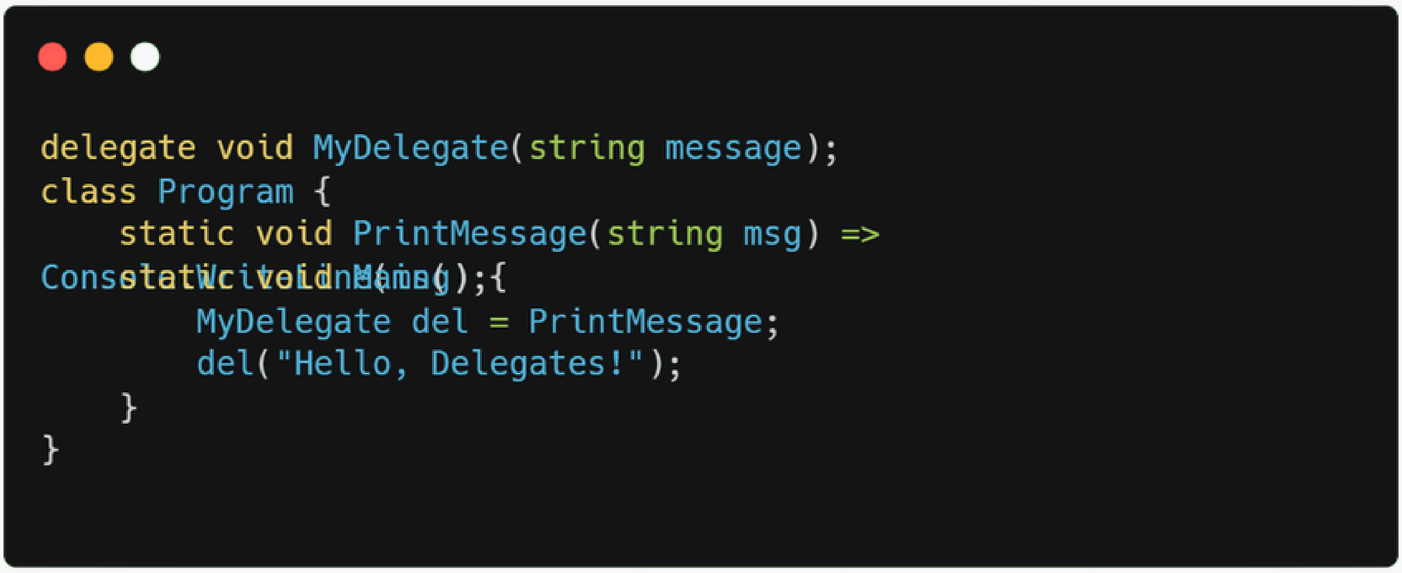
Delegates

- **What is it?**

  A delegate is a type that holds a reference to a method, allowing functions to be passed as parameters. It is similar to function pointers in C+

- **Use Cases:**
  - **Indirectly calling methods.**
  - **Implementing event-driven programming.**

- **Example:**



```
delegate void MyDelegate(string message);
class Program {
    static void PrintMessage(string msg) =>
Console.WriteLine(msg);
    static void Main() {
        MyDelegate del = PrintMessage;
        del("Hello, Delegates!");
    }
}
```

