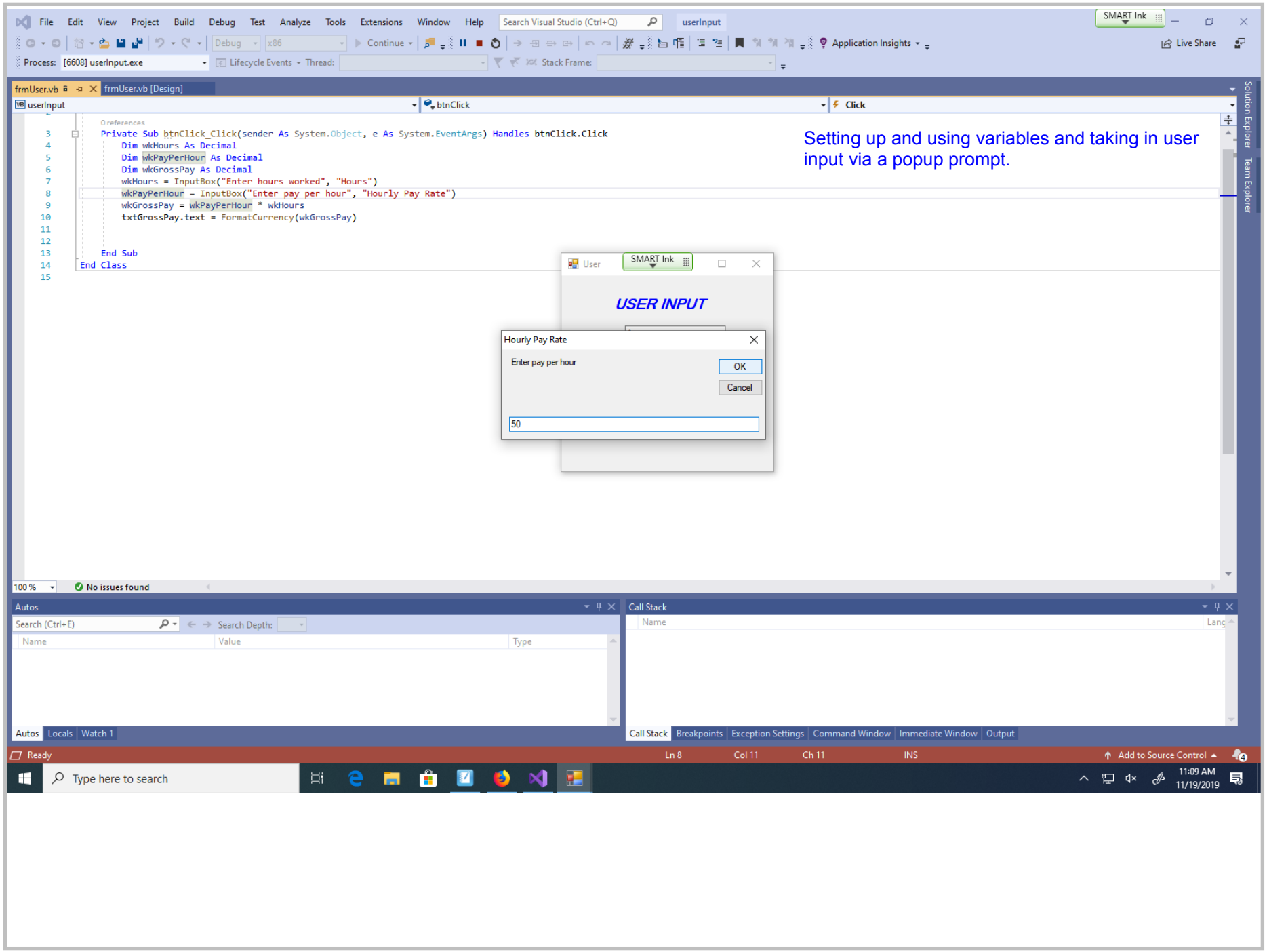
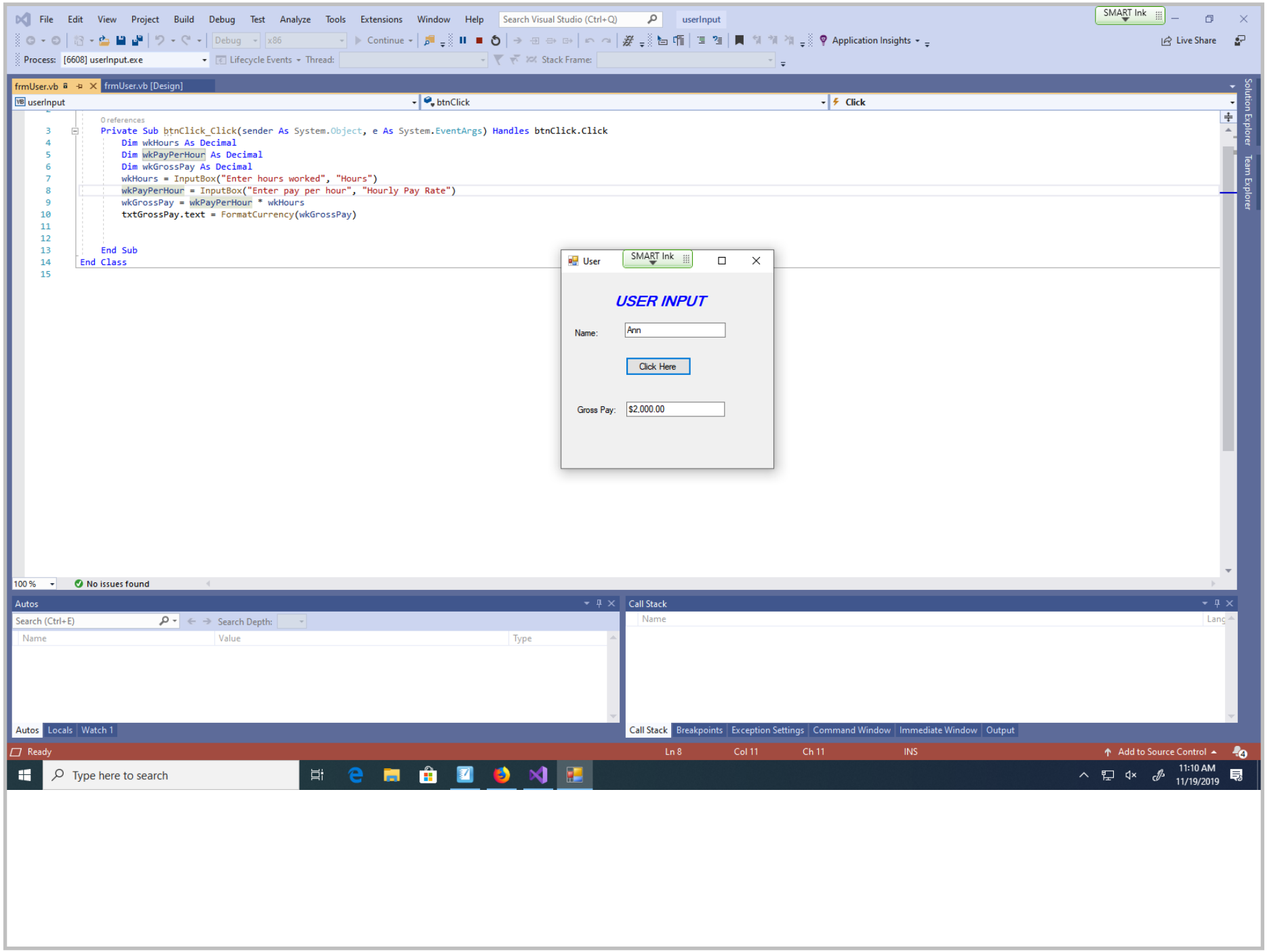
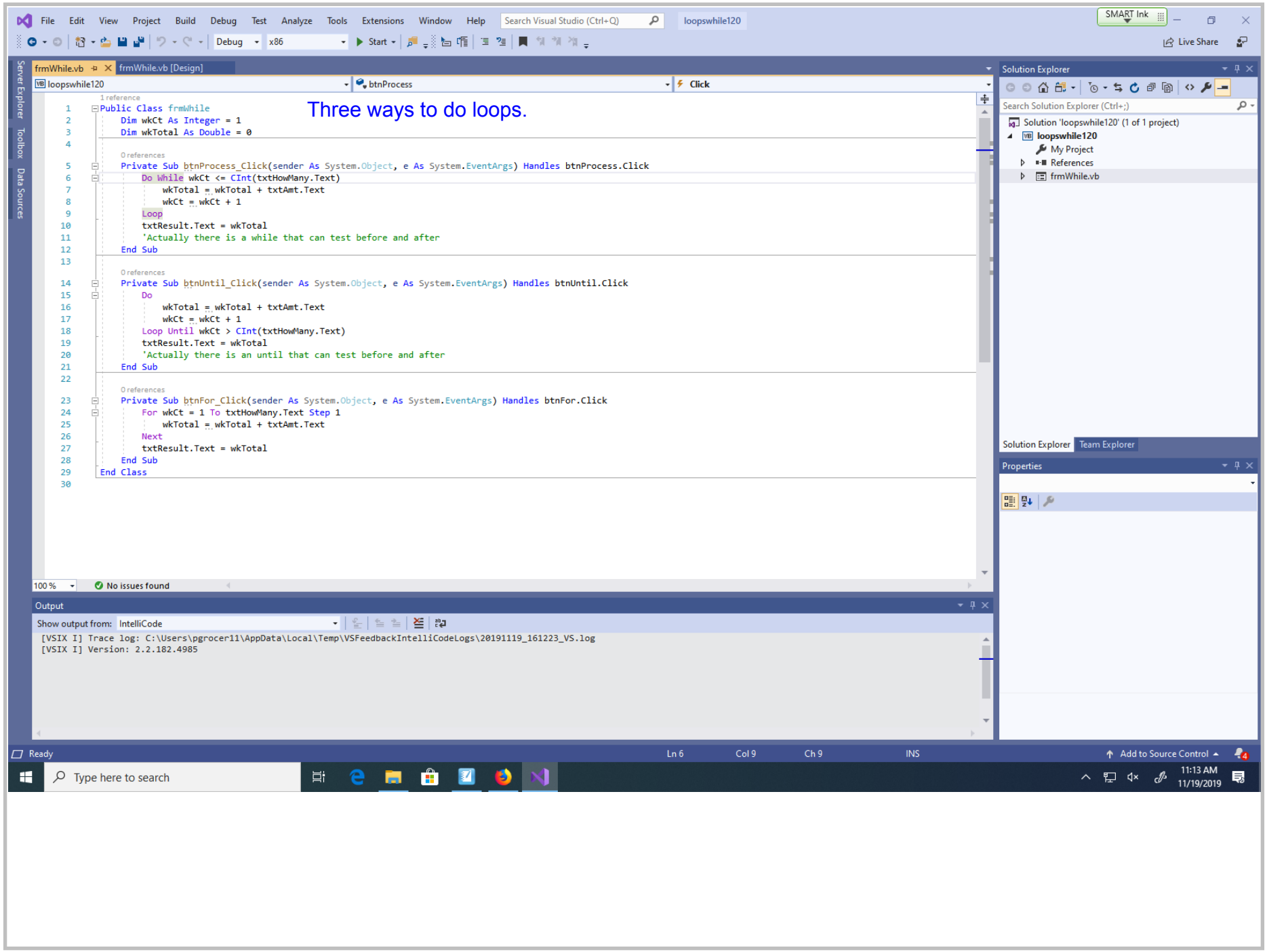
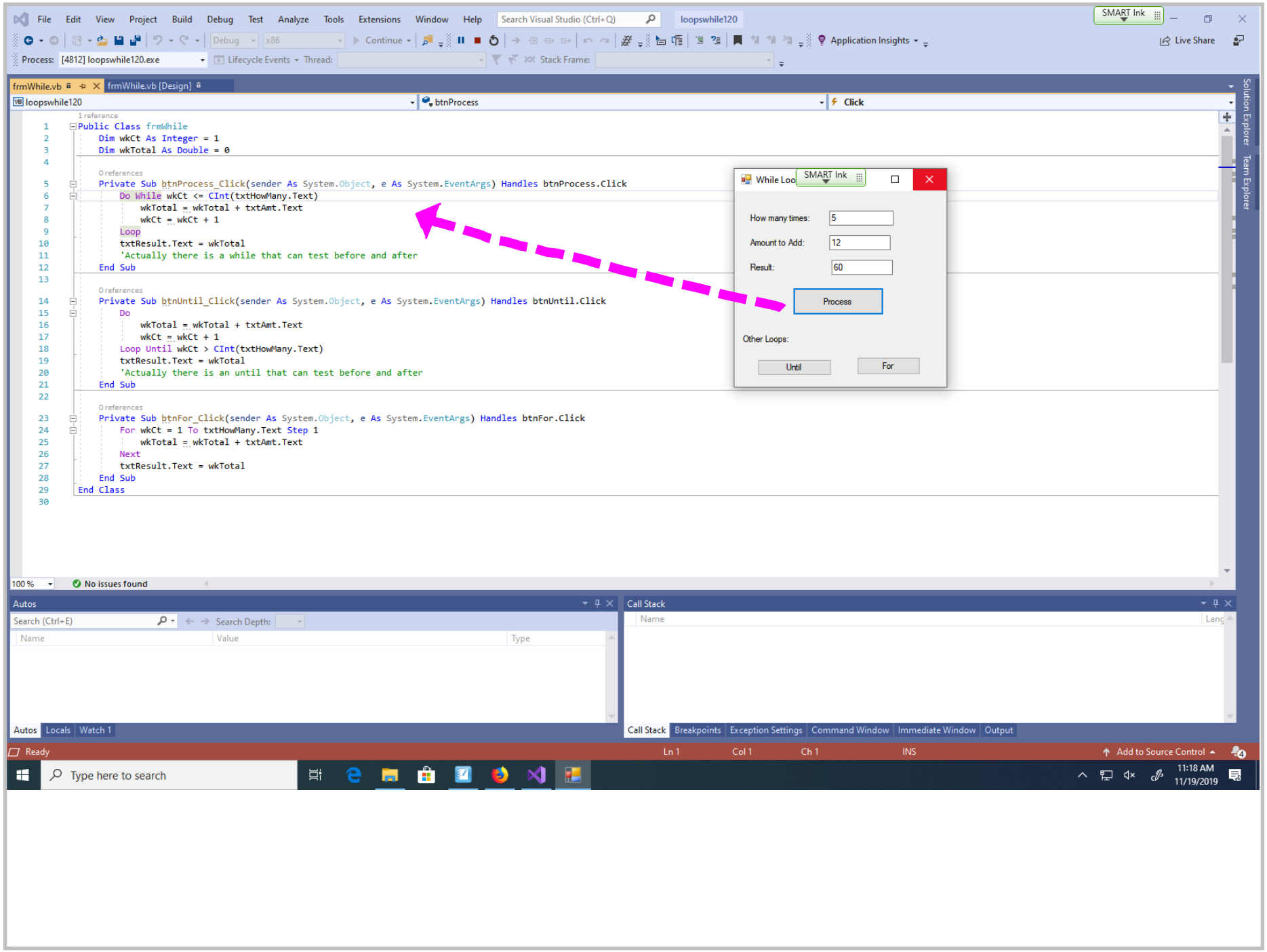
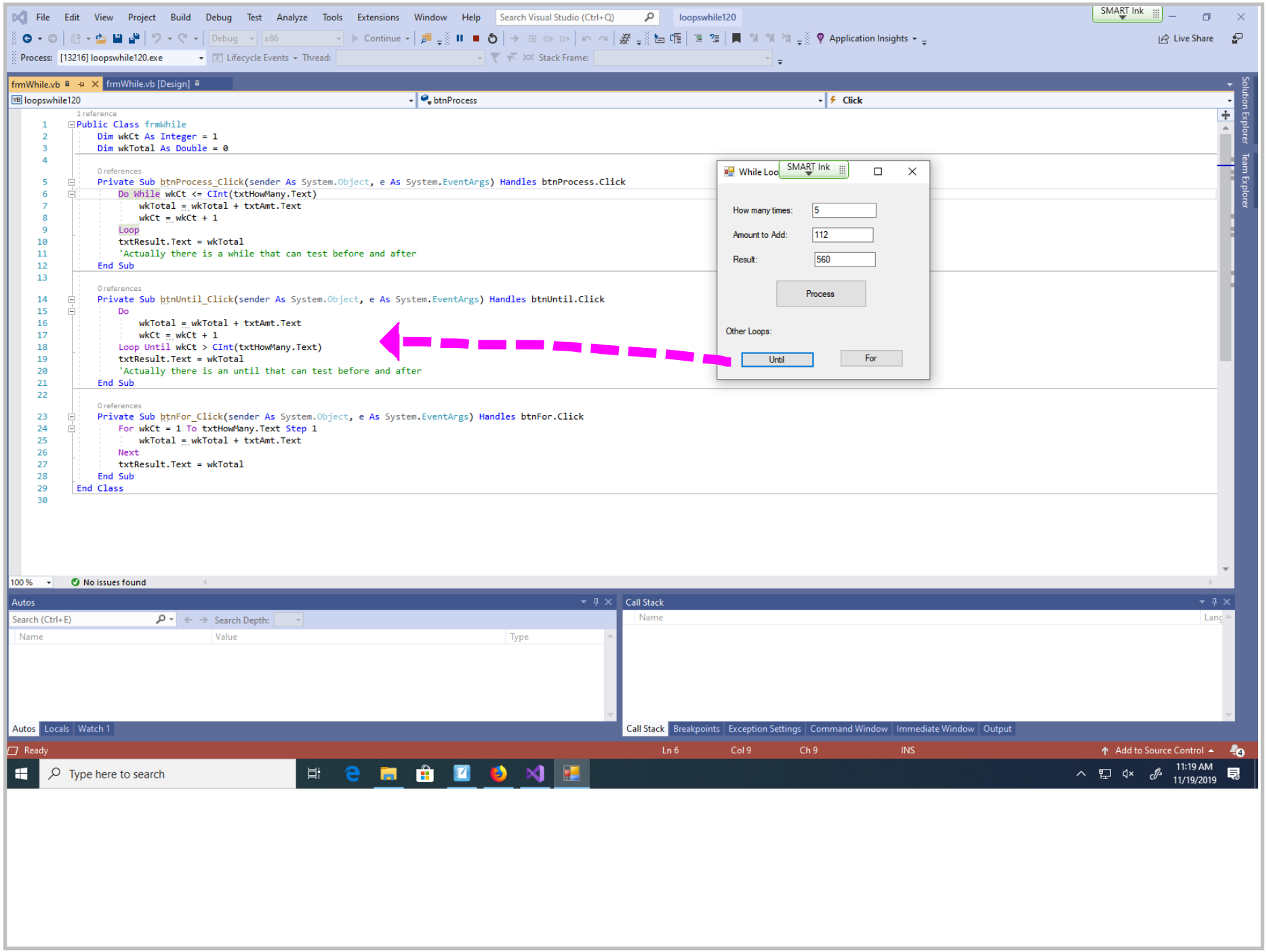

