.NET EVENTS

Introduction

- All events in the .NET Framework class library are based on the EventHandler delegate, which is defined as follows:

  public delegate void EventHandler(object sender, EventArgs e);

- To define an event, you use the event keyword in the signature of your event class, and specify the type of delegate for the event.
- Typically, to raise an event, you add a method that is marked as protected and virtual. Name this method OnEventName; for example, OnDataReceived. The method should take one parameter that specifies an event data
- **Example:**

  The following example shows how to declare an event named ThresholdReached. The event is associated with the EventHandler delegate and raised in a method named OnThresholdReached.

Event Declaration

```
class Counter
{    public event EventHandler ThresholdReached;//ThresholdReached EventName
    protected virtual void OnThresholdReached(EventArgs e)
    {       EventHandler handler = ThresholdReached;
        if (handler != null)
        {
            handler(this, e);
        }
    }
    // provide remaining implementation for the class
}
```

Delegates

- The .NET Framework provides the EventHandler and EventHandler<TEventArgs> delegates to support most event scenarios.
- Use the EventHandler delegate for all events that do not include event data.
- Use the EventHandler<TEventArgs> delegate for events that include data about the event.
- These delegates have no return type value and take two parameters (an object for the source of the event and an object for event data).

Event Data

- Data that is associated with an event can be provided through an event data class
- The EventArgs class is the base type for all event data classes
- . EventArgs is also the class you use when an event does not have any data associated with it.
- You can pass the EventArgs.Empty value when no data is provided.
- The EventHandler delegate includes the EventArgs class as a parameter.
- The following example shows an event data class named ThresholdReachedEventArgs. It contains properties that are specific to the event being raised.

```
public class ThresholdReachedEventArgs : EventArgs
{
    public int Threshold { get; set; }
    public DateTime TimeReached { get; set; }
}
```

Event Data Class

```
public class ThresholdReachedEventArgs : EventArgs
{
    public int Threshold { get; set; }
    public DateTime TimeReached { get; set; }
}
```

Reading and Demos

- Run the following applications:
  - The first example shows how to raise and consume an event that doesn't have data.
    Down load the netevent1 (zip)
  - The second example shows how to raise and consume an event that provides data.
    Down load the netevent2 (zip)
  - The third example shows how to declare a delegate for an event.
    Down load the netevent3 (zip)