Events

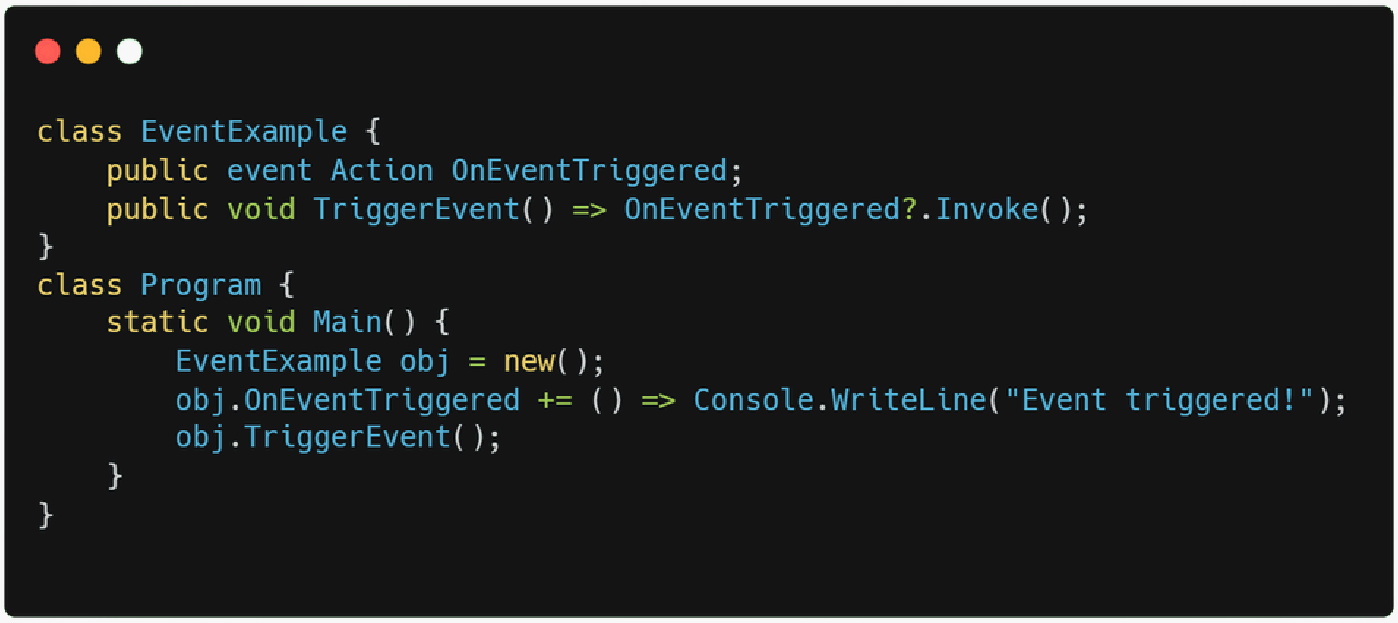
- What is it?

  An event is a mechanism that allows objects to notify subscribers when something happens.

- Use Cases:

  - GUI programming (button clicks, etc.).
  - Notification systems.

- Example:



```
class EventExample {
    public event Action OnEventTriggered;
    public void TriggerEvent() => OnEventTriggered?.Invoke();
}
class Program {
    static void Main() {
        EventExample obj = new();
        obj.OnEventTriggered += () => Console.WriteLine("Event triggered!");
        obj.TriggerEvent();
    }
}
```

Anonymous Methods

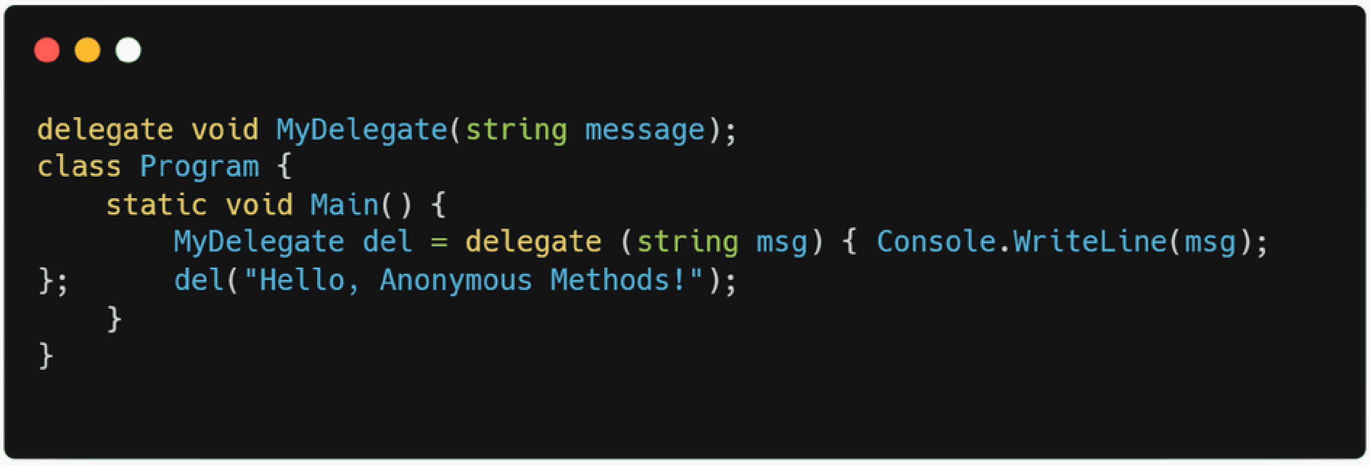
- What is it?

  Anonymous methods allow you to define inline methods without explicitly declaring them.