File Edit View Project Build Debug Test Analyze Tools Extensions Window Help
Search Visual Studio (Ctrl+Q)
userInput
SMART Ink
Application Insights
Live Share
frmUser.vb
frmUser.vb [Design]
userInput
btnClick
Click
0 references
Private Sub btnClick_Click(sender As System.Object, e As System.EventArgs) Handles btnClick.Click
Dim wkHours As Decimal
Dim wkPayPerHour As Decimal
Dim wkGrossPay As Decimal
wkHours = InputBox("Enter hours worked", "Hours")
wkPayPerHour = InputBox("Enter pay per hour", "Hourly Pay Rate")
wkGrossPay = wkPayPerHour * wkHours
txtGrossPay.text = FormatCurrency(wkGrossPay)
End Sub
End Class
Setting up and using variables and taking in user input via a popup prompt.
User
SMART Ink
USER INPUT
Hourly Pay Rate
Enter pay per hour
OK
Cancel
50
100 %
No issues found
Autos
Search (Ctrl+E)
Search Depth:
Name
Value
Type
Autos
Locals
Watch 1
Call Stack
Name
Call Stack
Breakpoints
Exception Settings
Command Window
Immediate Window
Output
Ready
Ln 8
Col 11
Ch 11
INS
Add to Source Control
Type here to search
11:09 AM
11/19/2019

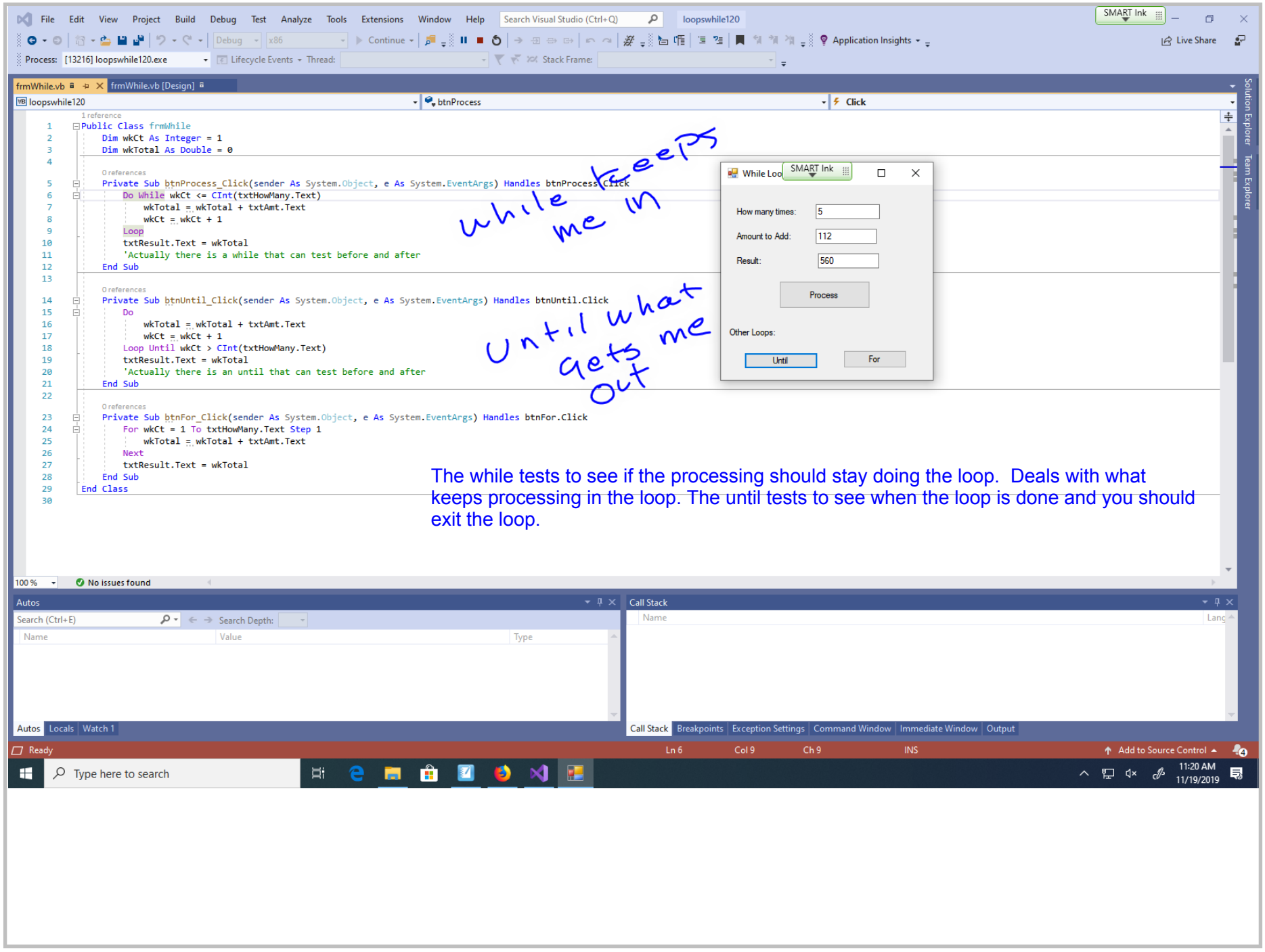

File Edit View Project Build Debug Test Analyze Tools Extensions Window Help
Search Visual Studio (Ctrl+Q)
userInput
SMART Ink
Application Insights
Live Share
Process: [6608] userInput.exe
Lifecycle Events
Thread:
Stack Frame:
frmUser.vb
frmUser.vb [Design]
userInput
btnClick
Click
0 references
Private Sub btnClick_Click(sender As System.Object, e As System.EventArgs) Handles btnClick.Click
Dim wkHours As Decimal
Dim wkPayPerHour As Decimal
Dim wkGrossPay As Decimal
wkHours = InputBox("Enter hours worked", "Hours")
wkPayPerHour = InputBox("Enter pay per hour", "Hourly Pay Rate")
wkGrossPay = wkPayPerHour * wkHours
txtGrossPay.text = FormatCurrency(wkGrossPay)
End Sub
End Class
User
SMART Ink
USER INPUT
Name:
Ann
Click Here
Gross Pay:
$2,000.00
100 %
No issues found
Autos
Search (Ctrl+E)
Search Depth:
Name
Value
Type
Autos
Locals
Watch 1
Call Stack
Name
Call Stack
Breakpoints
Exception Settings
Command Window
Immediate Window
Output
Ready
Ln 8
Col 11
Ch 11
INS
Add to Source Control
Type here to search
11:10 AM
11/19/2019

