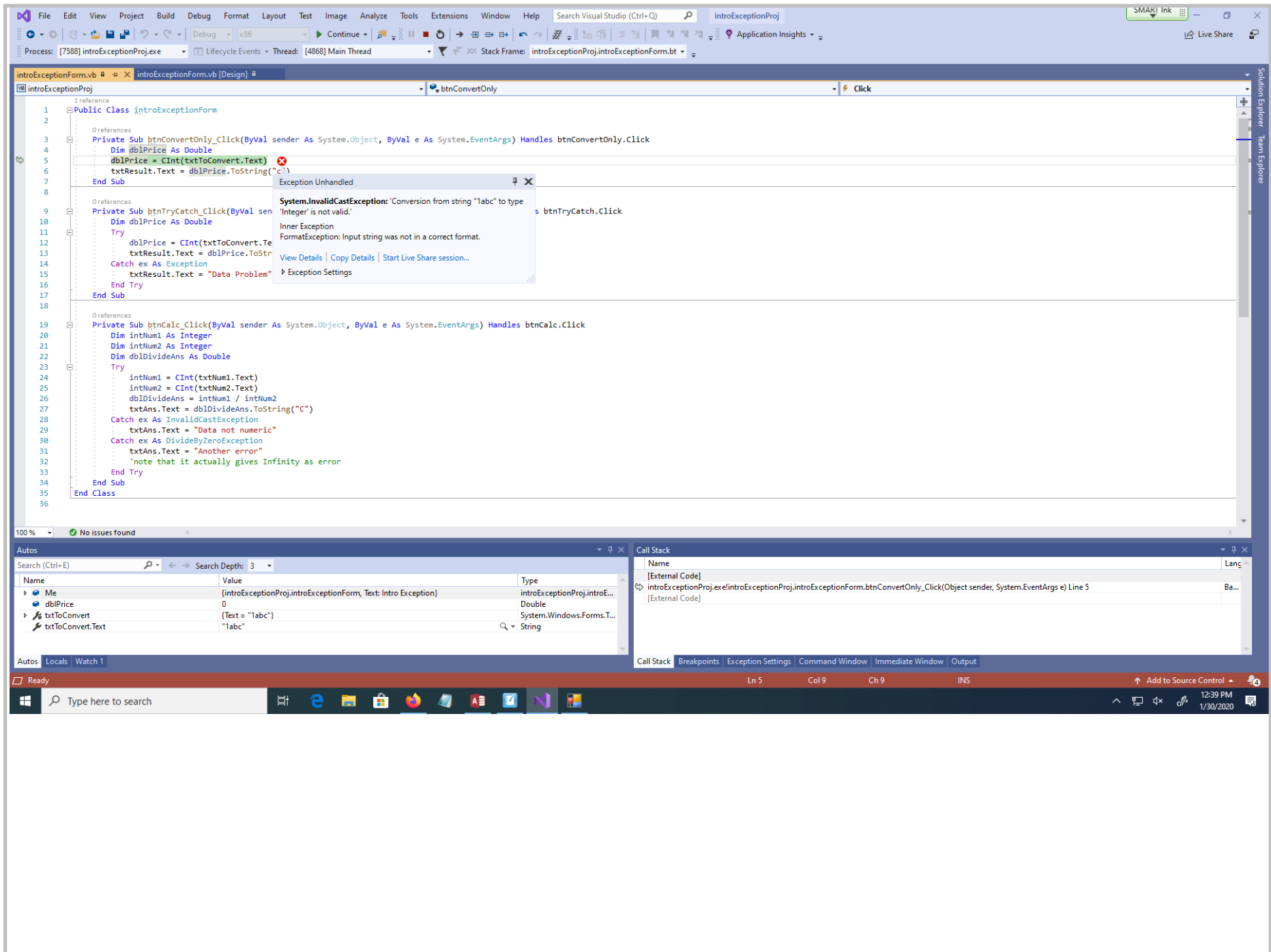


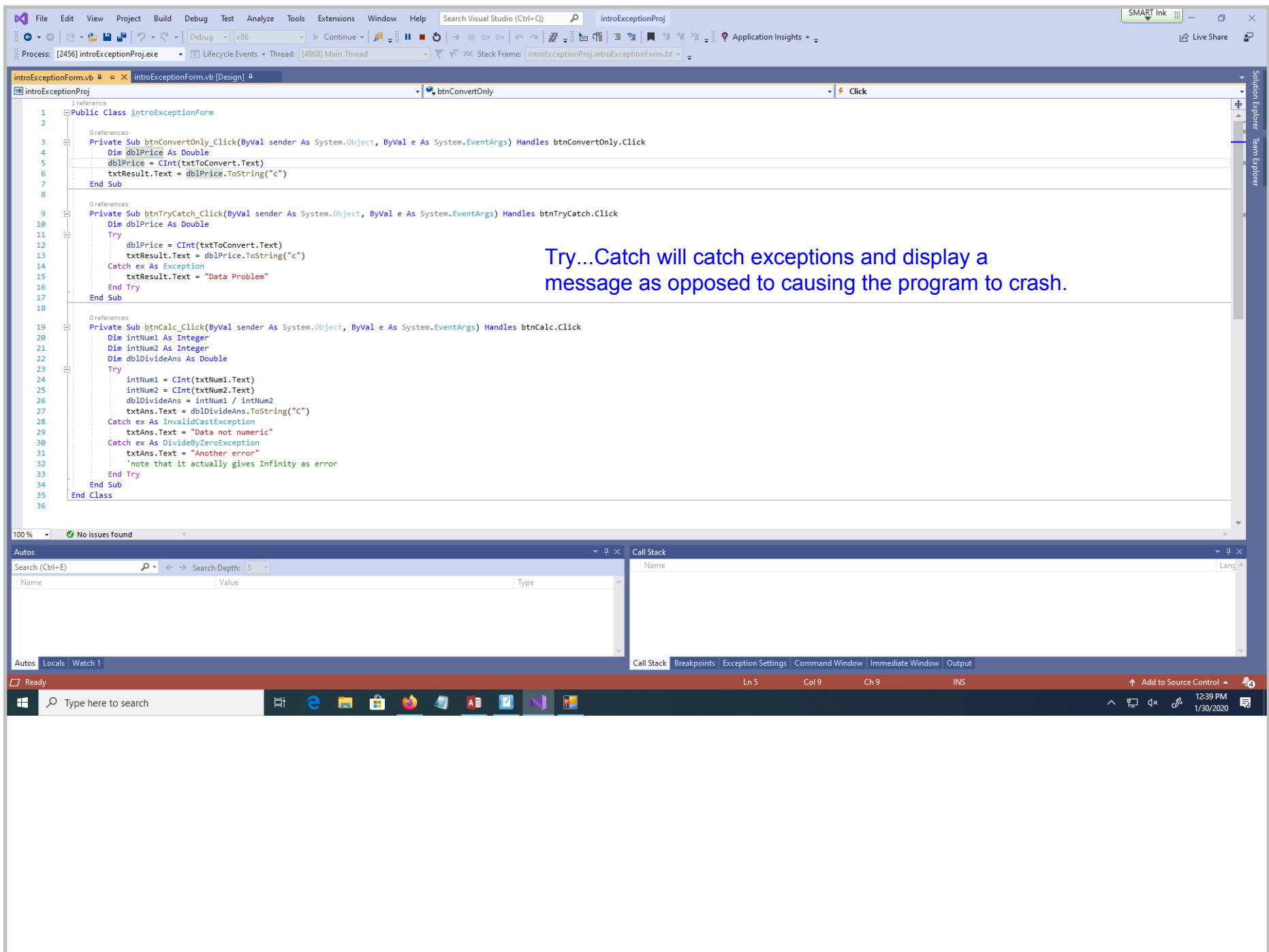
The screenshot displays the Visual Studio IDE with a C# code file open. The code defines a class `IntroExceptionForm` with three event handlers: `btnConvertOnly_Click`, `btnTryCatch_Click`, and `btnCalc_Click`. The `btnTryCatch_Click` method uses a `try-catch` block to handle a `FormatException` and sets the text of `txtResult` to "Data Problem". The `btnCalc_Click` method uses `try-catch` blocks to handle `InvalidCastException` (setting text to "Data not numeric") and `DivideByZeroException` (setting text to "Another error").

Two blue annotations are present in the code editor: "Non numerics will cause a crash." near line 13 and "Chap #3 dealing with exceptions." near line 29.

A floating window titled "Intro Exce" is overlaid on the code. It contains a "To Convert:" field with the value "12.78", a "Result:" field with the value "\$13.00", and two buttons: "Convert Only" and "Convert Try...Catch". Below this, there are "Numbers:" fields and a "Calculate" button, and an "Answer:" field.

