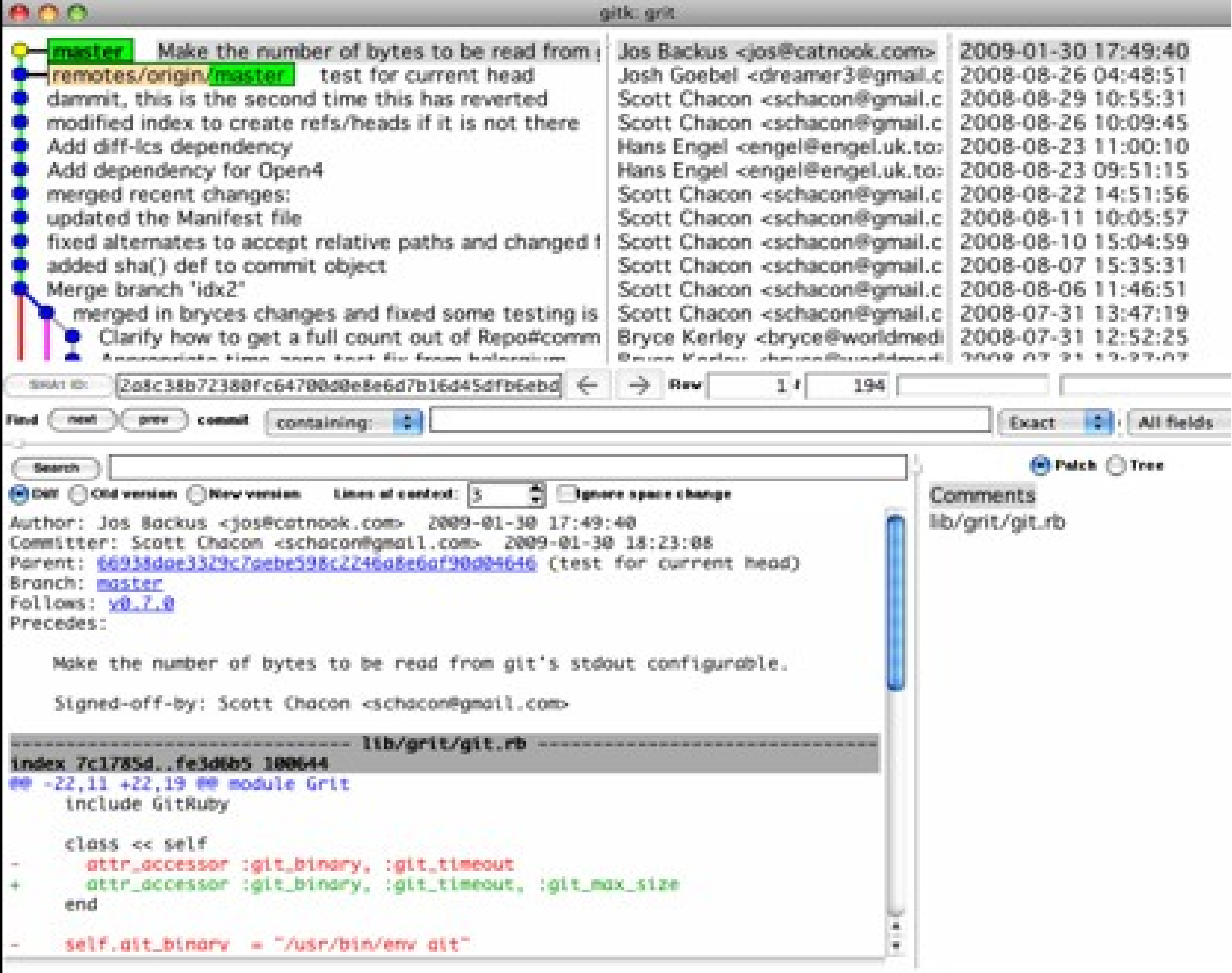
How to get productive on your first day.