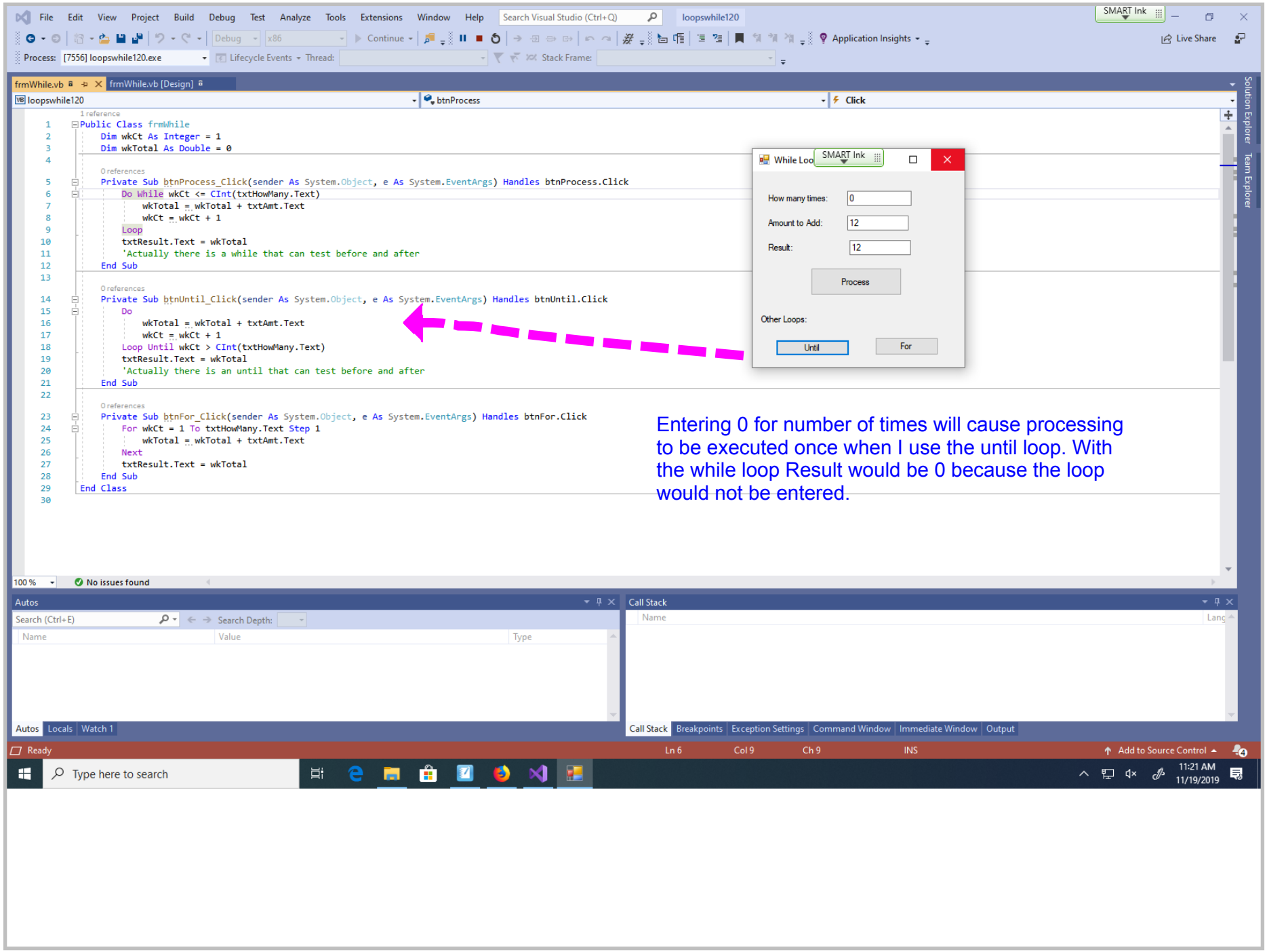

Three ways to do loops.
Public Class frmWhile
Dim wkCt As Integer = 1
Dim wkTotal As Double = 0
Private Sub btnProcess_Click(sender As System.Object, e As System.EventArgs) Handles btnProcess.Click
Do While wkCt <= CInt(txtHowMany.Text)
wkTotal = wkTotal + txtAmt.Text
wkCt = wkCt + 1
Loop
txtResult.Text = wkTotal
'Actually there is a while that can test before and after
End Sub
Private Sub btnUntil_Click(sender As System.Object, e As System.EventArgs) Handles btnUntil.Click
Do
wkTotal = wkTotal + txtAmt.Text
wkCt = wkCt + 1
Loop Until wkCt > CInt(txtHowMany.Text)
txtResult.Text = wkTotal
'Actually there is an until that can test before and after
End Sub
Private Sub btnFor_Click(sender As System.Object, e As System.EventArgs) Handles btnFor.Click
For wkCt = 1 To txtHowMany.Text Step 1
wkTotal = wkTotal + txtAmt.Text
Next
txtResult.Text = wkTotal
End Sub
End Class
[VSIX I] Trace log: C:\Users\pgrocer11\AppData\Local\Temp\VSFeedbackIntelliCodeLogs\20191119_161223_VS.log
[VSIX I] Version: 2.2.182.4985

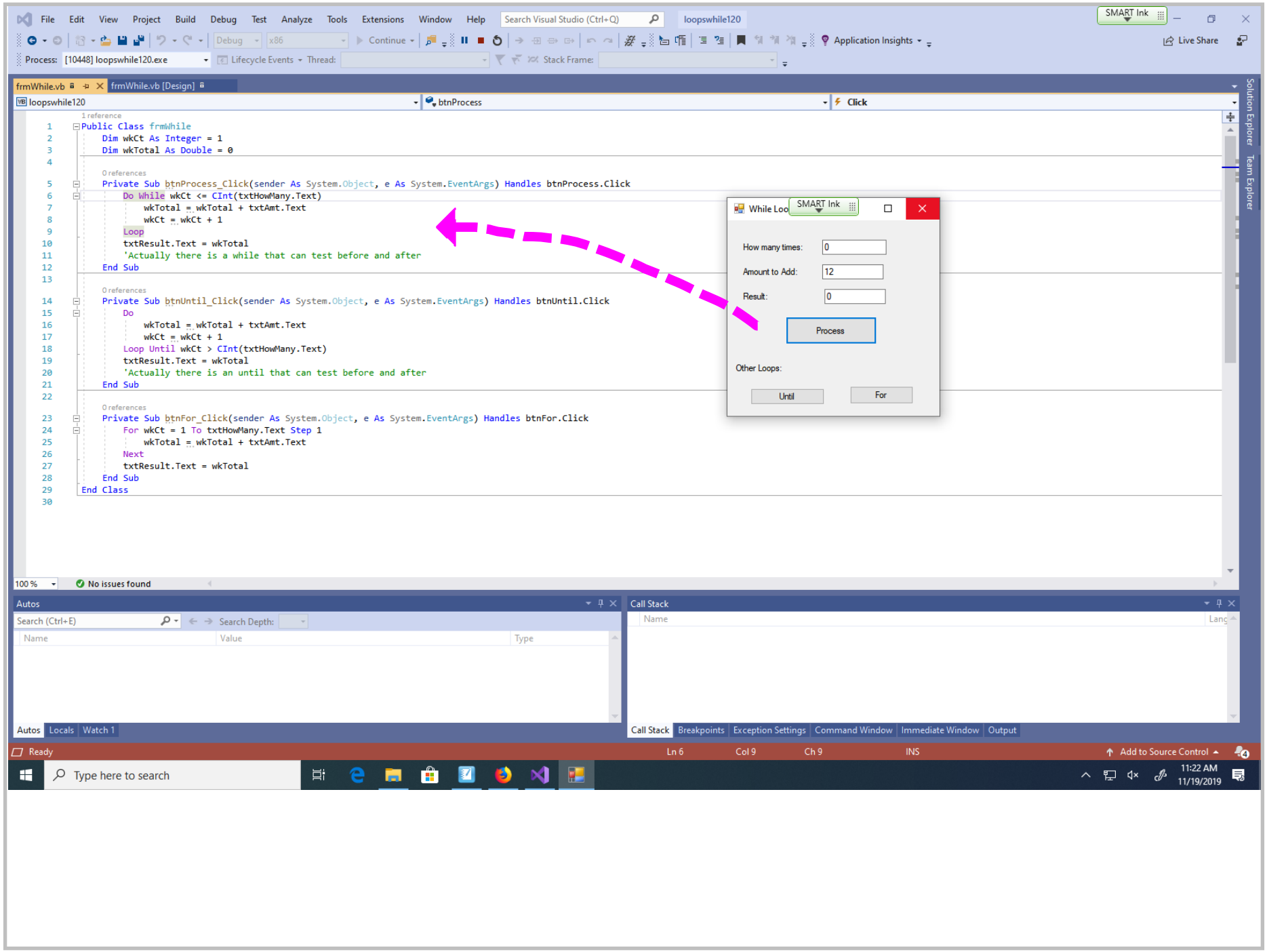


Title: Nov 19-11:18 AM (4 of 20)

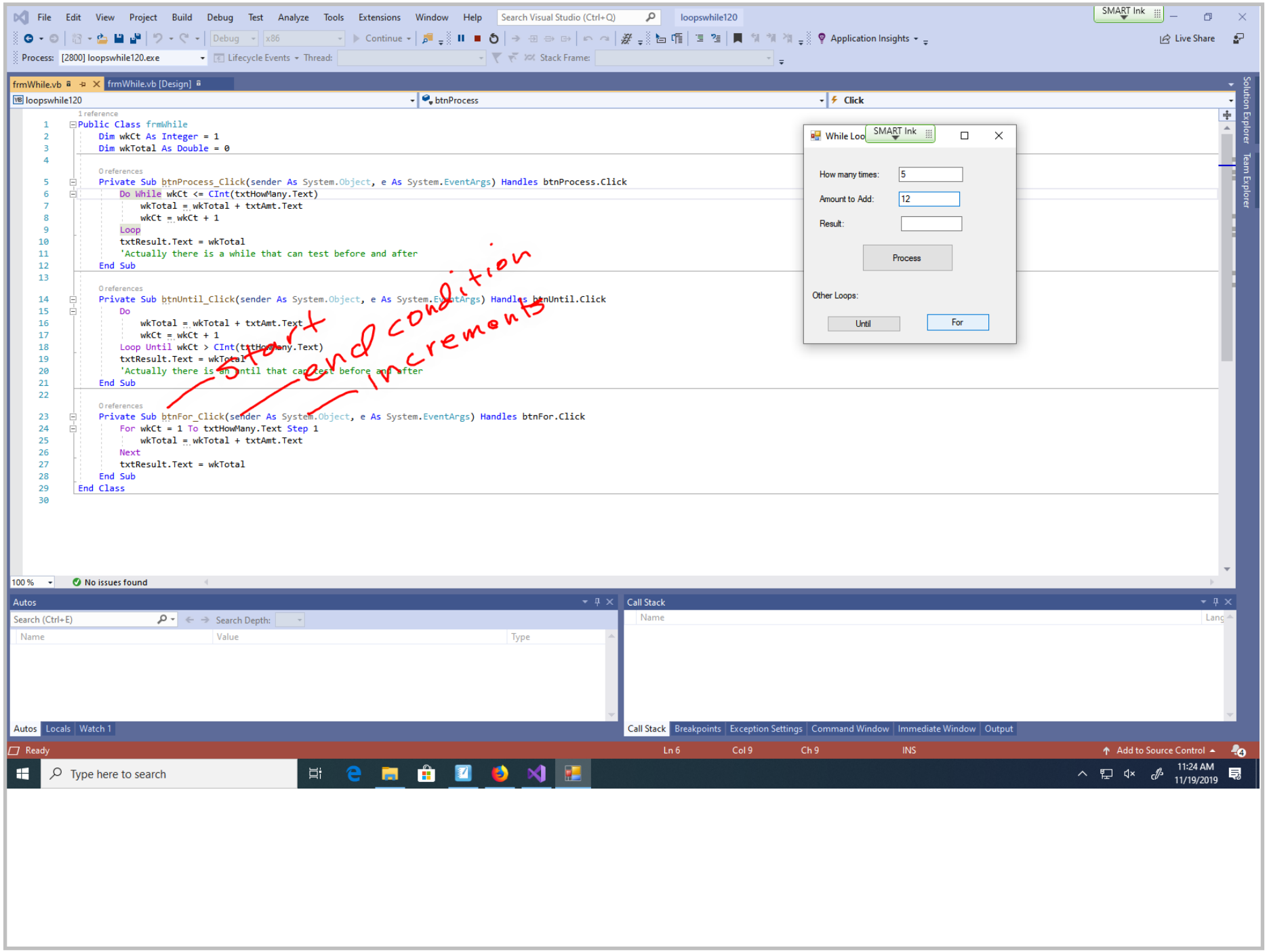

File Edit View Project Build Debug Test Analyze Tools Extensions Window Help
Search Visual Studio (Ctrl+Q)
loopswhile120
SMART Ink
Application Insights
Live Share
Process: [13216] loopswhile120.exe
Lifecycle Events
Thread:
Stack Frame:
frmWhile.vb
frmWhile.vb [Design]
loopswhile120
btnProcess
Click
1 reference
Public Class frmWhile
Dim wkCt As Integer = 1
Dim wkTotal As Double = 0
0 references
Private Sub btnProcess_Click(sender As System.Object, e As System.EventArgs) Handles btnProcess.Click
Do While wkCt <= CInt(txtHowMany.Text)
wkTotal = wkTotal + txtAmt.Text
wkCt = wkCt + 1
Loop
txtResult.Text = wkTotal
'Actually there is a while that can test before and after
End Sub
0 references
Private Sub btnUntil_Click(sender As System.Object, e As System.EventArgs) Handles btnUntil.Click
Do
wkTotal = wkTotal + txtAmt.Text
wkCt = wkCt + 1
Loop Until wkCt > CInt(txtHowMany.Text)
txtResult.Text = wkTotal
'Actually there is an until that can test before and after
End Sub
0 references
Private Sub btnFor_Click(sender As System.Object, e As System.EventArgs) Handles btnFor.Click
For wkCt = 1 To txtHowMany.Text Step 1
wkTotal = wkTotal + txtAmt.Text
Next
txtResult.Text = wkTotal
End Sub
End Class
While Loo
SMART Ink
How many times:
5
Amount to Add:
112
Result:
560
Process
Other Loops:
Until
For
100 %
No issues found
Autos
Search (Ctrl+E)
Search Depth:
Name
Value
Type
Autos
Locals
Watch 1
Call Stack
Name
Call Stack
Breakpoints
Exception Settings
Command Window
Immediate Window
Output
Ready
Ln 6
Col 9
Ch 9
INS
Add to Source Control
Type here to search
11:19 AM
11/19/2019