The bottom of the IDE shows the "Autos" window with a search bar and a table with columns "Name", "Value", and "Type". The "Call Stack" window is also visible. The Windows taskbar at the bottom shows the system tray with the time "12:38 PM" and date "1/30/2020".





The image shows a Visual Studio IDE window titled 'introExceptionProj'. The main editor displays the code for 'introExceptionForm.vb'. The code includes three event handlers: 'btnConvertOnly_Click', 'btnTryCatch_Click', and 'btnCalc_Click'. The 'btnTryCatch_Click' method uses a try-catch block to handle a 'Data Problem' exception. The 'btnCalc_Click' method uses try-catch blocks to handle 'InvalidCastException', 'DivideByZeroException', and a custom error message for infinity. A floating dialog box titled 'Intro Exce' is overlaid on the code, showing input fields for 'To Convert' (1abc), 'Result' (Data Problem), 'Numbers' (12 and 0), and 'Answer' (=). The dialog has buttons for 'Convert Only', 'Convert Try...Catch', and 'Calculate'. Two dashed pink arrows point from the 'Data Problem' string in the code to the 'Result' field in the dialog, and from the 'note that it actually gives Infinity as error' comment to the 'Answer' field. The bottom of the screen shows the Windows taskbar with the time 12:42 PM on 1/30/2020.

```
1 Public Class introExceptionForm
2
3     Private Sub btnConvertOnly_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnConvertOnly.Click
4         Dim dblPrice As Double
5         dblPrice = CInt(txtToConvert.Text)
6         txtResult.Text = dblPrice.ToString("c")
7     End Sub
8
9     Private Sub btnTryCatch_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnTryCatch.Click
10        Dim dblPrice As Double
11        Try
12            dblPrice = CInt(txtToConvert.Text)
13            txtResult.Text = dblPrice.ToString("c")
14        Catch ex As Exception
15            txtResult.Text = "Data Problem"
16        End Try
17    End Sub
18
19    Private Sub btnCalc_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnCalc.Click
20        Dim intNum1 As Integer
21        Dim intNum2 As Integer
22        Dim dblDivideAns As Double
23        Try
24            intNum1 = CInt(txtNum1.Text)
25            intNum2 = CInt(txtNum2.Text)
26            dblDivideAns = intNum1 / intNum2
27            txtAns.Text = dblDivideAns.ToString("C")
28        Catch ex As InvalidCastException
29            txtAns.Text = "Data not numeric"
30        Catch ex As DivideByZeroException
31            txtAns.Text = "Another error"
32            'note that it actually gives Infinity as error
33        End Try
34    End Sub
35 End Class
36
```