Greg Young

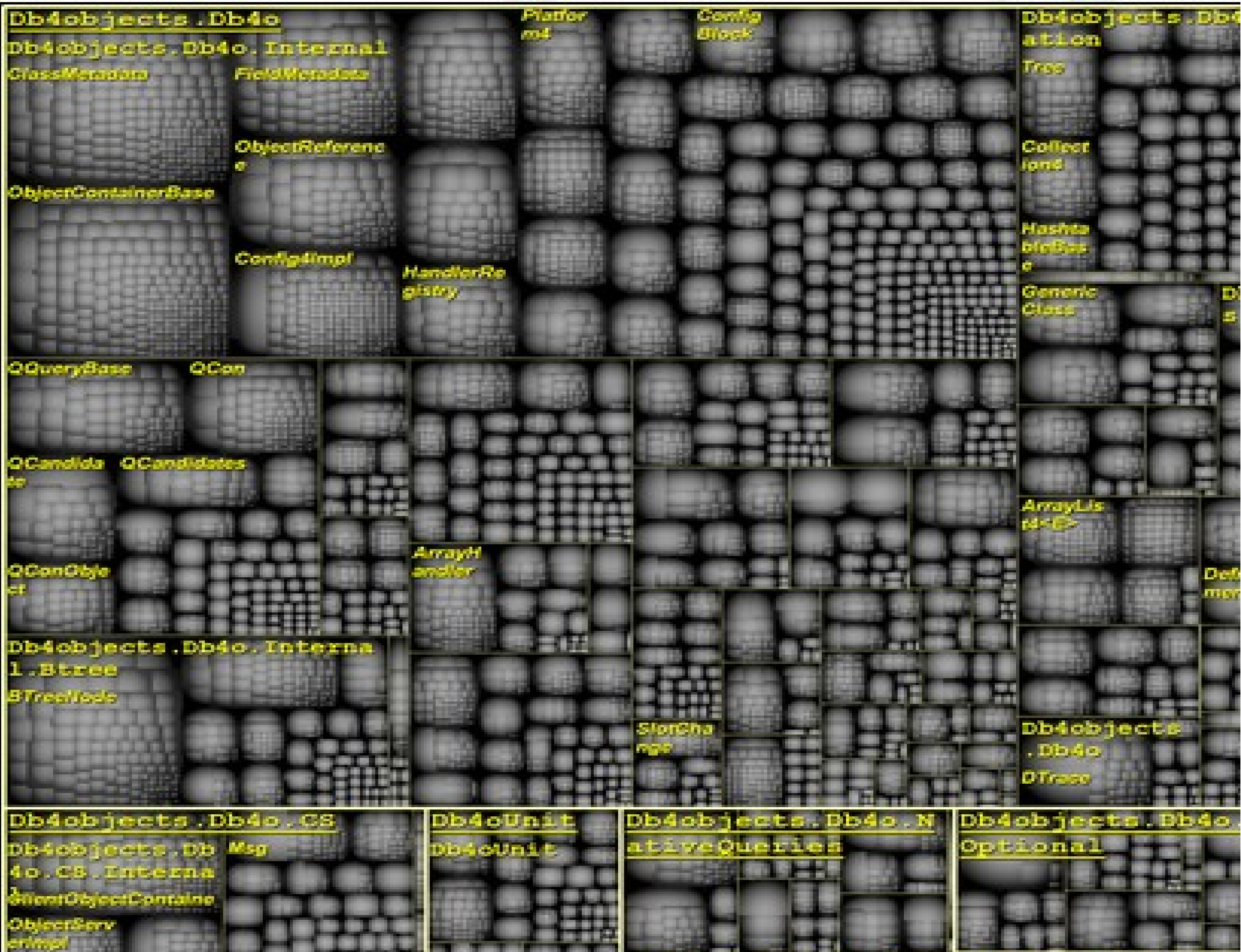
Data about our code is as valuable as the data we produce for our clients.

