces

Uses interface keyword

Typically contain no implementation code, though possible since C# 8

Can’t be instantiated

Name typically starts with “I”

```
public interface IEmployee
{

    void PerformWork();

}
```

A Sample Interface

```
public void Manager: IEmployee
{

    public void PerformWork()
    {

        ...

    }

}
```

Implementing an Interface

Comparing Our Options

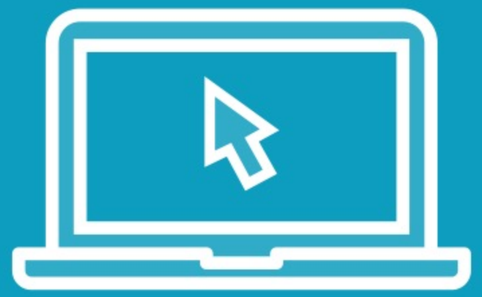
Abstract class

Only single inheritance is permitted

Interface

Multiple interfaces can be implemented

Demo



Creating a custom interface

Implementing an interface

Exploring and Using Built-in Interfaces

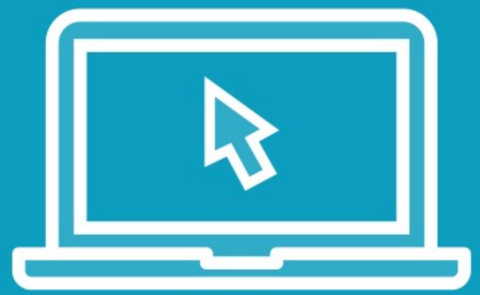
Commonly Used Interfaces in C#

IComparable	IEquatable	ICloneable
IEnumerable	IList	IDisposable

The ICloneable Interface

```
namespace System
{
    public interface ICloneable
    {
        object Clone();
    }
}
```

Demo



Implementing IComparable

Using the CompareTo() method

Using Polymorphism with Interfaces

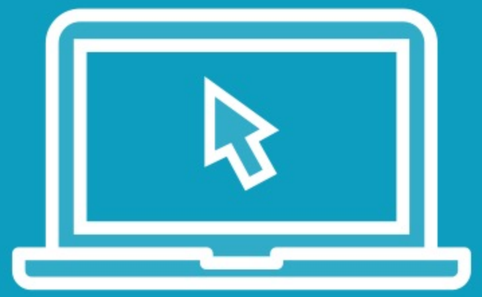
Using Polymorphism with Interfaces

```
IEmployee e1 = new Manager();

e1.PerformWork();
```

Using Polymorphism with Interfaces

Demo



Using polymorphism

Summary

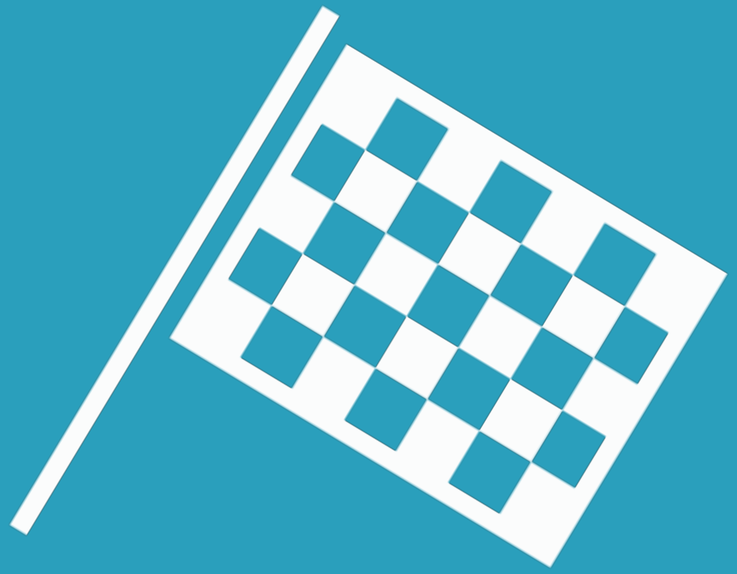
- Interfaces define a contract for classes
- All members must be implemented
- Similar to abstract classes
- Many built-in interfaces exist in .NET

Resources

Other relevant courses in the C# path:

- **C# Interfaces**
  - **Jeremy Clark**
- **IDisposable Best Practices for C# Developers**
  - **Elton Stoneman**


Path:
C# Development Fundamentals
www.pluralsight.com/paths/csharp



Congratulations on finishing this course!