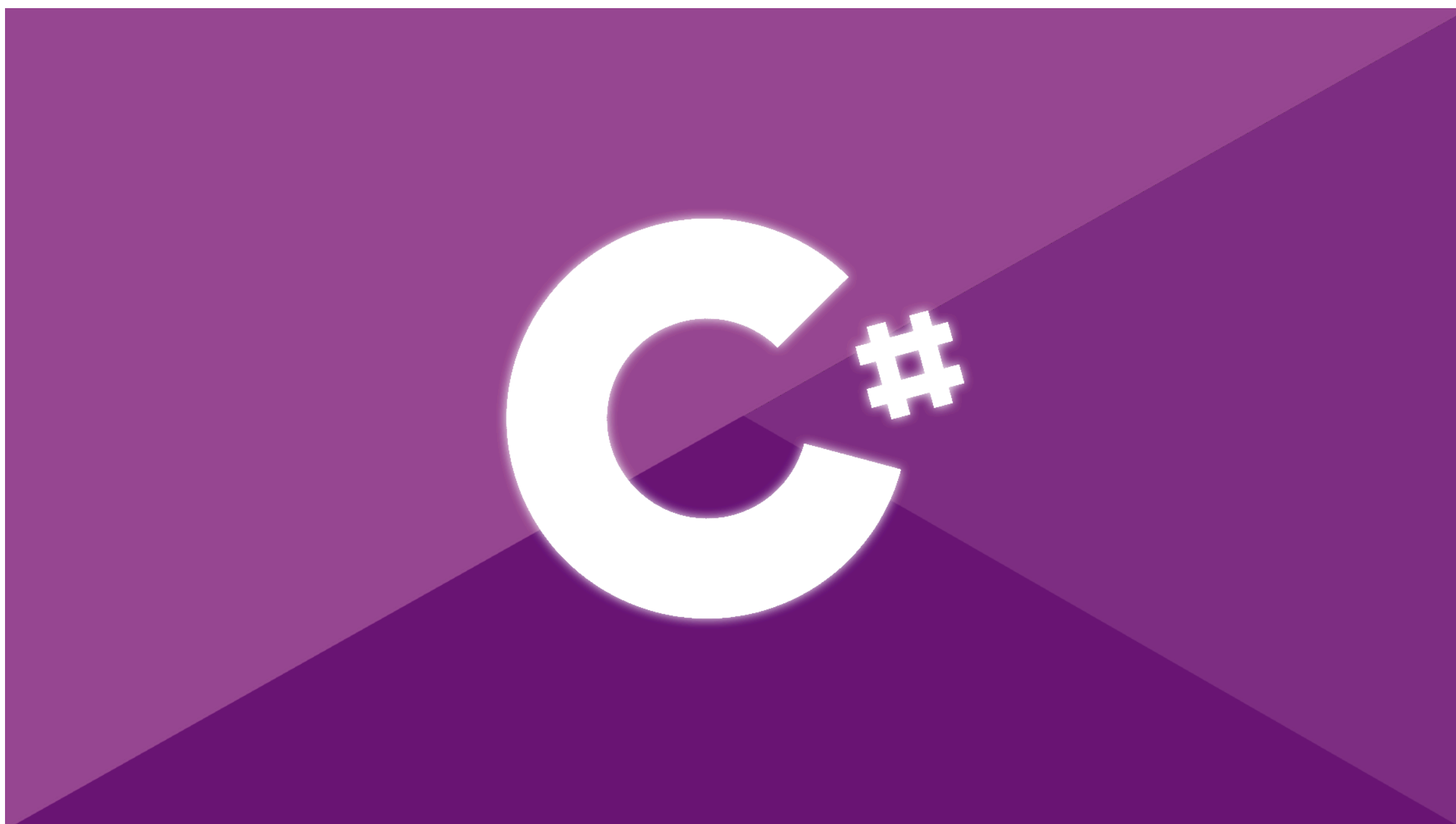


## 学习如何在C#中轻松实现串口数据接收：清晰步骤与实例代码

作者：微信公众号：【架构师老卢】

11-30 7:51

132



**概述：**以上C#示例演示了如何使用SerialPort类实现串口数据接收。通过设置串口属性、定义数据接收事件处理程序，你可以轻松地打开串口、监听数据，并在事件处理程序中对接收到的数据进行处理。这提供了一个基本框架，可根据实际需求进行定制。

在C#中实现串口数据接收通常需要使用System.IO.Ports命名空间提供的SerialPort类。以下是一个简单的例子，演示了如何在C#中接收串口数据。

首先，确保你的项目引用了System.IO.Ports命名空间。你可以在代码中添加如下的using语句：

```
1 using System;
2 using System.IO.Ports;
```

然后，创建一个SerialPort对象，并设置必要的属性，如端口号、波特率等。在这个例子中，我们使用COM1端口和波特率为9600。你需要根据实际情况修改这些参数。

```
1 SerialPort serialPort = new SerialPort();
2 serialPort.PortName = "COM1"; // 设置端口号
3 serialPort.BaudRate = 9600; // 设置波特率
```

接下来，设置数据接收的事件处理程序。你可以使用DataReceived事件来处理接收到的数据。在事件处理程序中，你可以读取接收到的数据并进行处理。

```
1 serialPort.DataReceived += new SerialDataReceivedEventHandler(DataReceivedHandler);
2
3 // 数据接收事件处理程序
4 private static void DataReceivedHandler(object sender, SerialDataReceivedEventArgs e)
5 {
6     SerialPort sp = (SerialPort)sender;
7     string data = sp.ReadExisting(); // 读取接收到的数据
8     Console.WriteLine("Received data: " + data);
9     // 在这里进行对接收到的数据的处理
10 }
```

最后，打开串口并开始接收数据。

```
1 serialPort.Open(); // 打开串口
2
3 // 接收数据
4 Console.WriteLine("Press any key to stop receiving data...");
5 Console.ReadKey();
6
7 serialPort.Close(); // 关闭串口
```

以下是完整的例子：

```
1 using System;
2 using System.IO.Ports;
3
4 class Program
5 {
6     static void Main()
7     {
8         SerialPort serialPort = new SerialPort();
9         serialPort.PortName = "COM1"; // 设置端口号
10        serialPort.BaudRate = 9600; // 设置波特率
11
12        serialPort.DataReceived += new SerialDataReceivedEventHandler(DataReceivedHandler);
13
14        try
15        {
16            serialPort.Open(); // 打开串口
17
18            Console.WriteLine("Press any key to stop receiving data...");
19            Console.ReadKey();
20        }
21        catch (Exception ex)
22        {
23            Console.WriteLine("Error: " + ex.Message);
24        }
25        finally
26        {
27            serialPort.Close(); // 关闭串口
28        }
29    }
30
31    // 数据接收事件处理程序
32    private static void DataReceivedHandler(object sender, SerialDataReceivedEventArgs e)
33    {
34        SerialPort sp = (SerialPort)sender;
35        string data = sp.ReadExisting(); // 读取接收到的数据
36        Console.WriteLine("Received data: " + data);
37        // 在这里进行对接收到的数据的处理
38    }
39 }
```

请根据实际需求修改端口号、波特率以及数据处理部分的代码。这个例子只是一个基本的框架，具体的实现可能需要根据你的应用场景进行调整。

