Delegates, Events, and Lambdas

Mirolim Majidov

1. Delegates and using of that

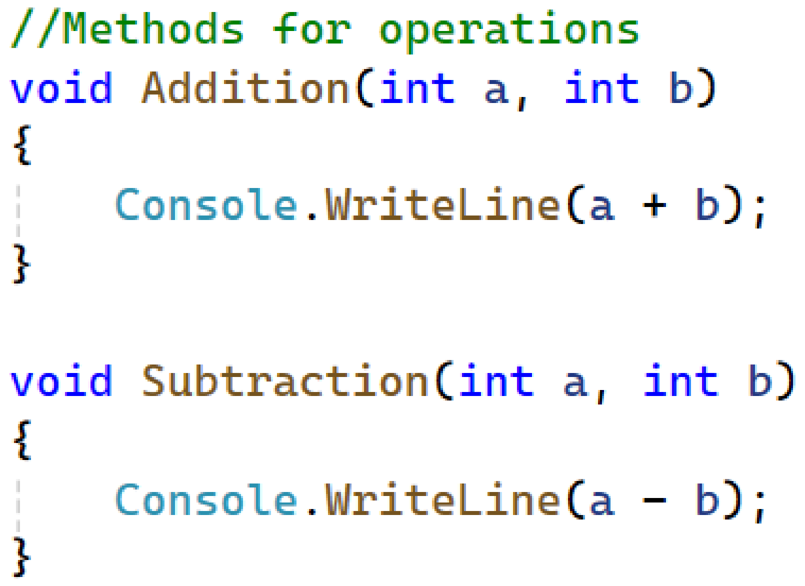
What is this?

Delegates in C# are objects that refer to methods. They are essentially type-safe function pointers.

How/where can I use this?

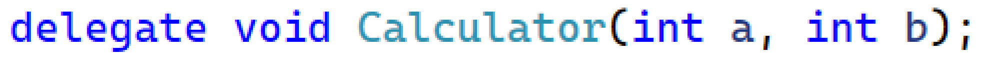
Delegates are often used in scenarios where you want to pass methods as parameters or create callback mechanisms.

Example:

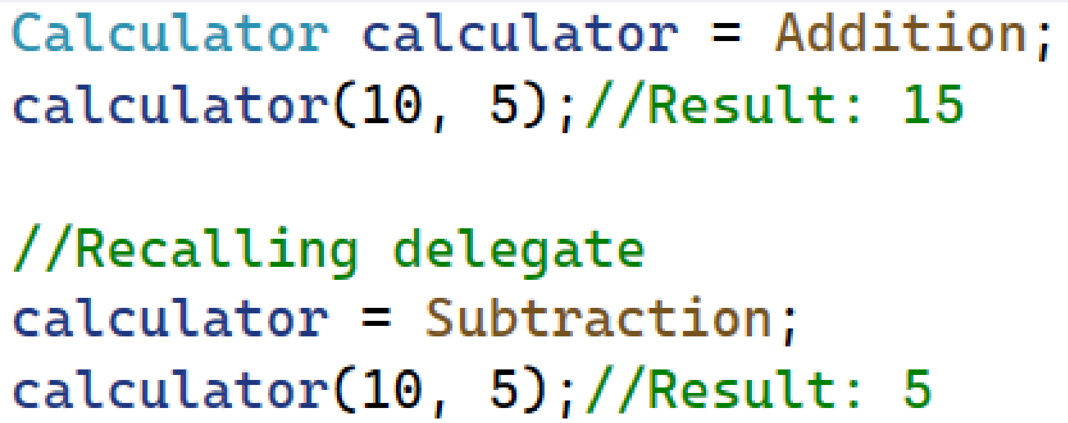


```
//Methods for operations
void Addition(int a, int b)
{
    Console.WriteLine(a + b);
}

void Subtraction(int a, int b)
{
    Console.WriteLine(a - b);
}
```



```
delegate void Calculator(int a, int b);
```



```
Calculator calculator = Addition;
calculator(10, 5);//Result: 15

//Recalling delegate
calculator = Subtraction;
calculator(10, 5);//Result: 5
```

