Armen Melkumyan • 1st
Technical / Solutions Architect
1yr •

🔍💡 From .Net C# Interview Questions: Understanding Delegates

📌 Delegates in C#:

Delegates in C# 🚀 are types representing references to methods with specific parameter lists and return types.

They form the backbone of events 🎉 and callback methods 🔄 in C#.

Every delegate is derived from the System.Delegate class.

🔗 The Concept of Delegates:

Imagine a delegate as a pointer 🎯 or reference to a function.

When you create a delegate, you're specifying the method signature (return type and parameters) you intend to call.

This makes delegates incredibly versatile for dynamic method invocation.

💻 Practical Uses:

Delegates are crucial in scenarios requiring event handling and callbacks.

They allow methods to be passed as parameters, enhancing flexibility in code.

🧩 Why Delegates Matter:

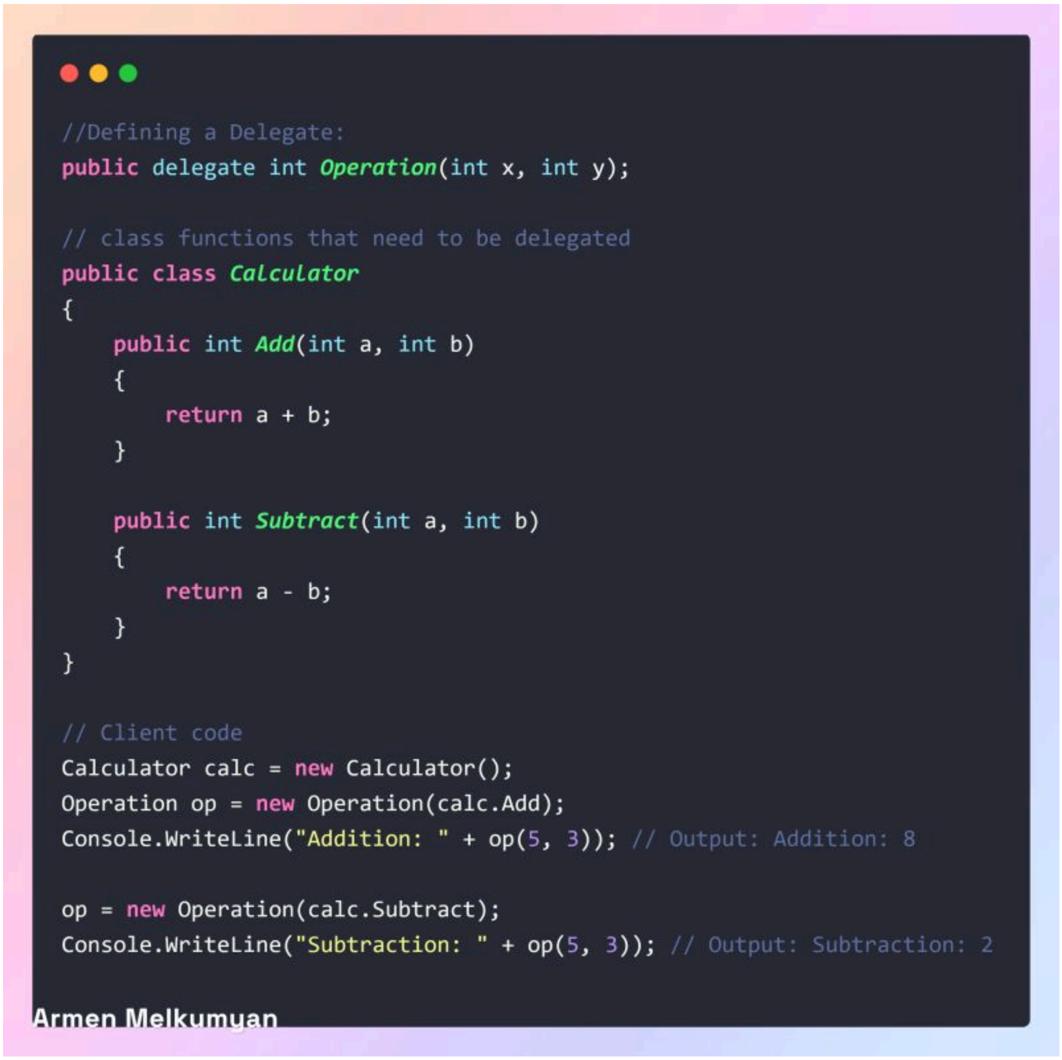
Essential for implementing event-driven programming.

Enable cleaner and more maintainable code by decoupling methods and their execution context.

#CSharpDelegates #DotNetProgramming #SoftwareEngineering

#TechInterviews #CSharpDevelopment #EventDrivenProgramming

#CleanCode #DeveloperCommunity



```
//Defining a Delegate:
public delegate int Operation(int x, int y);

// class functions that need to be delegated
public class Calculator
{
    public int Add(int a, int b)
    {
        return a + b;
    }

    public int Subtract(int a, int b)
    {
        return a - b;
    }
}

// Client code
Calculator calc = new Calculator();
Operation op = new Operation(calc.Add);
Console.WriteLine("Addition: " + op(5, 3)); // Output: Addition: 8

op = new Operation(calc.Subtract);
Console.WriteLine("Subtraction: " + op(5, 3)); // Output: Subtraction: 2
```

Armen Melkumyan

36 1 repost