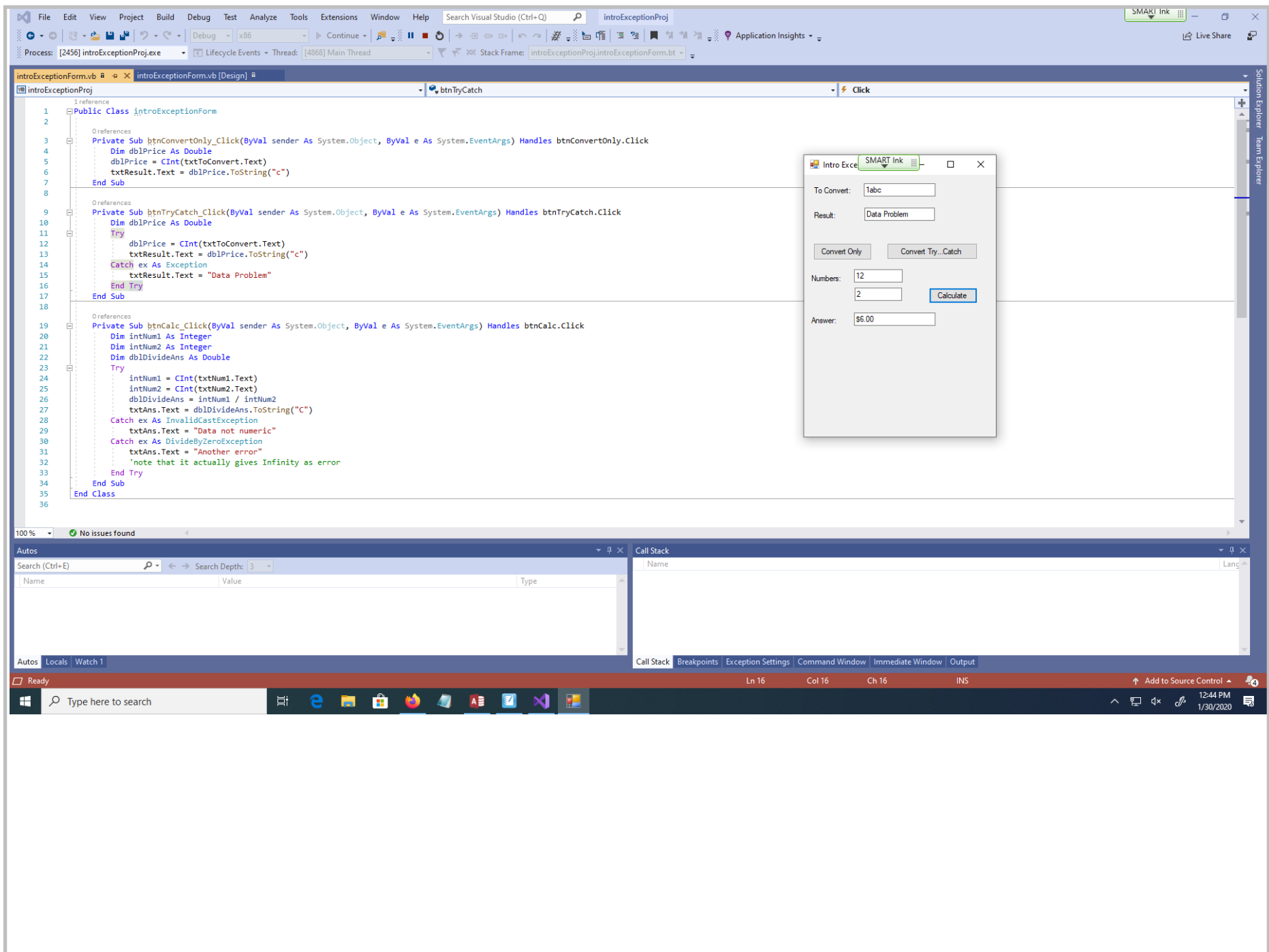
Visual Studio interface showing a VB.NET code file named `introExceptionForm.vb` in Design mode. The code defines a `Public Class introExceptionForm` with three event handlers: `btnConvertOnly_Click`, `btnTryCatch_Click`, and `btnCalc_Click`. The `btnTryCatch_Click` method uses a `Try` block to convert a string to a double, and catches an `Exception` to display "Data Problem". The `btnCalc_Click` method uses a `Try` block to perform integer division, catching `InvalidCastException` (displaying "Data not numeric") and `DivideByZeroException` (displaying "Another error").