2. Action and Func Delegates

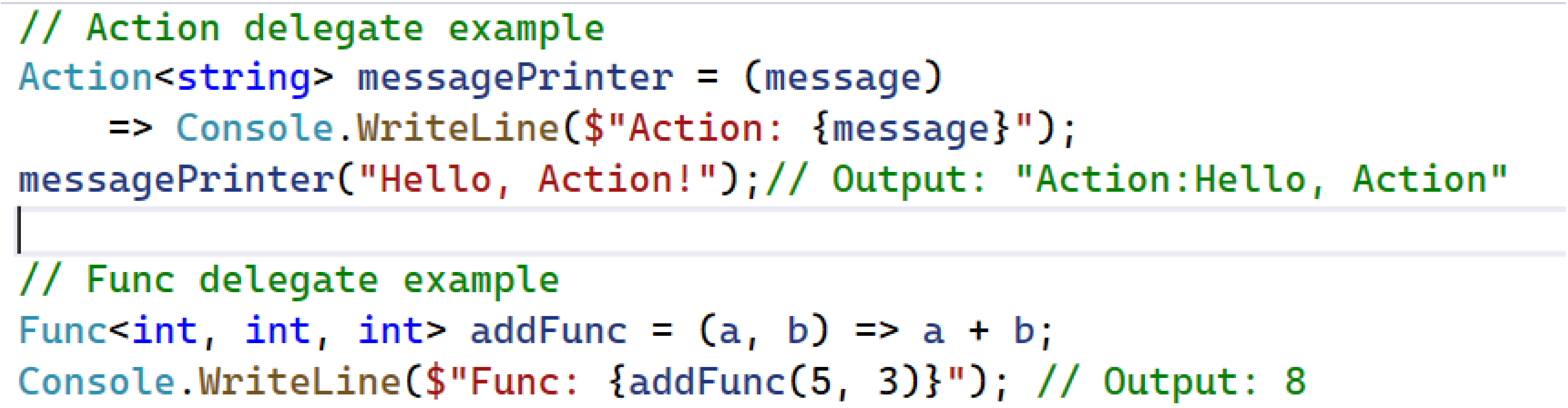
What is this?

Action and Func are generic delegate types provided by C# for handling methods with and without return values, respectively.

How/where can I use this?

Action is useful when you want to define a method that performs an action but does not return a value. Func is handy when you want to define a method that returns a value.

Example:



```
// Action delegate example
Action<string> messagePrinter = (message)
    => Console.WriteLine($"Action: {message}");
messagePrinter("Hello, Action!");// Output: "Action:Hello, Action"

// Func delegate example
Func<int, int, int> addFunc = (a, b) => a + b;
Console.WriteLine($"Func: {addFunc(5, 3)}"); // Output: 8
```

3. Anonymous Methods

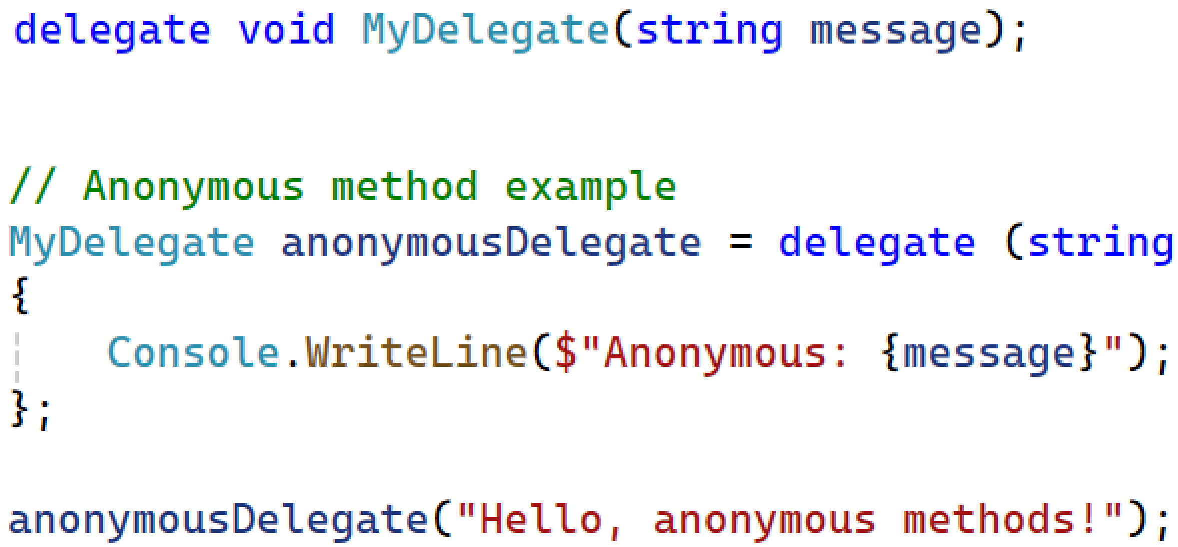
What is this?

Anonymous methods are methods without a name that can be defined inline using the delegate keyword.

How/where can I use this?

They are useful when a small, one-time-use method is needed, often in event handling and asynchronous programming.

Example:



```
delegate void MyDelegate(string message);


// Anonymous method example
MyDelegate anonymousDelegate = delegate (string message)
{
    Console.WriteLine($"Anonymous: {message}");
};

anonymousDelegate("Hello, anonymous methods!");
```