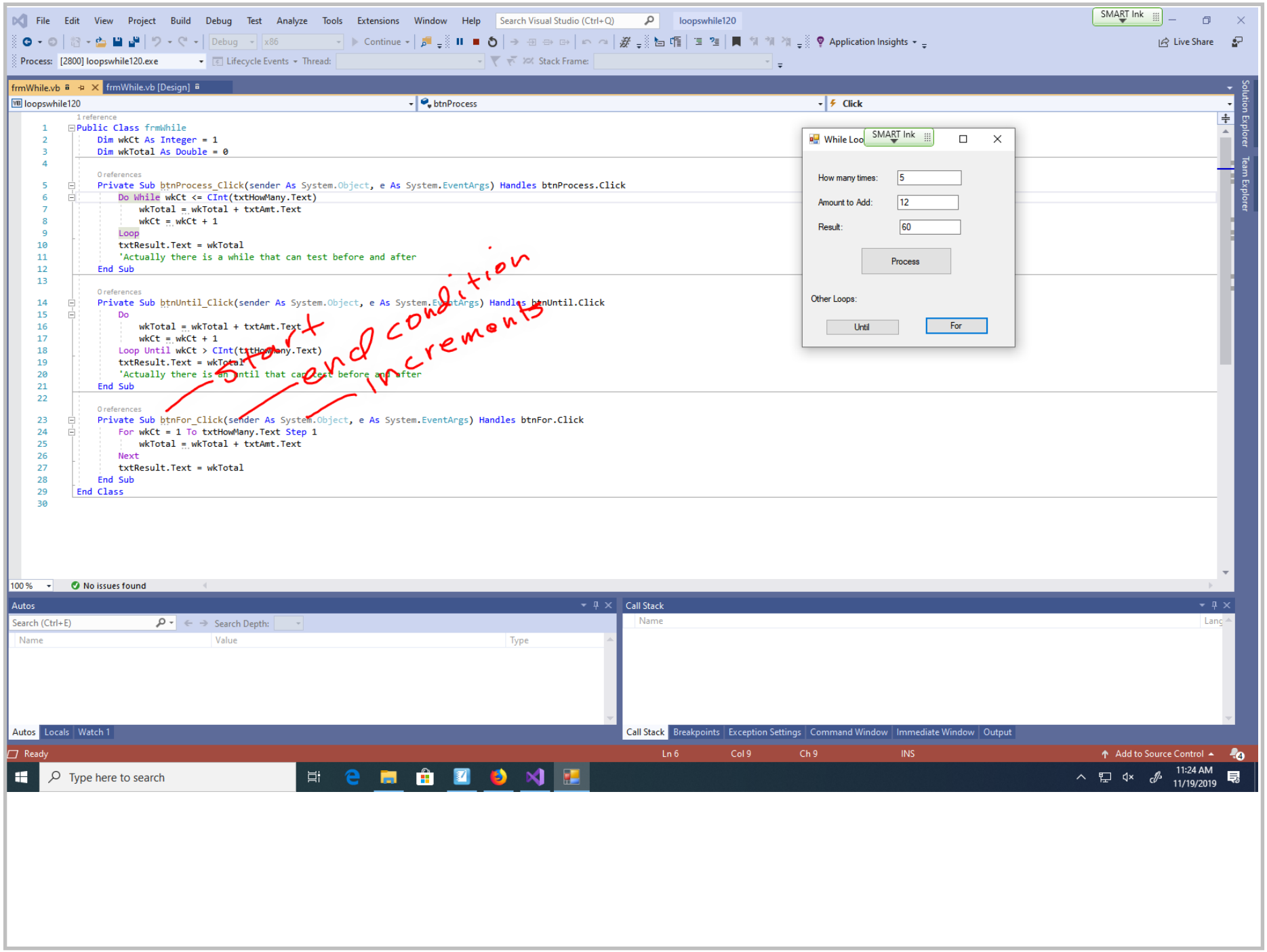

frmWhile.vb
frmWhile.vb [Design]
loopswhile120
btnProcess
Click
Public Class frmWhile
Dim wkCt As Integer = 1
Dim wkTotal As Double = 0
Private Sub btnProcess_Click(sender As System.Object, e As System.EventArgs) Handles btnProcess.Click
Do While wkCt <= CInt(txtHowMany.Text)
wkTotal = wkTotal + txtAmt.Text
wkCt = wkCt + 1
Loop
txtResult.Text = wkTotal
'Actually there is a while that can test before and after
End Sub
Private Sub btnUntil_Click(sender As System.Object, e As System.EventArgs) Handles btnUntil.Click
Do
wkTotal = wkTotal + txtAmt.Text
wkCt = wkCt + 1
Loop Until wkCt > CInt(txtHowMany.Text)
txtResult.Text = wkTotal
'Actually there is an until that can test before and after
End Sub
Private Sub btnFor_Click(sender As System.Object, e As System.EventArgs) Handles btnFor.Click
For wkCt = 1 To txtHowMany.Text Step 1
wkTotal = wkTotal + txtAmt.Text
Next
txtResult.Text = wkTotal
End Sub
End Class
while keeps me in
Until what gets me out
While Loo
SMART Ink
How many times: 5
Amount to Add: 112
Result: 560
Process
Other Loops:
Until
For
The while tests to see if the processing should stay doing the loop. Deals with what keeps processing in the loop. The until tests to see when the loop is done and you should exit the loop.
Autos
Call Stack
Ready
Type here to search
11:20 AM
11/19/2019

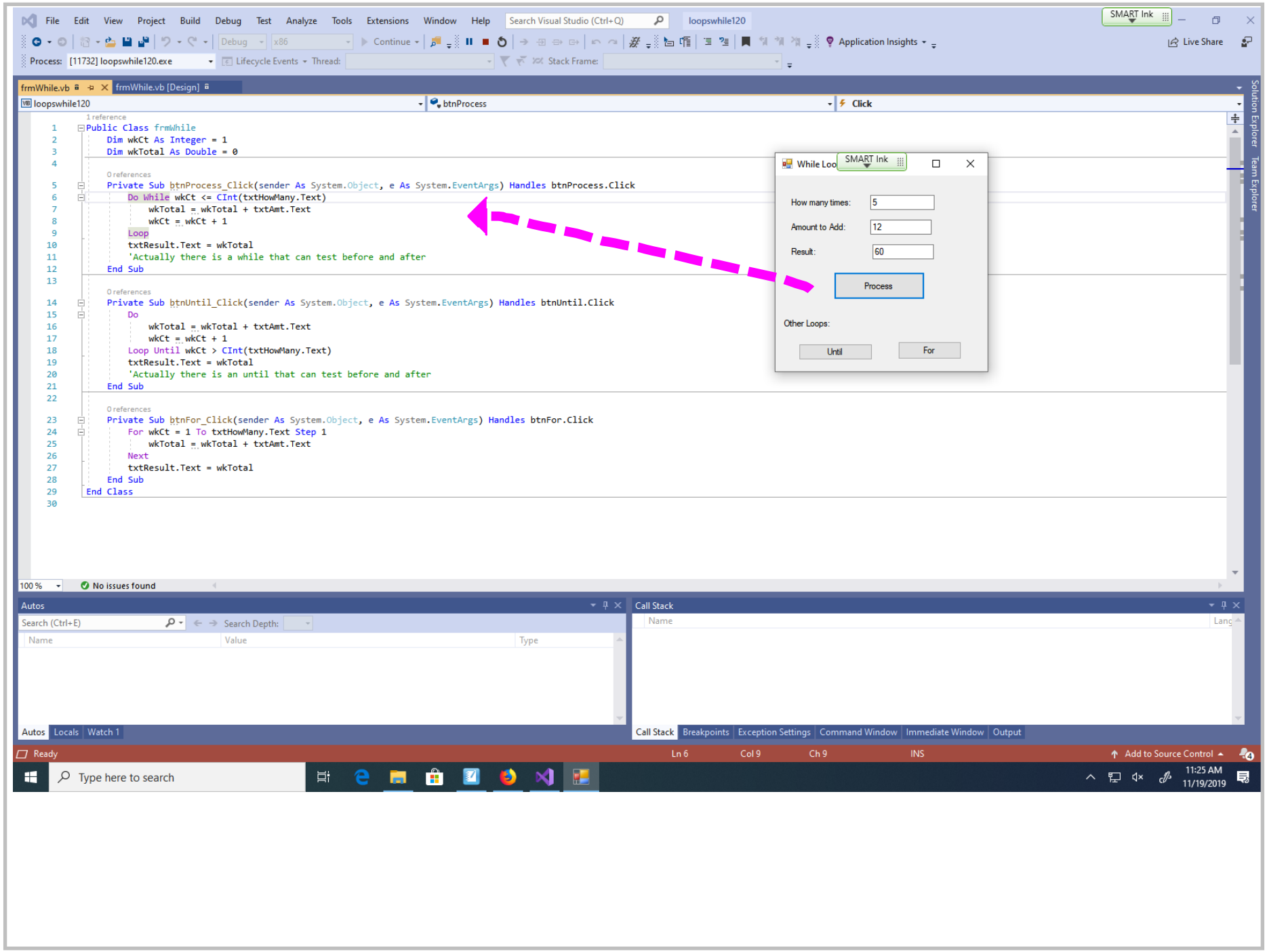

frmWhile.vb
frmWhile.vb [Design]
loopswhile120
btnProcess
Click
Public Class frmWhile
Dim wkCt As Integer = 1
Dim wkTotal As Double = 0
Private Sub btnProcess_Click(sender As System.Object, e As System.EventArgs) Handles btnProcess.Click
Do While wkCt <= CInt(txtHowMany.Text)
wkTotal = wkTotal + txtAmt.Text
wkCt = wkCt + 1
Loop
txtResult.Text = wkTotal
'Actually there is a while that can test before and after
End Sub
Private Sub btnUntil_Click(sender As System.Object, e As System.EventArgs) Handles btnUntil.Click
Do
wkTotal = wkTotal + txtAmt.Text
wkCt = wkCt + 1
Loop Until wkCt > CInt(txtHowMany.Text)
txtResult.Text = wkTotal
'Actually there is an until that can test before and after
End Sub
Private Sub btnFor_Click(sender As System.Object, e As System.EventArgs) Handles btnFor.Click
For wkCt = 1 To txtHowMany.Text Step 1
wkTotal = wkTotal + txtAmt.Text
Next
txtResult.Text = wkTotal
End Sub
End Class
While Loo
SMART Ink
How many times:
0
Amount to Add:
12
Result:
12
Process
Other Loops:
Until
For
Entering 0 for number of times will cause processing to be executed once when I use the until loop. With the while loop Result would be 0 because the loop would not be entered.
No issues found
Autos
Call Stack
Ready
Type here to search
11:21 AM
11/19/2019