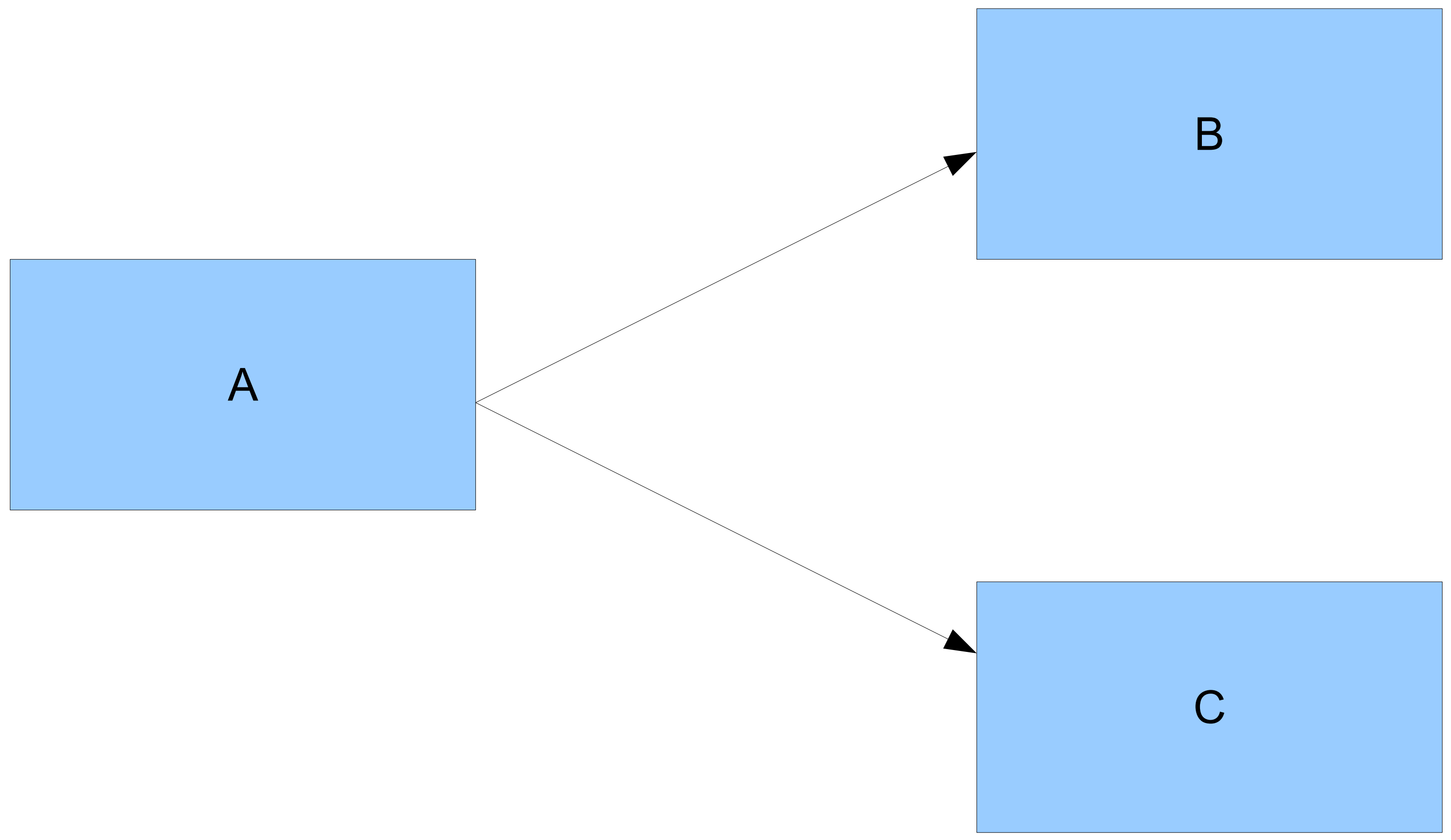


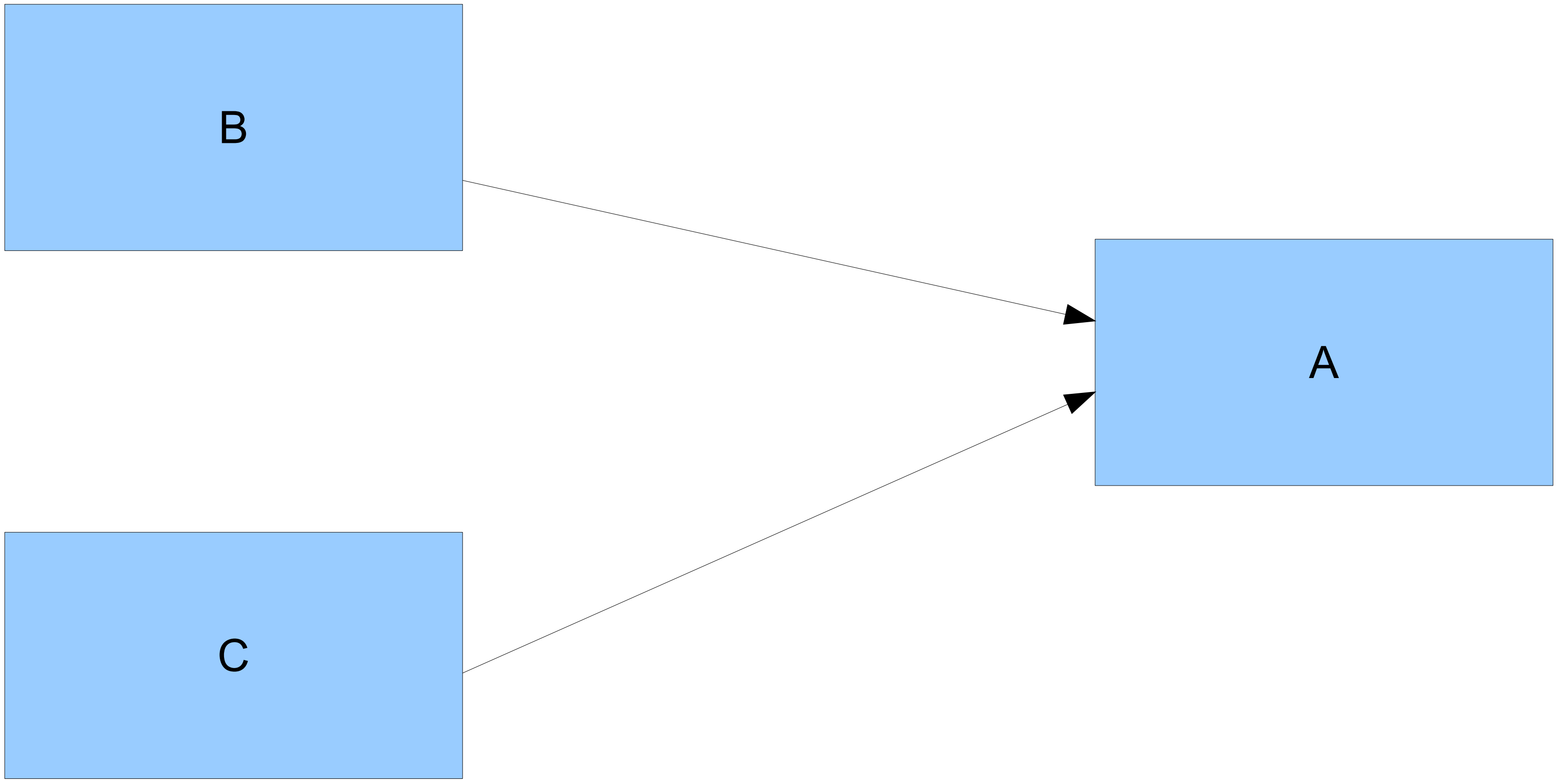
```
master  Make the number of bytes to be read from          Jos Backus <jos@catnook.com>            2009-01-30 17:49:40
remotes/origin/master  test for current head              Josh Goebel <dreamer3@gmail.c           2008-08-26 04:48:51
dammit, this is the second time this has reverted         Scott Chacon <schacon@gmail.c           2008-08-29 10:55:31
modified index to create refs/heads if it is not there    Scott Chacon <schacon@gmail.c           2008-08-26 10:09:45
Add diff-lcs dependency                                   Hans Engel <engel@engel.uk.to:          2008-08-23 11:00:10