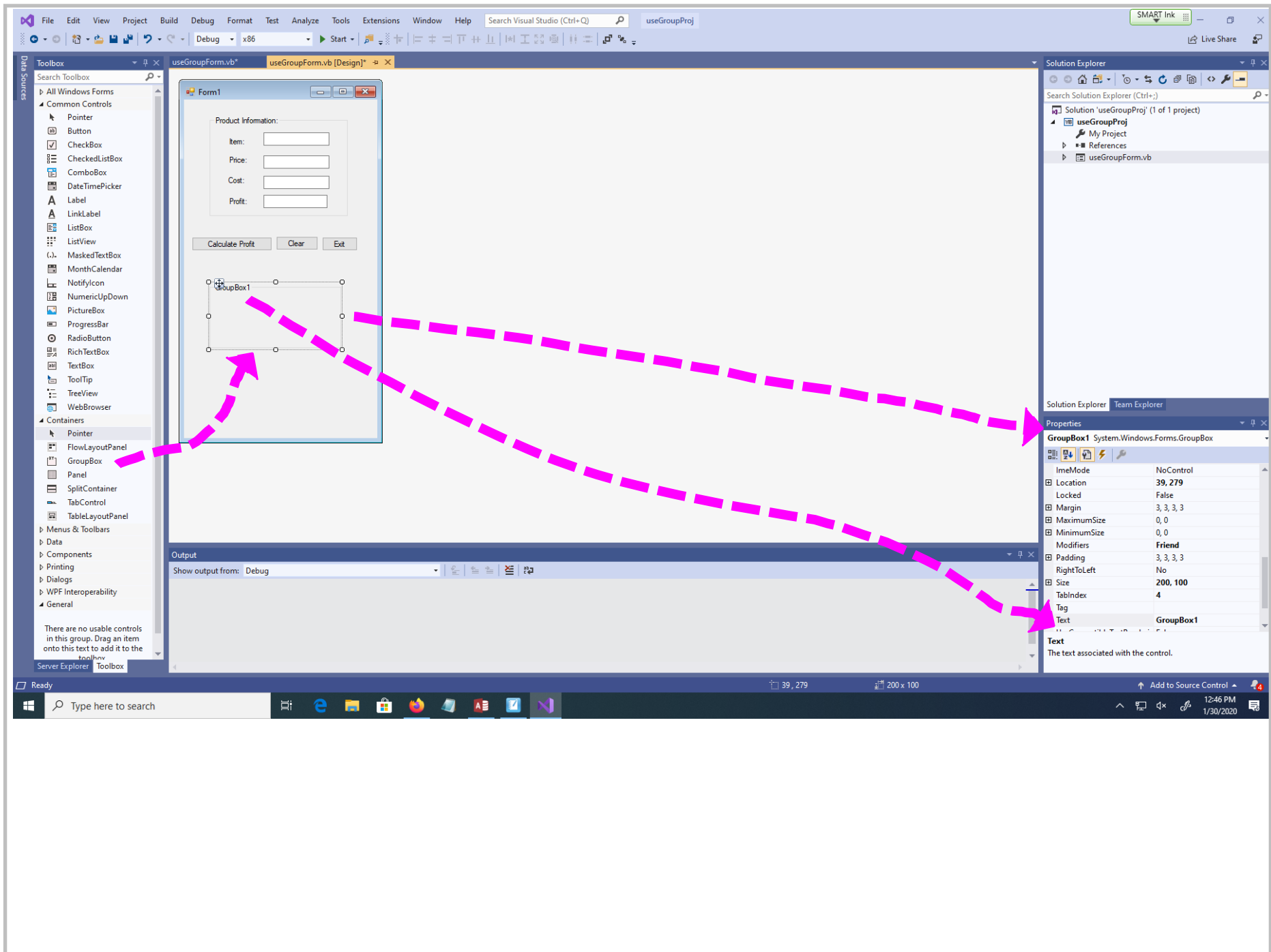
A dialog box titled "Intro Exce" is overlaid on the code, showing the results of the `btnTryCatch_Click` method. The dialog contains the following fields and controls:

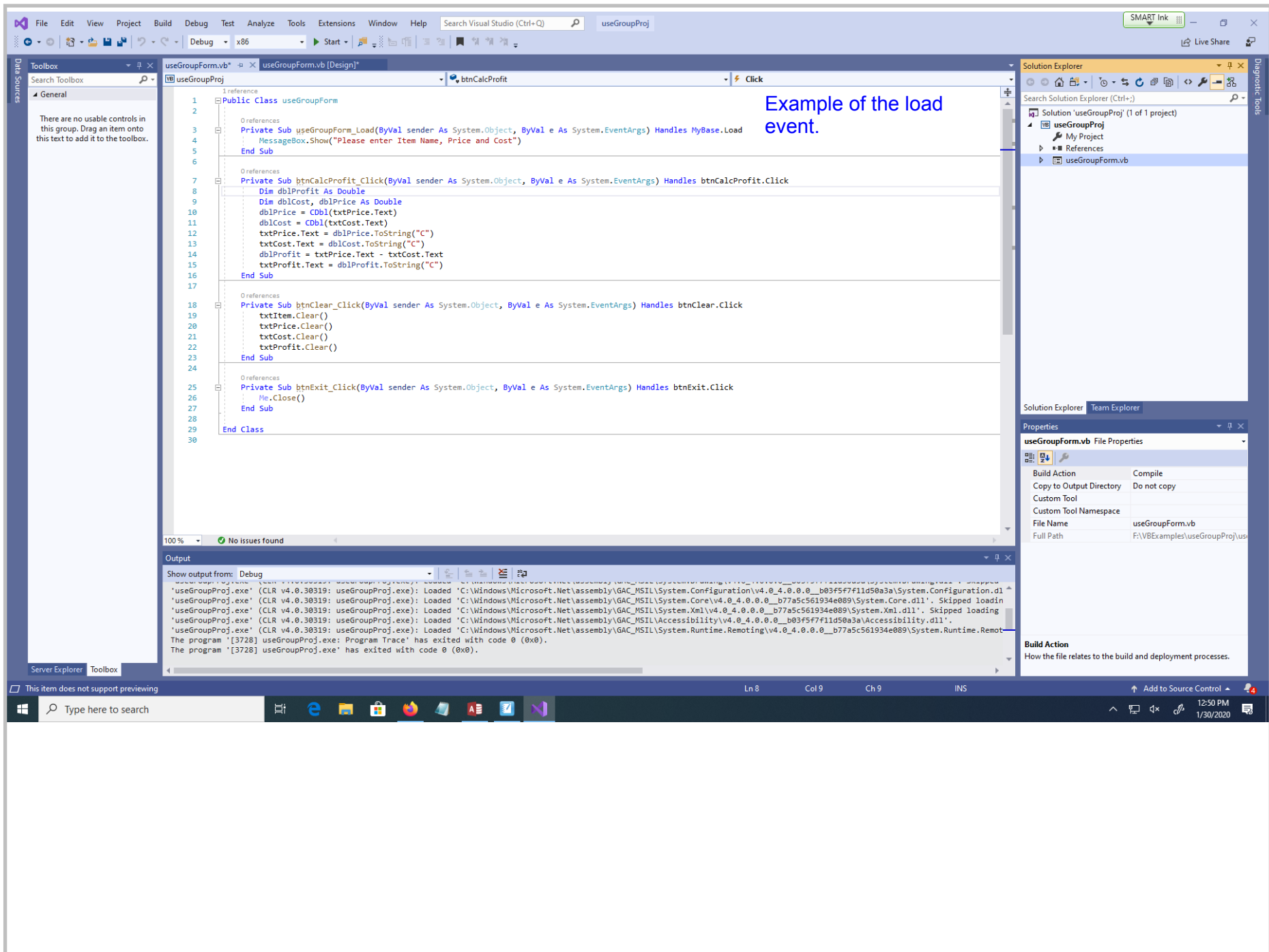
- To Convert:
- Result:
- Buttons:
- Numbers:
- Answer:

A pink dashed arrow points from the `catch` block in the code to the "Answer" field in the dialog box.

The bottom of the screen shows the Windows taskbar with the date and time: 12:43 PM 1/30/2020.







The image shows a Visual Studio IDE window with a VB.NET code file open. The code defines three event handlers for buttons: `btnIntDiv_Click`, `btnMod_Click`, and `btnPower_Click`. Each handler takes text input from text boxes and calculates a result based on mathematical operations (division, modulus, and power), then updates the text of other text boxes. A floating window titled "Math Exar" is overlaid on the code, featuring a "SMART Ink" logo and a simple calculator interface with input fields, a result field, and buttons for various operations: Add, Add no, Subtract, Multiply, Divide, Integer Division, Get Remainder, and Power. The "Get Remainder" button is highlighted. The IDE interface includes a menu bar, a toolbar, a Solution Explorer on the right, and a status bar at the bottom showing "Ready", "Ln 43", "Col 9", "Ch 9", "INS", and the date/time "12:55 PM 1/30/2020".

```
27     wkSecond = txtSecond.Text
28     wkResult = wkFirst / wkSecond
29     txtResult.Text = wkResult
30     txtOther.Text = wkResult
31 End Sub
32
33 References
34 Private Sub btnIntDiv_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnIntDiv.Click
35     Dim wkFirst As Double, wkSecond As Double, wkResult As Double
36     wkFirst = txtFirst.Text
37     wkSecond = txtSecond.Text
38     wkResult = wkFirst \ wkSecond
39     txtResult.Text = wkResult
40     txtOther.Text = wkResult
41 End Sub
42
43 References
44 Private Sub btnMod_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnMod.Click
45     Dim wkFirst As Double, wkSecond As Double, wkResult As Double
46     wkFirst = txtFirst.Text
47     wkSecond = txtSecond.Text
48     wkResult = wkFirst Mod wkSecond
49     txtResult.Text = wkResult
50     txtOther.Text = wkResult
51 End Sub
52
53 References
54 Private Sub btnPower_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnPower.Click
55     Dim wkFirst As Double, wkSecond As Double, wkResult As Double
56     wkFirst = txtFirst.Text
57     wkSecond = txtSecond.Text
58     wkResult = wkFirst ^ wkSecond
59     txtResult.Text = wkResult
60     txtOther.Text = wkResult
61 End Sub
62 End Class
```