4. Lambdas

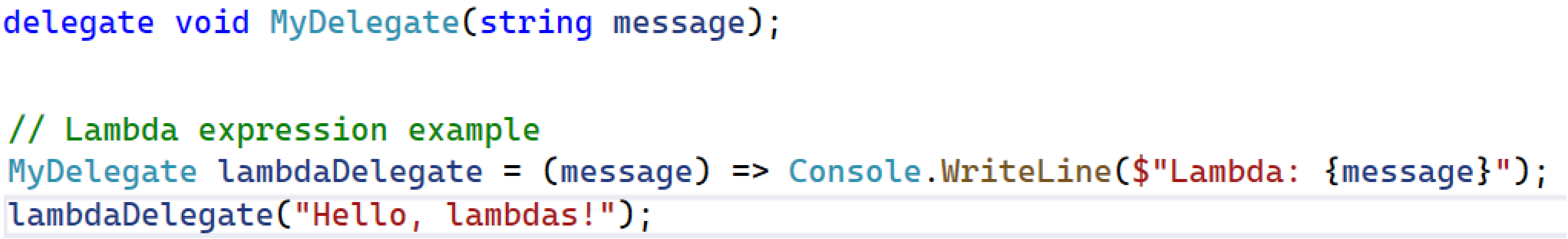
What is this?

Lambdas are concise, anonymous methods that can be used to create delegates or expression tree types..

How/where can I use this?

Lambdas are commonly used for short, simple operations. They are extensively used in LINQ queries and functional programming.

Example:



```
delegate void MyDelegate(string message);

// Lambda expression example
MyDelegate lambdaDelegate = (message) => Console.WriteLine($"Lambda: {message}");
lambdaDelegate("Hello, lambdas!");
```

5. Events

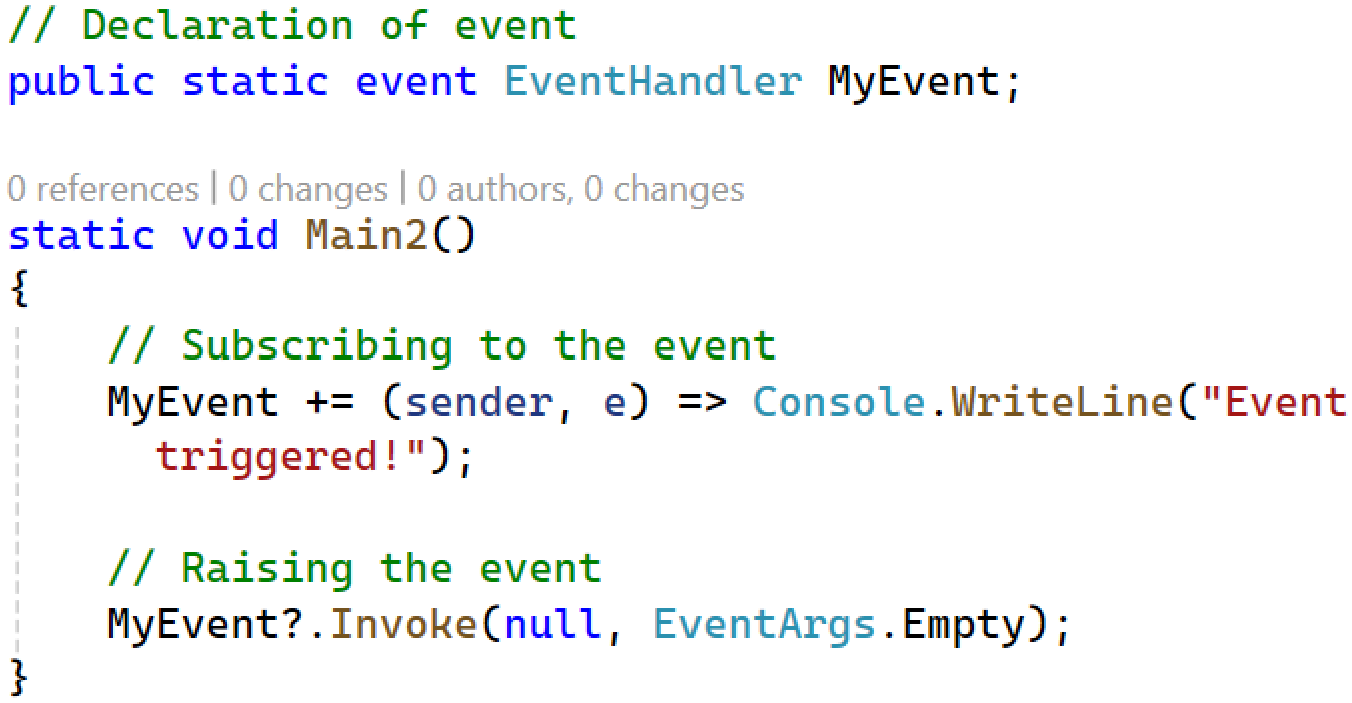
What is this?

Events provide a way for classes to communicate with each other. They are a type of delegate that allows a class to notify other classes when something happens.

How/where can I use this?

Events are commonly used in GUI programming, decoupling components in a system, and implementing the observer pattern.

Example:



```
// Declaration of event
public static event EventHandler MyEvent;

0 references | 0 changes | 0 authors, 0 changes
static void Main2()
{
    // Subscribing to the event
    MyEvent += (sender, e) => Console.WriteLine("Event
      triggered!");

    // Raising the event
    MyEvent?.Invoke(null, EventArgs.Empty);
}
```