Add dependency for Open4                                  Hans Engel <engel@engel.uk.to:          2008-08-23 09:51:15
merged recent changes:                                    Scott Chacon <schacon@gmail.c           2008-08-22 14:51:56
updated the Manifest file                                 Scott Chacon <schacon@gmail.c           2008-08-11 10:05:57
fixed alternates to accept relative paths and changed     Scott Chacon <schacon@gmail.c           2008-08-10 15:04:59
added sha() def to commit object                          Scott Chacon <schacon@gmail.c           2008-08-07 15:35:31
Merge branch 'idx2'                                       Scott Chacon <schacon@gmail.c           2008-08-06 11:46:51
  merged in bryces changes and fixed some testing is      Scott Chacon <schacon@gmail.c           2008-07-31 13:47:19
    Clarify how to get a full count out of Repo#comm      Bryce Kerley <bryce@worldmedi           2008-07-31 12:52:25
```

```
SHA1 ID: 2a8c38b72380fc64700d0e8e6d7b16d45dfb6ebd    Row 1 / 194
```

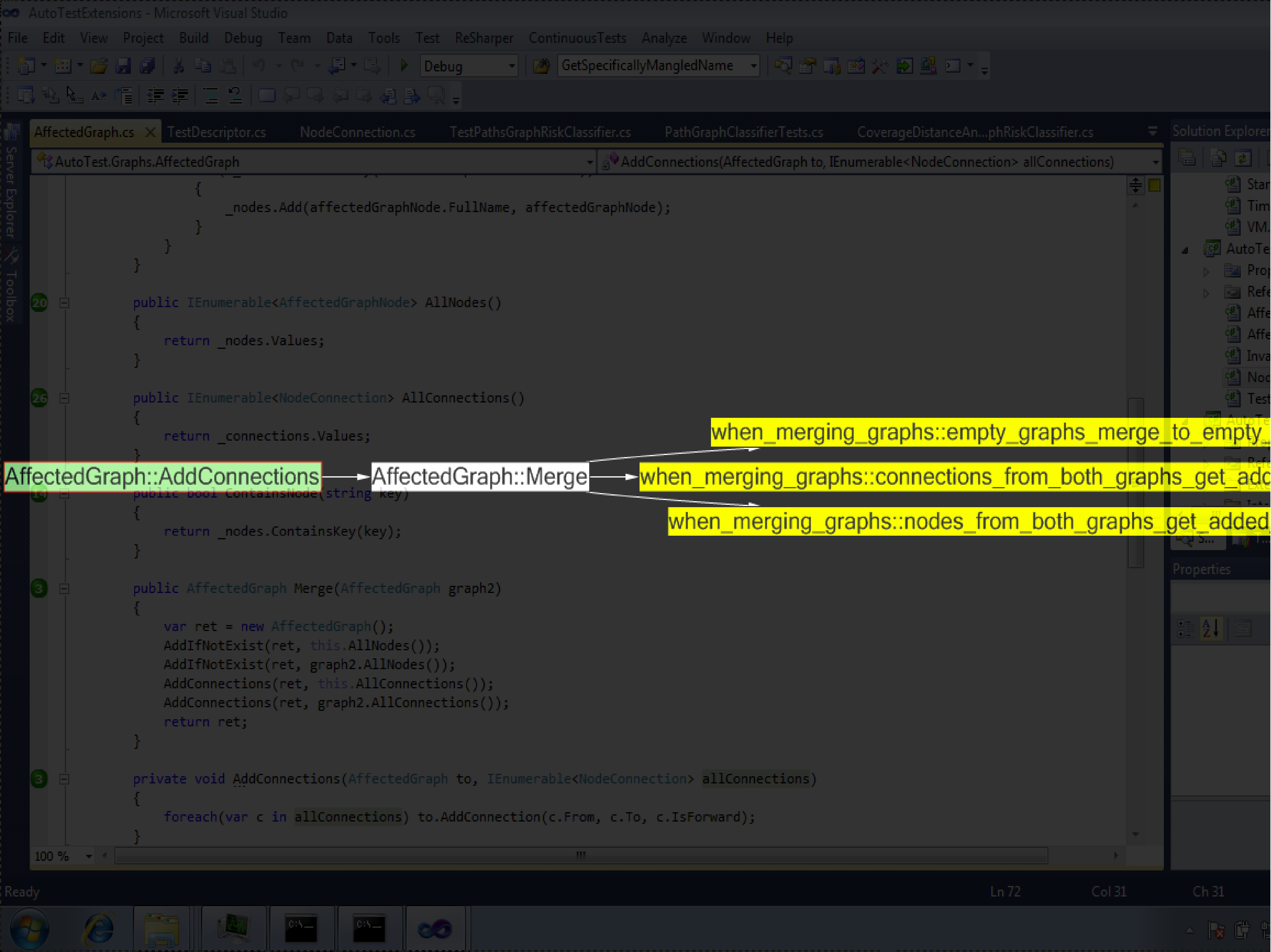
```
Author: Jos Backus <jos@catnook.com>  2009-01-30 17:49:40
Committer: Scott Chacon <schacon@gmail.com>  2009-01-30 18:23:08
Parent: 66938dae3329c7aebe598c2246a8e6af90d04646 (test for current head)
Branch: master
Follows: v0.7.0
Precedes:

    Make the number of bytes to be read from git's stdout configurable.