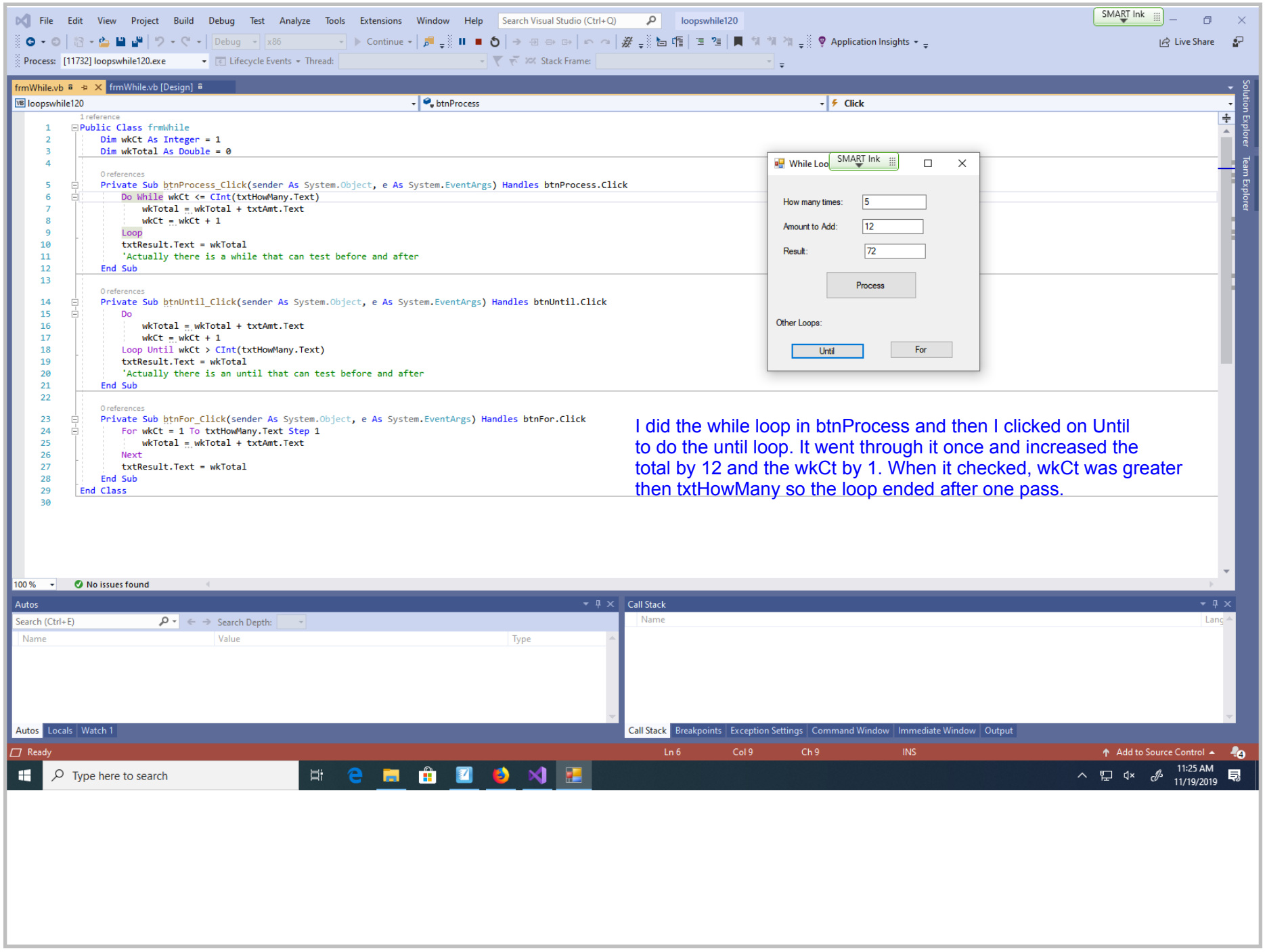


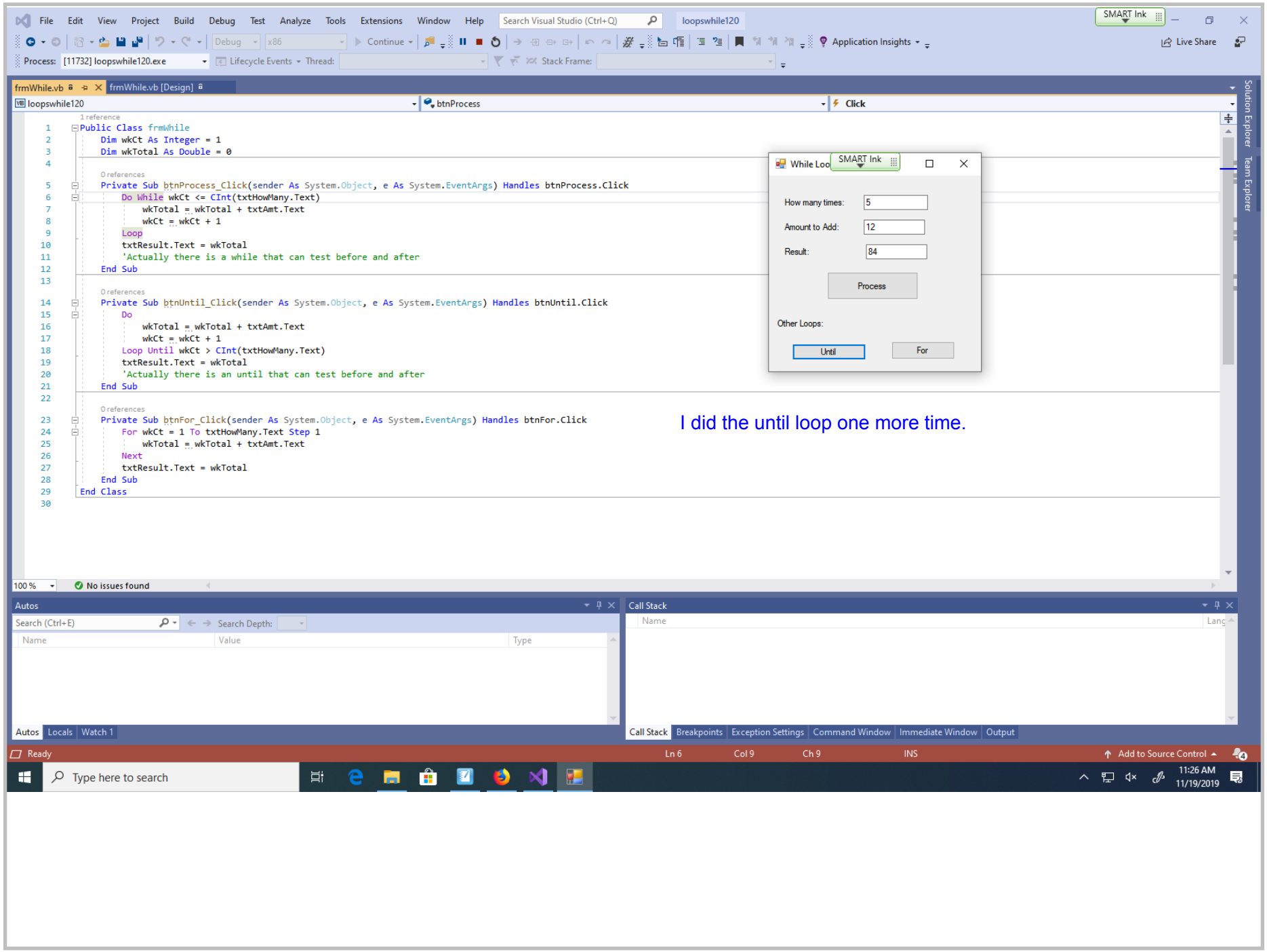
Title: Nov 19-11:22 AM (8 of 20)

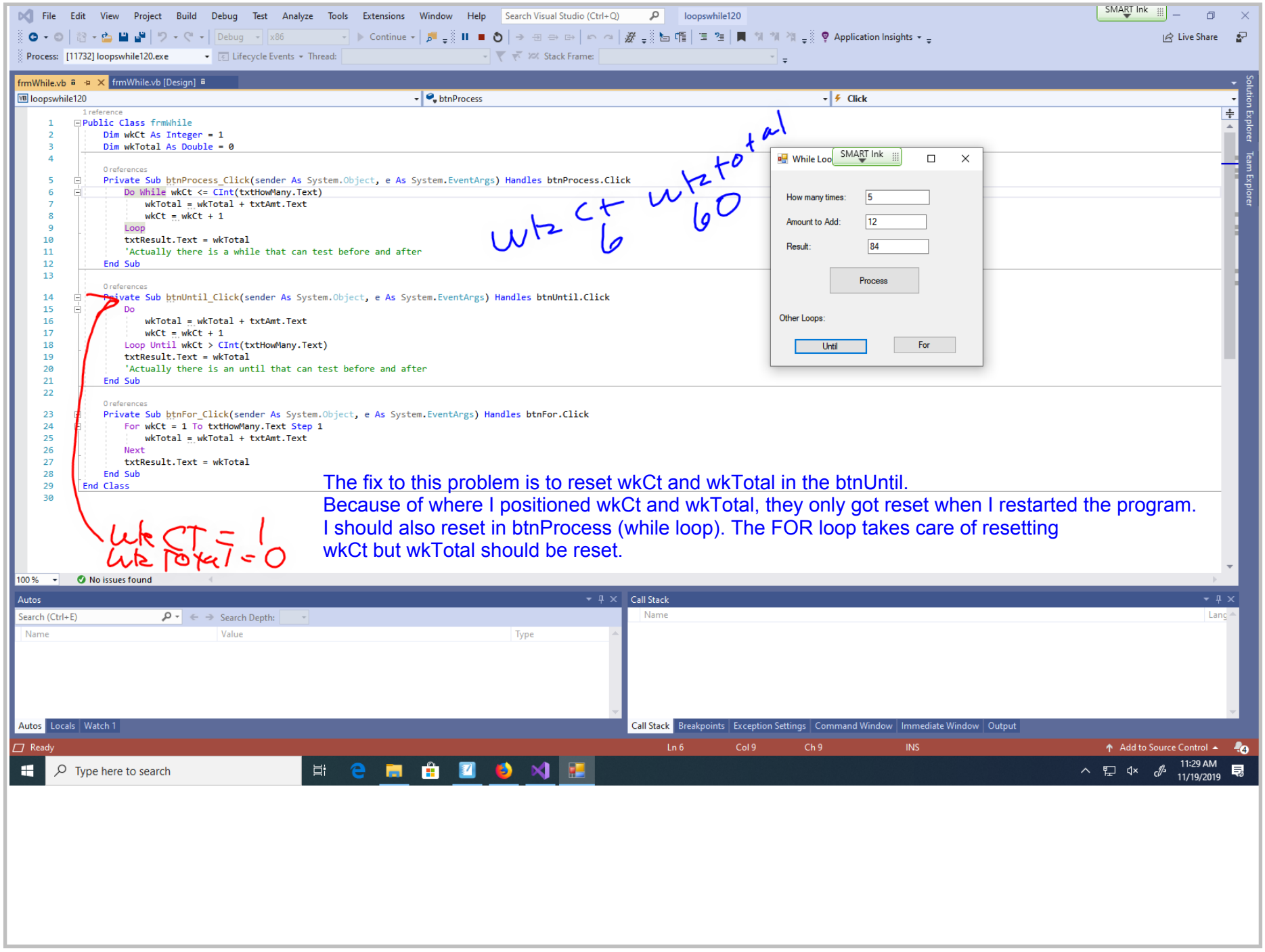

File Edit View Project Build Debug Test Analyze Tools Extensions Window Help
loopswhile120
Application Insights
Live Share
frmWhile.vb
frmWhile.vb [Design]
loopswhile120
btnProcess
Click
1 reference
Public Class frmWhile
Dim wkCt As Integer = 1
Dim wkTotal As Double = 0
0 references
Private Sub btnProcess_Click(sender As System.Object, e As System.EventArgs) Handles btnProcess.Click
Do While wkCt <= CInt(txtHowMany.Text)
wkTotal = wkTotal + txtAmt.Text
wkCt = wkCt + 1
Loop
txtResult.Text = wkTotal
'Actually there is a while that can test before and after
End Sub
0 references
Private Sub btnUntil_Click(sender As System.Object, e As System.EventArgs) Handles btnUntil.Click
Do
wkTotal = wkTotal + txtAmt.Text
wkCt = wkCt + 1
Loop Until wkCt > CInt(txtHowMany.Text)
txtResult.Text = wkTotal
'Actually there is an until that can test before and after
End Sub
0 references
Private Sub btnFor_Click(sender As System.Object, e As System.EventArgs) Handles btnFor.Click
For wkCt = 1 To txtHowMany.Text Step 1
wkTotal = wkTotal + txtAmt.Text
Next
txtResult.Text = wkTotal
End Sub
End Class
start
end condition
increments
While Loo
SMART Ink
How many times:
5
Amount to Add:
12
Result:
Process
Other Loops:
Until
For
No issues found
Autos
Call Stack
Ready
Ln 6
Col 9
Ch 9
INS
Add to Source Control
Type here to search
11:24 AM
11/19/2019