- Use Cases:

  - **Useful for short, one-time-use methods.**
  - **Used with delegates and events.**

- Example:



```
delegate void MyDelegate(string message);
class Program {
    static void Main() {
        MyDelegate del = delegate (string msg) { Console.WriteLine(msg);
};      del("Hello, Anonymous Methods!");
    }
}
```

Lambda Expressions

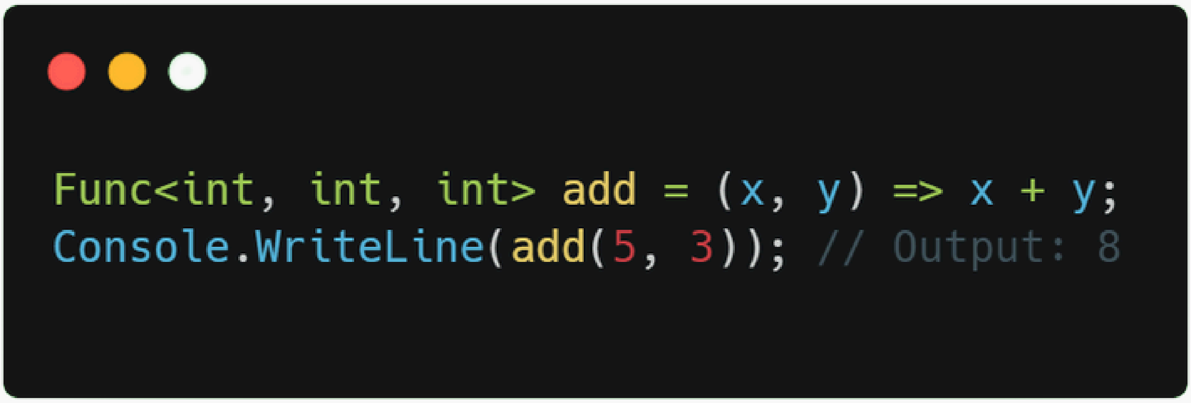
- What is it?

  A lambda expression is a concise way to write anonymous methods, commonly used with LINQ and delegates.

- Use Cases:

  - Simplifying code readability.
  - Used for filtering and data manipulation.

- Example:



```
Func<int, int, int> add = (x, y) => x + y;
Console.WriteLine(add(5, 3)); // Output: 8
```

Extension Methods

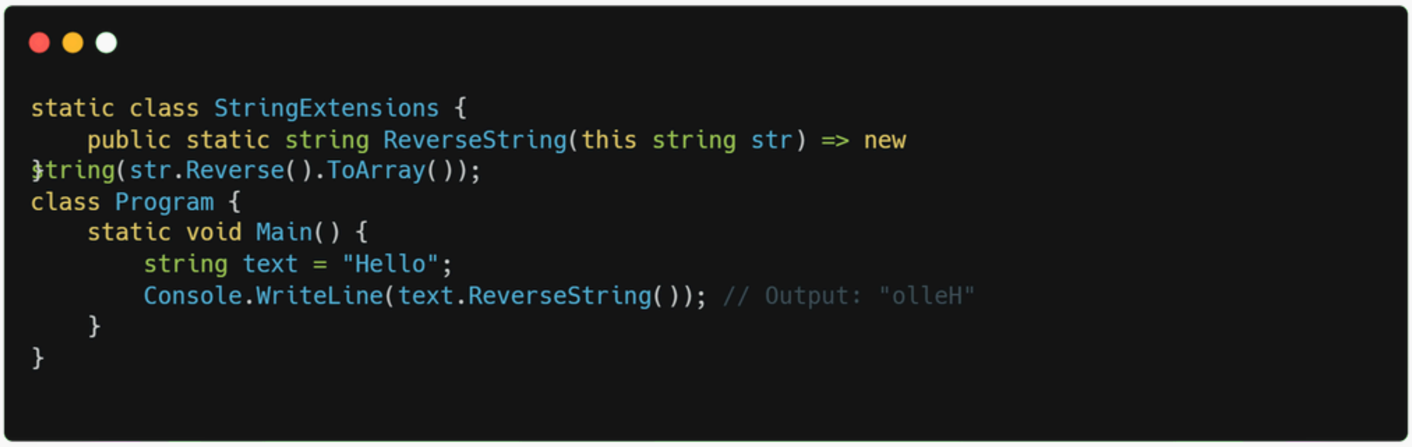
- What is it?

  Extension methods allow adding new functionalities to existing types without modifying their source code.

- Use Cases:

  - **Extending functionality of sealed or external classes.**
  - **Making utility functions more readable and reusable.**

- Example:



```csharp
static class StringExtensions {
    public static string ReverseString(this string str) => new
}string(str.Reverse().ToArray());
class Program {
    static void Main() {
        string text = "Hello";
        Console.WriteLine(text.ReverseString()); // Output: "olleH"
    }
}
```

by Ali Hatem
cis team