    Signed-off-by: Scott Chacon <schacon@gmail.com>

------------------------------- lib/grit/git.rb -------------------------------
index 7c1785d..fe3d6b5 100644
@@ -22,11 +22,19 @@ module Grit
     include GitRuby

     class << self
-      attr_accessor :git_binary, :git_timeout
+      attr_accessor :git_binary, :git_timeout, :git_max_size
     end

-    self.git_binary  = "/usr/bin/env git"
```


Db4objects.Db4o
Db4objects.Db4o.Internal
ClassMetadata
FieldMetadata
Platform4
ConfigBlock
Db4objects.Db4
ation
Tree
Collections
HashtableBase
ObjectReference
ObjectContainerBase
Config4Impl
HandlerRegistry
GenericClass
QQueryBase
QCon
QCandidate
QCandidates
QConObject
ArrayHandler
ArrayList4<E>
Db4objects.Db4o.Internal.Btree
BTreeNode
SlotChange
Db4objects.Db4o
DTrace
Db4objects.Db4o.CS
Db4objects.Db4o.CS.Internal
ClientObjectContainer
ObjectServerImpl
Msg
Db4oUnit
Db4oUnit
Db4objects.Db4o.NativeQueries
Db4objects.Db4o.Optional

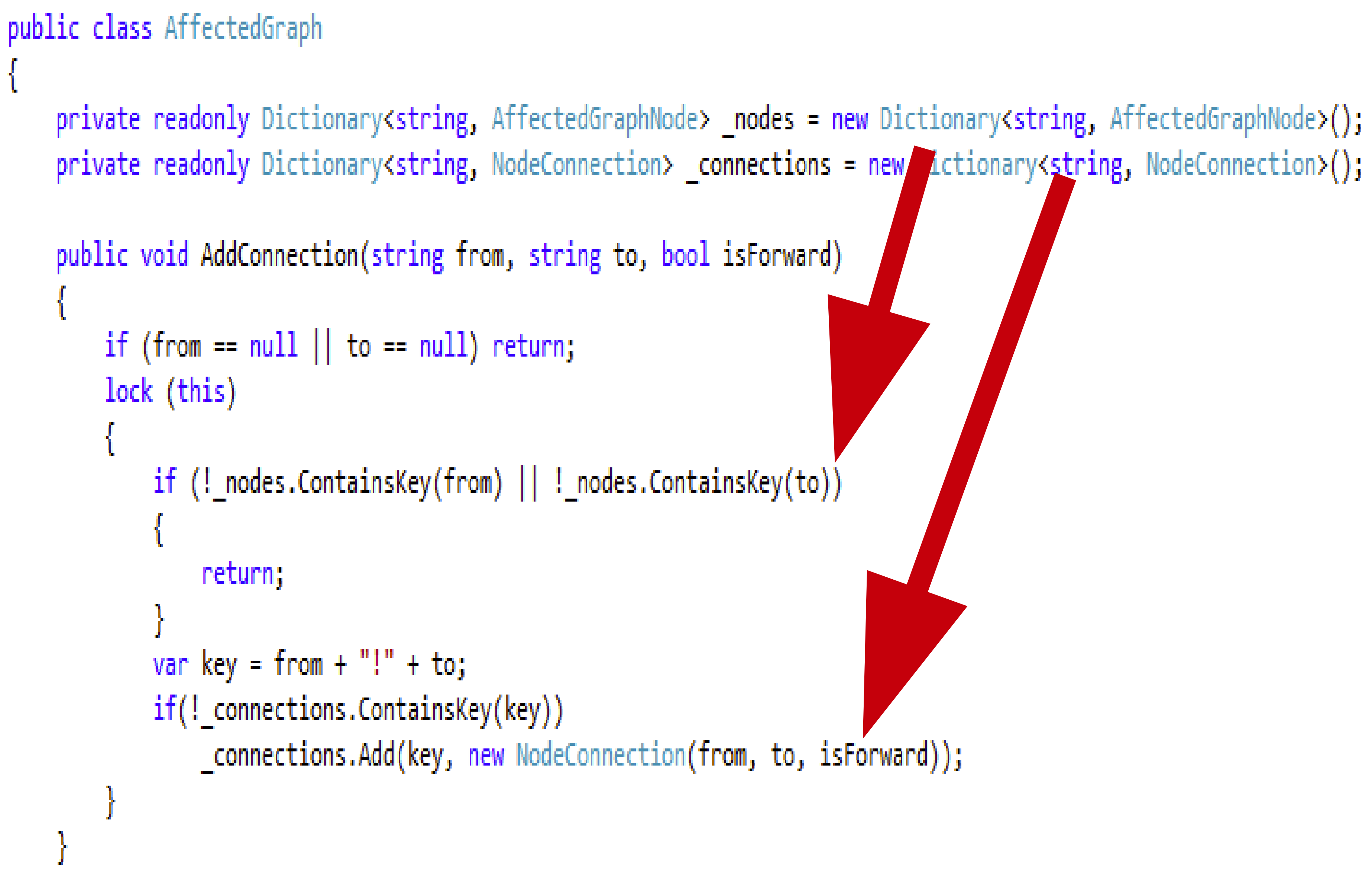

A
B
C


B
A
C

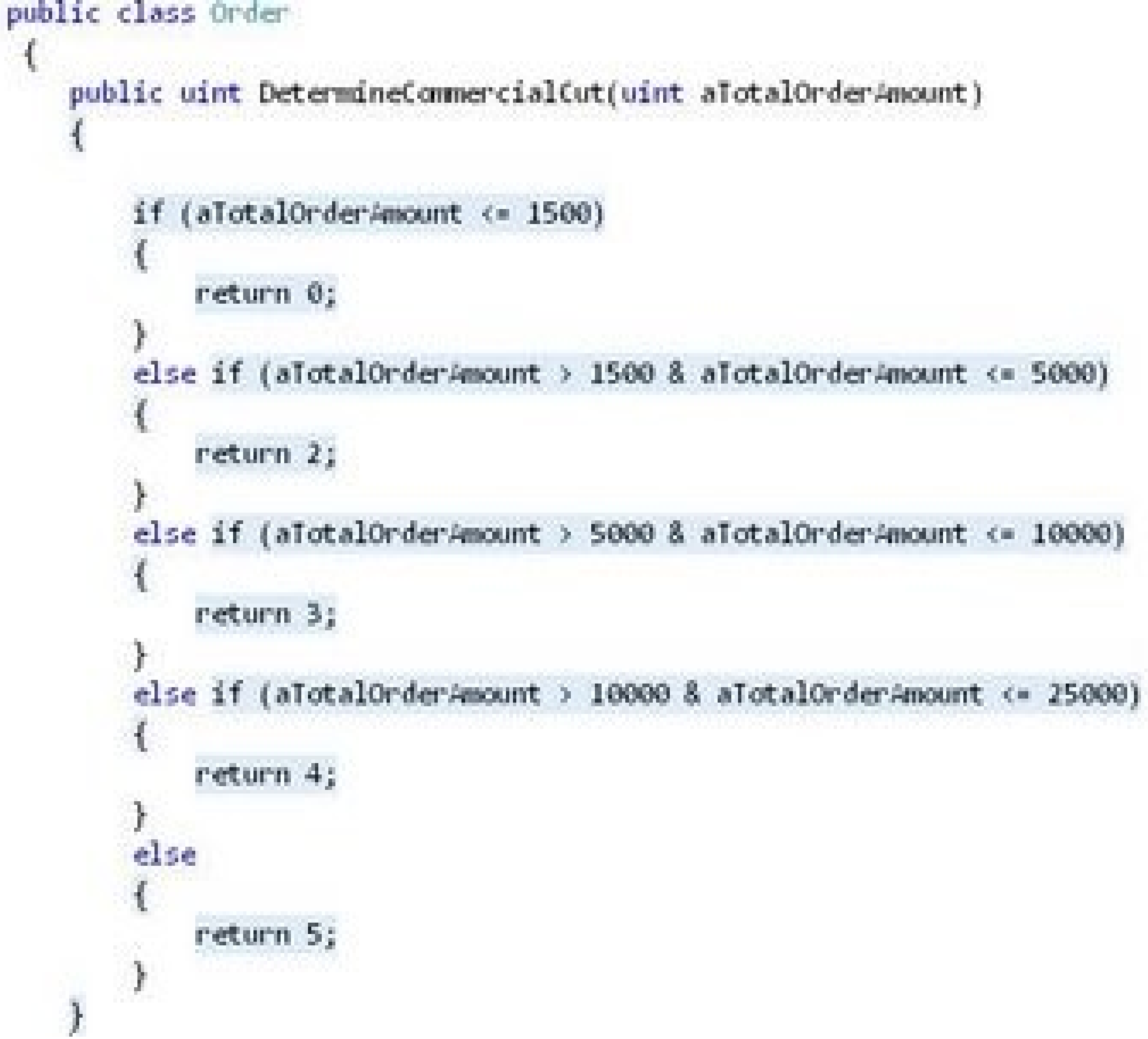
$$\begin{tikzcd}
 & \boxed{IB} \\
\boxed{A} \arrow[ur] \arrow[dr] & \\
 & \boxed{IC}
\end{tikzcd}$$


AutoTestExtensions - Microsoft Visual Studio
File Edit View Project Build Debug Team Data Tools Test ReSharper ContinuousTests Analyze Window Help
Debug
GetSpecificallyMangledName
AffectedGraph.cs
TestDescriptor.cs
NodeConnection.cs
TestPathsGraphRiskClassifier.cs
PathGraphClassifierTests.cs
CoverageDistanceAn...phRiskClassifier.cs
Solution Explorer
AutoTest.Graphs.AffectedGraph
AddConnections(AffectedGraph to, IEnumerable<NodeConnection> allConnections)
AffectedGraph::AddConnections
AffectedGraph::Merge
when_merging_graphs::empty_graphs_merge_to_empty
when_merging_graphs::connections_from_both_graphs_get_adc