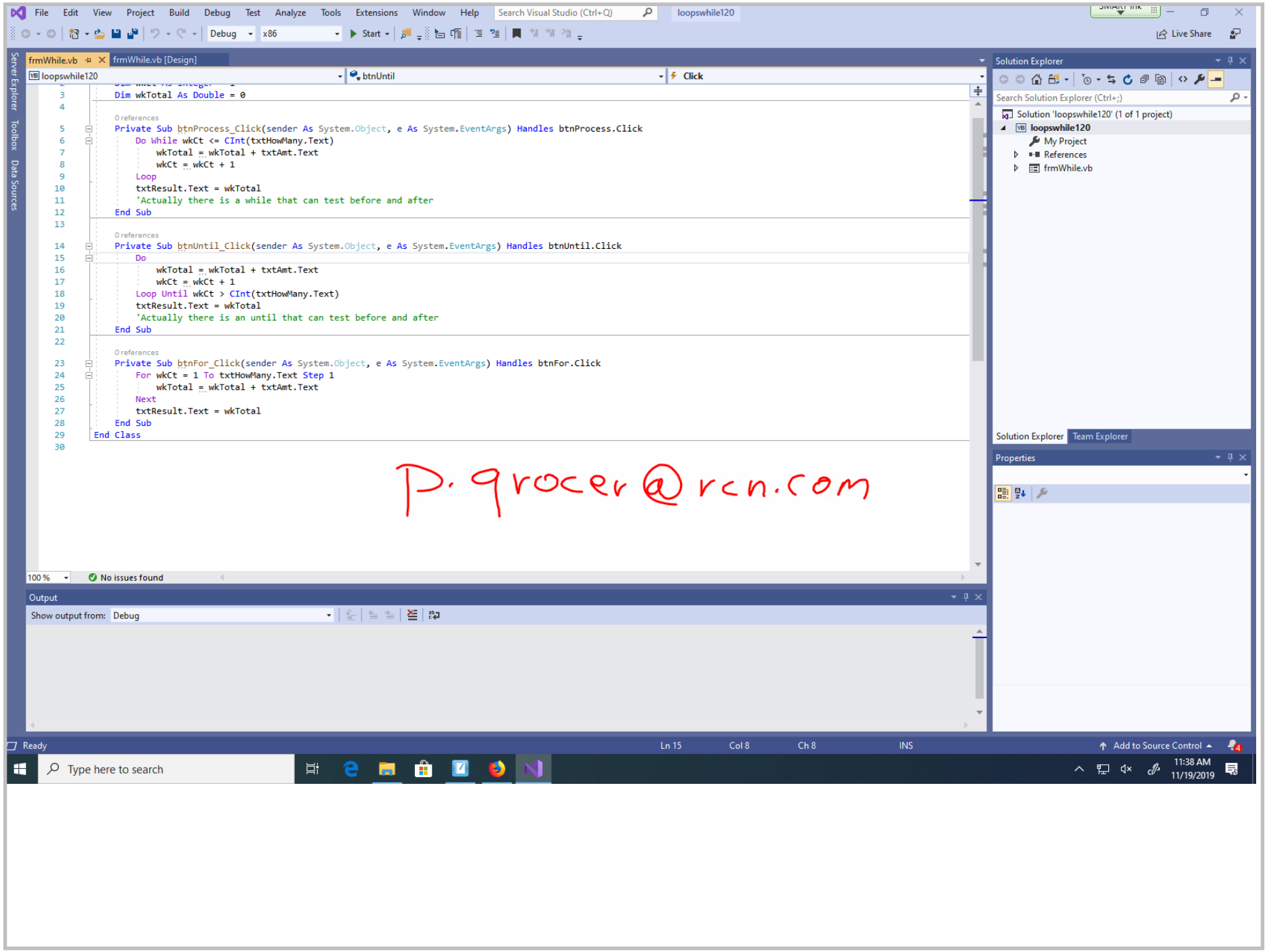

File Edit View Project Build Debug Test Analyze Tools Extensions Window Help
loopswhile120
frmWhile.vb
frmWhile.vb [Design]
loopswhile120
btnProcess
Click
Public Class frmWhile
Dim wkCt As Integer = 1
Dim wkTotal As Double = 0
Private Sub btnProcess_Click(sender As System.Object, e As System.EventArgs) Handles btnProcess.Click
Do While wkCt <= CInt(txtHowMany.Text)
wkTotal = wkTotal + txtAmt.Text
wkCt = wkCt + 1
Loop
txtResult.Text = wkTotal
'Actually there is a while that can test before and after
End Sub
Private Sub btnUntil_Click(sender As System.Object, e As System.EventArgs) Handles btnUntil.Click
Do
wkTotal = wkTotal + txtAmt.Text
wkCt = wkCt + 1
Loop Until wkCt > CInt(txtHowMany.Text)
txtResult.Text = wkTotal
'Actually there is an until that can test before and after
End Sub
Private Sub btnFor_Click(sender As System.Object, e As System.EventArgs) Handles btnFor.Click
For wkCt = 1 To txtHowMany.Text Step 1
wkTotal = wkTotal + txtAmt.Text
Next
txtResult.Text = wkTotal
End Sub
End Class
start
end condition
increments
While Loo
SMART Ink
How many times: 5
Amount to Add: 12
Result: 60
Process
Other Loops:
Until
For
No issues found
Autos
Call Stack
Ready
Ln 6 Col 9 Ch 9 INS
Add to Source Control
Type here to search
11:24 AM
11/19/2019

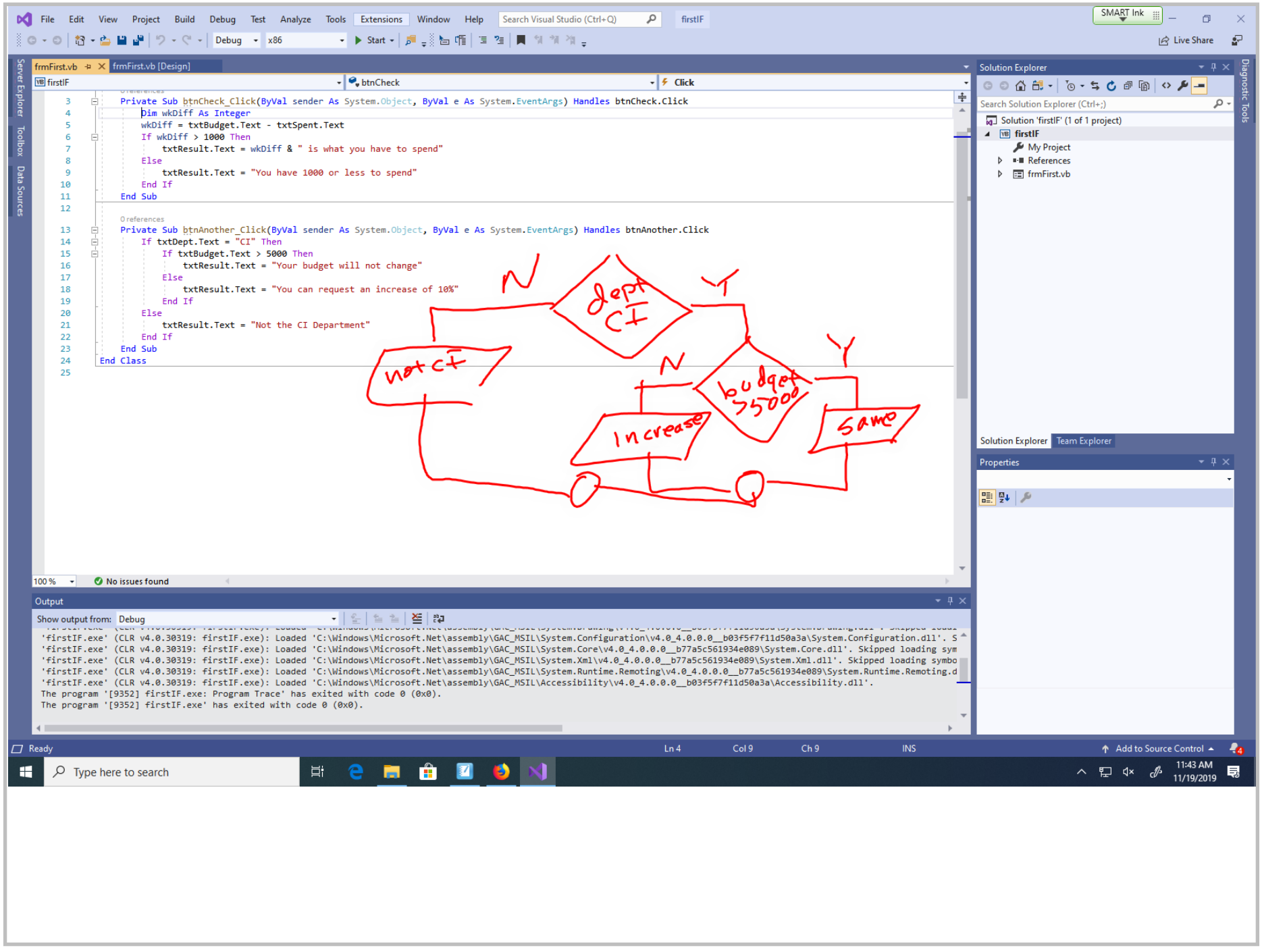

File Edit View Project Build Debug Test Analyze Tools Extensions Window Help
Search Visual Studio (Ctrl+Q)
loopswhile120
SMART Ink
Application Insights
Live Share
frmWhile.vb
frmWhile.vb [Design]
loopswhile120
btnProcess
Click
1 reference
Public Class frmWhile
Dim wkCt As Integer = 1
Dim wkTotal As Double = 0
0 references
Private Sub btnProcess_Click(sender As System.Object, e As System.EventArgs) Handles btnProcess.Click
Do While wkCt <= CInt(txtHowMany.Text)
wkTotal = wkTotal + txtAmt.Text
wkCt = wkCt + 1
Loop
txtResult.Text = wkTotal
'Actually there is a while that can test before and after
End Sub
0 references
Private Sub btnUntil_Click(sender As System.Object, e As System.EventArgs) Handles btnUntil.Click
Do
wkTotal = wkTotal + txtAmt.Text
wkCt = wkCt + 1
Loop Until wkCt > CInt(txtHowMany.Text)
txtResult.Text = wkTotal
'Actually there is an until that can test before and after
End Sub
0 references
Private Sub btnFor_Click(sender As System.Object, e As System.EventArgs) Handles btnFor.Click
For wkCt = 1 To txtHowMany.Text Step 1
wkTotal = wkTotal + txtAmt.Text
Next
txtResult.Text = wkTotal
End Sub
End Class
While Loo
SMART Ink
How many times:
5
Amount to Add:
12
Result:
60
Process
Other Loops:
Until
For
100 %
No issues found
Autos
Search (Ctrl+E)
Search Depth:
Name
Value
Type
Autos
Locals
Watch 1
Call Stack
Name
Call Stack
Breakpoints
Exception Settings
Command Window
Immediate Window
Output
Ready
Ln 6
Col 9
Ch 9
INS
Add to Source Control
Type here to search
11:25 AM
11/19/2019

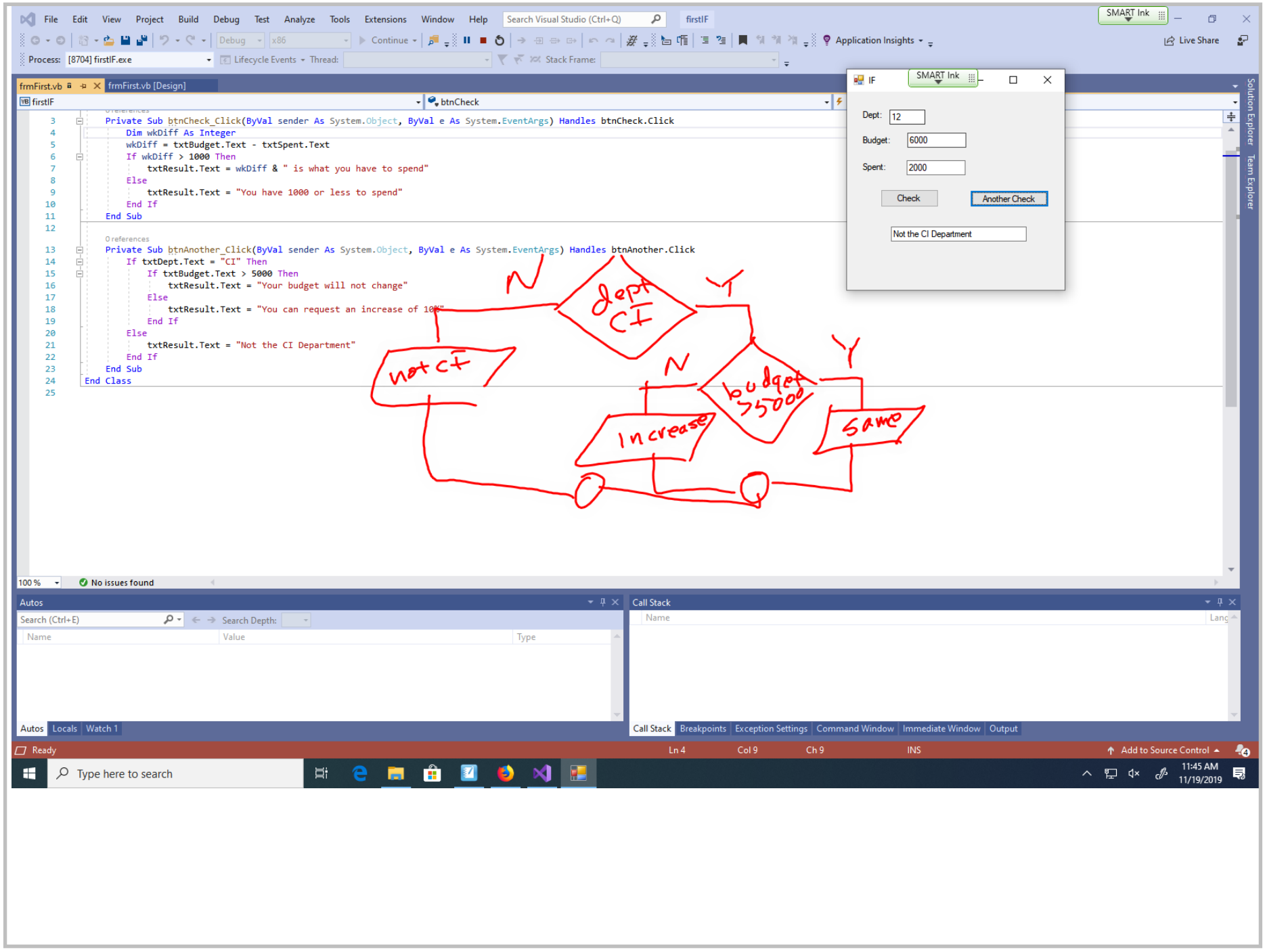

frmWhile.vb
frmWhile.vb [Design]
loopswhile120
btnProcess
Click
Public Class frmWhile
Dim wkCt As Integer = 1
Dim wkTotal As Double = 0
Private Sub btnProcess_Click(sender As System.Object, e As System.EventArgs) Handles btnProcess.Click
Do While wkCt <= CInt(txtHowMany.Text)
wkTotal = wkTotal + txtAmt.Text
wkCt = wkCt + 1
Loop
txtResult.Text = wkTotal
'Actually there is a while that can test before and after
End Sub
Private Sub btnUntil_Click(sender As System.Object, e As System.EventArgs) Handles btnUntil.Click
Do
wkTotal = wkTotal + txtAmt.Text
wkCt = wkCt + 1
Loop Until wkCt > CInt(txtHowMany.Text)
txtResult.Text = wkTotal
'Actually there is an until that can test before and after
End Sub
Private Sub btnFor_Click(sender As System.Object, e As System.EventArgs) Handles btnFor.Click
For wkCt = 1 To txtHowMany.Text Step 1
wkTotal = wkTotal + txtAmt.Text
Next
txtResult.Text = wkTotal
End Sub
End Class
While Loo
How many times:
5
Amount to Add:
12
Result:
72
Process
Other Loops:
Until
For
I did the while loop in btnProcess and then I clicked on Until to do the until loop. It went through it once and increased the total by 12 and the wkCt by 1. When it checked, wkCt was greater then txtHowMany so the loop ended after one pass.
No issues found
Autos
Call Stack
Ready
Type here to search

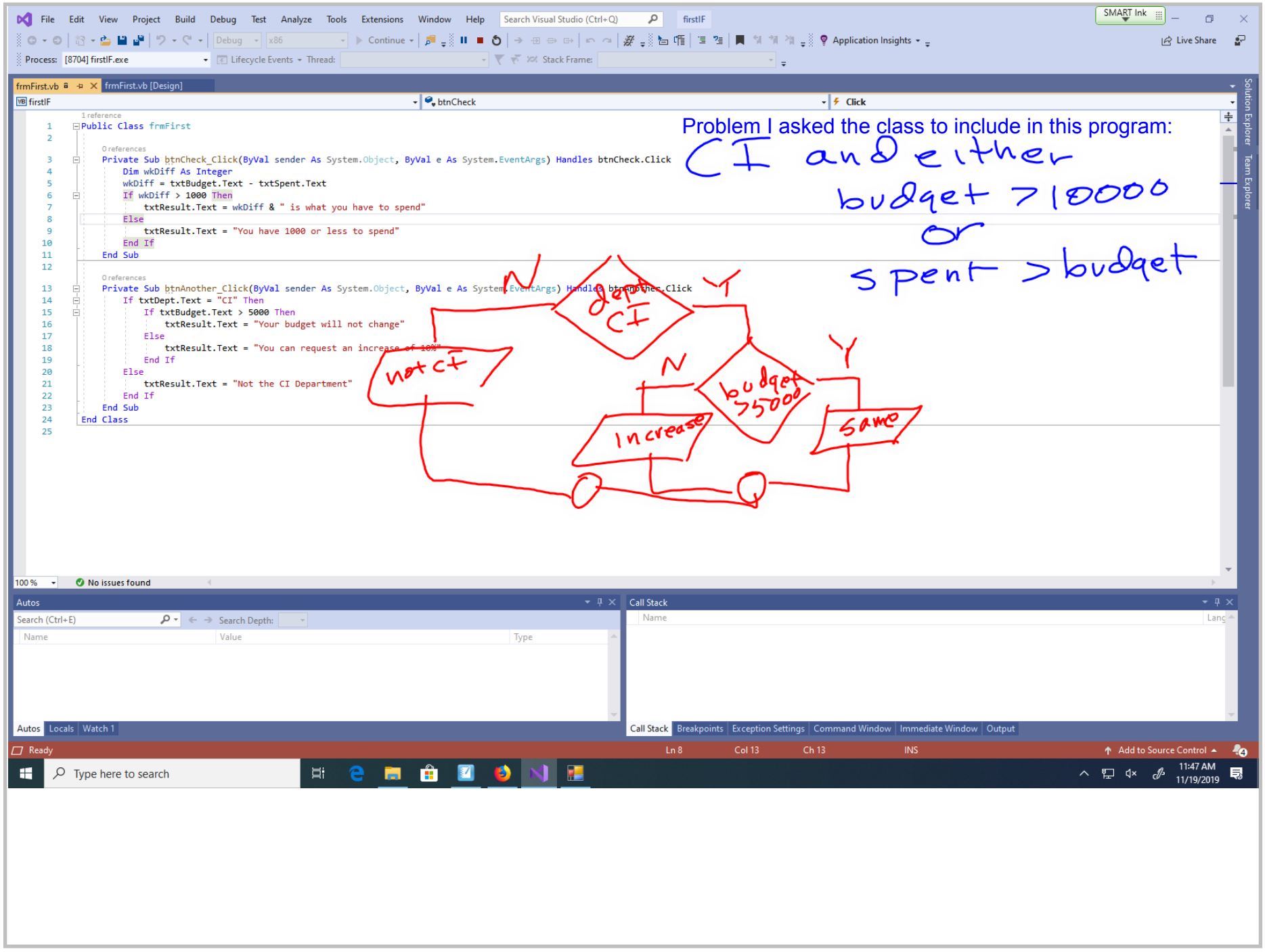

File Edit View Project Build Debug Test Analyze Tools Extensions Window Help
Search Visual Studio (Ctrl+Q)
loopswhile120
SMART Ink
Application Insights
Live Share
Process: [11732] loopswhile120.exe
Lifecycle Events
Thread:
Stack Frame:
frmWhile.vb
frmWhile.vb [Design]
loopswhile120
btnProcess
Click
1 reference
Public Class frmWhile
Dim wkCt As Integer = 1
Dim wkTotal As Double = 0
0 references
Private Sub btnProcess_Click(sender As System.Object, e As System.EventArgs) Handles btnProcess.Click
Do While wkCt <= CInt(txtHowMany.Text)
wkTotal = wkTotal + txtAmt.Text
wkCt = wkCt + 1
Loop
txtResult.Text = wkTotal
'Actually there is a while that can test before and after
End Sub
0 references
Private Sub btnUntil_Click(sender As System.Object, e As System.EventArgs) Handles btnUntil.Click
Do
wkTotal = wkTotal + txtAmt.Text
wkCt = wkCt + 1
Loop Until wkCt > CInt(txtHowMany.Text)
txtResult.Text = wkTotal
'Actually there is an until that can test before and after
End Sub
0 references
Private Sub btnFor_Click(sender As System.Object, e As System.EventArgs) Handles btnFor.Click
For wkCt = 1 To txtHowMany.Text Step 1
wkTotal = wkTotal + txtAmt.Text
Next
txtResult.Text = wkTotal
End Sub
End Class
While Loo
SMART Ink
How many times:
5
Amount to Add:
12
Result:
84
Process
Other Loops:
Until
For
I did the until loop one more time.
100 %
No issues found
Autos
Search (Ctrl+E)
Search Depth:
Name
Value
Type
Autos
Locals
Watch 1
Call Stack
Name
Call Stack
Breakpoints
Exception Settings
Command Window
Immediate Window
Output
Ready
Ln 6
Col 9
Ch 9
INS
Add to Source Control
Type here to search
11:26 AM
11/19/2019

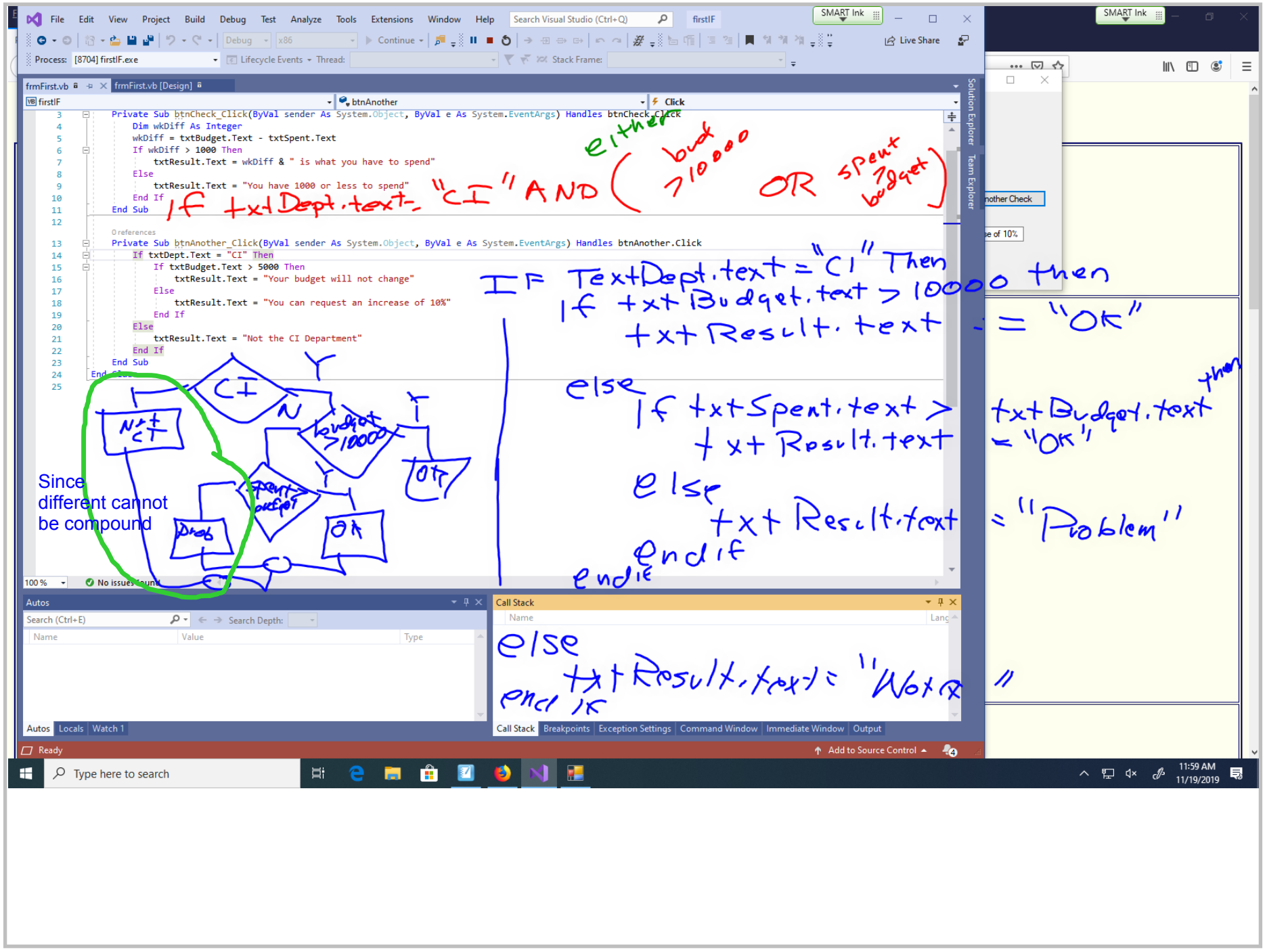

File Edit View Project Build Debug Test Analyze Tools Extensions Window Help
loopswhile120
SMART Ink
Live Share
frmWhile.vb
frmWhile.vb [Design]
loopswhile120
btnProcess
Click
1 reference
Public Class frmWhile
Dim wkCt As Integer = 1
Dim wkTotal As Double = 0
0 references
Private Sub btnProcess_Click(sender As System.Object, e As System.EventArgs) Handles btnProcess.Click
Do While wkCt <= CInt(txtHowMany.Text)
wkTotal = wkTotal + txtAmt.Text
wkCt = wkCt + 1
Loop
txtResult.Text = wkTotal
'Actually there is a while that can test before and after
End Sub
0 references
Private Sub btnUntil_Click(sender As System.Object, e As System.EventArgs) Handles btnUntil.Click
Do
wkTotal = wkTotal + txtAmt.Text
wkCt = wkCt + 1
Loop Until wkCt > CInt(txtHowMany.Text)
txtResult.Text = wkTotal
'Actually there is an until that can test before and after
End Sub
0 references
Private Sub btnFor_Click(sender As System.Object, e As System.EventArgs) Handles btnFor.Click
For wkCt = 1 To txtHowMany.Text Step 1
wkTotal = wkTotal + txtAmt.Text
Next
txtResult.Text = wkTotal
End Sub
End Class
wkCt wkTotal 60
wkCt 6
wkCT = 1
wkTotal = 0
While Loo
SMART Ink
How many times: 5
Amount to Add: 12
Result: 84
Process
Other Loops:
Until
For
The fix to this problem is to reset wkCt and wkTotal in the btnUntil.
Because of where I positioned wkCt and wkTotal, they only got reset when I restarted the program.
I should also reset in btnProcess (while loop). The FOR loop takes care of resetting
wkCt but wkTotal should be reset.
100 %
No issues found
Autos
Search (Ctrl+E)
Search Depth:
Name
Value
Type
Autos Locals Watch 1
Call Stack
Name
Call Stack Breakpoints Exception Settings Command Window Immediate Window Output
Ready
Ln 6 Col 9 Ch 9 INS
Add to Source Control
Type here to search
11:29 AM
11/19/2019

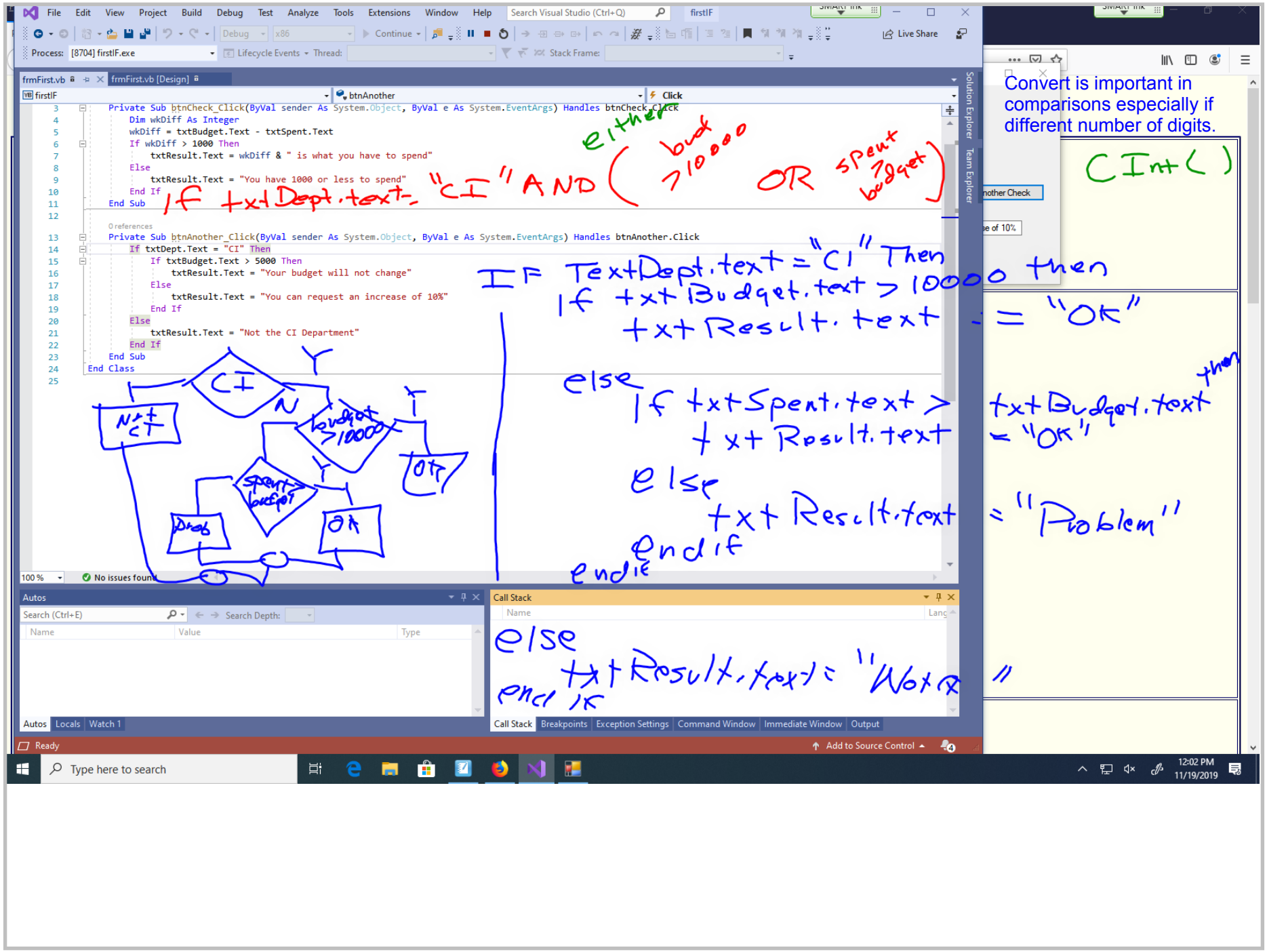

File Edit View Project Build Debug Test Analyze Tools Extensions Window Help
Search Visual Studio (Ctrl+Q)
loopswhile120
Debug
x86
Start
Live Share
frmWhile.vb
frmWhile.vb [Design]
loopswhile120
btnUntil
Click
Dim wkTotal As Double = 0
0 references
Private Sub btnProcess_Click(sender As System.Object, e As System.EventArgs) Handles btnProcess.Click
Do While wkCt <= CInt(txtHowMany.Text)
wkTotal = wkTotal + txtAmt.Text
wkCt = wkCt + 1
Loop
txtResult.Text = wkTotal
'Actually there is a while that can test before and after
End Sub
0 references
Private Sub btnUntil_Click(sender As System.Object, e As System.EventArgs) Handles btnUntil.Click
Do
wkTotal = wkTotal + txtAmt.Text
wkCt = wkCt + 1
Loop Until wkCt > CInt(txtHowMany.Text)
txtResult.Text = wkTotal
'Actually there is an until that can test before and after
End Sub
0 references
Private Sub btnFor_Click(sender As System.Object, e As System.EventArgs) Handles btnFor.Click
For wkCt = 1 To txtHowMany.Text Step 1
wkTotal = wkTotal + txtAmt.Text
Next
txtResult.Text = wkTotal
End Sub
End Class
P. grocer @ rcn.com
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'loopswhile120' (1 of 1 project)
loopswhile120
My Project
References
frmWhile.vb
Solution Explorer
Team Explorer
Properties
100 %
No issues found
Output
Show output from: Debug
Ready
Ln 15
Col 8
Ch 8
INS
Add to Source Control
Type here to search
11:38 AM
11/19/2019


File Edit View Project Build Debug Test Analyze Tools Extensions Window Help
Search Visual Studio (Ctrl+Q)
firstIF
SMART Ink
Live Share
Debug
x86
Start
frmFirst.vb
frmFirst.vb [Design]
firstIF
btnCheck
Click
Private Sub btnCheck_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnCheck.Click
Dim wkDiff As Integer
wkDiff = txtBudget.Text - txtSpent.Text
If wkDiff > 1000 Then
txtResult.Text = wkDiff & " is what you have to spend"
Else
txtResult.Text = "You have 1000 or less to spend"
End If
End Sub
0 references
Private Sub btnAnother_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnAnother.Click
If txtDept.Text = "CI" Then
If txtBudget.Text > 5000 Then
txtResult.Text = "Your budget will not change"
Else
txtResult.Text = "You can request an increase of 10%"
End If
Else
txtResult.Text = "Not the CI Department"
End If
End Sub
End Class
N
dept CI
Y
not CI
N
budget 5000
Y
increase
same
100 %
No issues found
Output
Show output from: Debug
'firstIF.exe' (CLR v4.0.30319: firstIF.exe): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System.Configuration\v4.0_4.0.0.0__b03f5f7f11d50a3a\System.Configuration.dll'. S
'firstIF.exe' (CLR v4.0.30319: firstIF.exe): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System.Core\v4.0_4.0.0.0__b77a5c561934e089\System.Core.dll'. Skipped loading sym
'firstIF.exe' (CLR v4.0.30319: firstIF.exe): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System.Xml\v4.0_4.0.0.0__b77a5c561934e089\System.Xml.dll'. Skipped loading symbo
'firstIF.exe' (CLR v4.0.30319: firstIF.exe): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System.Runtime.Remoting\v4.0_4.0.0.0__b77a5c561934e089\System.Runtime.Remoting.d
'firstIF.exe' (CLR v4.0.30319: firstIF.exe): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\Accessibility\v4.0_4.0.0.0__b03f5f7f11d50a3a\Accessibility.dll'.
The program '[9352] firstIF.exe: Program Trace' has exited with code 0 (0x0).
The program '[9352] firstIF.exe' has exited with code 0 (0x0).
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'firstIF' (1 of 1 project)
firstIF
My Project
References
frmFirst.vb
Solution Explorer
Team Explorer
Properties
Server Explorer
Toolbox
Data Sources
Diagnostic Tools
Ready
Ln 4
Col 9
Ch 9
INS
Add to Source Control
Type here to search
11:43 AM
11/19/2019


IF
SMART Ink
Dept: 12
Budget: 6000
Spent: 2000
Check
Another Check
Not the CI Department
dept CI
N
Y
not ci
budget >5000
N
Y
increase
same


Problem I asked the class to include in this program:
CI and either
budget >10000
or
spent > budget
frmFirst.vb
frmFirst.vb [Design]
firstIF
btnCheck
Click
1 reference
Public Class frmFirst
0 references
Private Sub btnCheck_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnCheck.Click
Dim wkDiff As Integer
wkDiff = txtBudget.Text - txtSpent.Text
If wkDiff > 1000 Then
txtResult.Text = wkDiff & " is what you have to spend"
Else
txtResult.Text = "You have 1000 or less to spend"
End If
End Sub
0 references
Private Sub btnAnother_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnAnother.Click
If txtDept.Text = "CI" Then
If txtBudget.Text > 5000 Then
txtResult.Text = "Your budget will not change"
Else
txtResult.Text = "You can request an increase of 10%"
End If
Else
txtResult.Text = "Not the CI Department"
End If
End Sub
End Class
dept CI
N
Y
not CI
budget 5000
N
Y
increase
same
100 %
No issues found
Autos
Search (Ctrl+E)
Search Depth:
Name
Value
Type
Call Stack
Name
Autos
Locals
Watch 1
Call Stack
Breakpoints
Exception Settings
Command Window
Immediate Window
Output
Ready
Ln 8
Col 13
Ch 13
INS
Add to Source Control
Type here to search
11:47 AM
11/19/2019


File Edit View Project Build Debug Test Analyze Tools Extensions Window Help
Search Visual Studio (Ctrl+Q)
firstIF
SMART Ink
Live Share
frmFirst.vb
frmFirst.vb [Design]
firstIF
btnAnother
Click
Private Sub btnCheck_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnCheck.Click
Dim wkDiff As Integer
wkDiff = txtBudget.Text - txtSpent.Text
If wkDiff > 1000 Then
txtResult.Text = wkDiff & " is what you have to spend"
Else
txtResult.Text = "You have 1000 or less to spend"
End If
End Sub
0 references
Private Sub btnAnother_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnAnother.Click
If txtDept.Text = "CI" Then
If txtBudget.Text > 5000 Then
txtResult.Text = "Your budget will not change"
Else
txtResult.Text = "You can request an increase of 10%"
End If
Else
txtResult.Text = "Not the CI Department"
End If
End Sub
either
budget >10000
OR spent budget
if txtDept.text = "CI" AND (
IF TextDept.text = "CI" Then
If txtBudget.text > 10000 then
txtResult.text = "OK"
else
If txtSpent.text > txtResult.text
else
txtResult.text = "Problem"
endif
endif
Not CI
CI
N
budget >10000
Y
OK
spent budget
Y
Drop
OK
Since different cannot be compound
100 %
No issues found
Autos
Search (Ctrl+E)
Search Depth:
Name
Value
Type
Autos
Locals
Watch 1
Call Stack
Name
Call Stack
Breakpoints
Exception Settings
Command Window
Immediate Window
Output
else
txtResult.text = "Work"
endif
Ready
Add to Source Control
Type here to search
11:59 AM
11/19/2019


Convert is important in comparisons especially if different number of digits.
CInt( )
Private Sub btnCheck_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnCheck.Click
Dim wkDiff As Integer
wkDiff = txtBudget.Text - txtSpent.Text
If wkDiff > 1000 Then
txtResult.Text = wkDiff & " is what you have to spend"
Else
txtResult.Text = "You have 1000 or less to spend"
End If
End Sub
Private Sub btnAnother_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnAnother.Click
If txtDept.Text = "CI" Then
If txtBudget.Text > 5000 Then
txtResult.Text = "Your budget will not change"
Else
txtResult.Text = "You can request an increase of 10%"
End If
Else
txtResult.Text = "Not the CI Department"
End If
End Sub
End Class
CI
Not CI
Budget >10000
OK
Spent > budget
Prob
OK
IF TextDept.text = "CI" Then
If txtBudget.text > 10000 then
txtResult.text = "OK"
else
If txtSpent.text > txtBudget.text then
txtResult.text = "OK"
else
txtResult.text = "Problem"
endif
endif
else
txtResult.text = "Work
end if
Autos
Call Stack
Ready
Type here to search
12:02 PM
11/19/2019