when_merging_graphs::nodes_from_both_graphs_get_added
Properties
100 %
Ready
Ln 72
Col 31
Ch 31

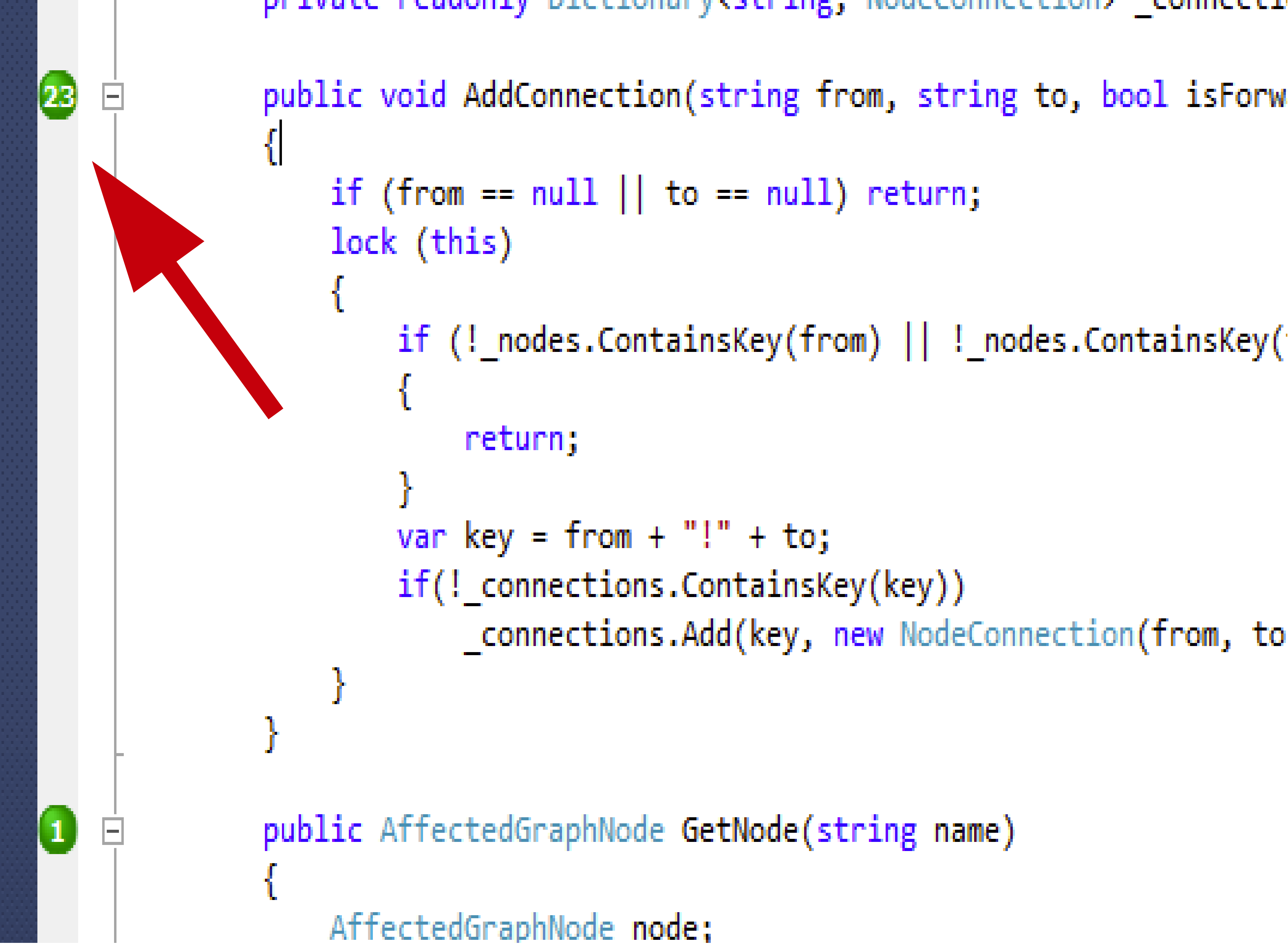
```
            {
                _nodes.Add(affectedGraphNode.FullName, affectedGraphNode);
            }
        }
    }

    public IEnumerable<AffectedGraphNode> AllNodes()
    {
        return _nodes.Values;
    }

    public IEnumerable<NodeConnection> AllConnections()
    {
        return _connections.Values;
    }

    public bool ContainsNode(string key)
    {
        return _nodes.ContainsKey(key);
    }

    public AffectedGraph Merge(AffectedGraph graph2)
    {
        var ret = new AffectedGraph();
        AddIfNotExist(ret, this.AllNodes());
        AddIfNotExist(ret, graph2.AllNodes());
        AddConnections(ret, this.AllConnections());
        AddConnections(ret, graph2.AllConnections());
        return ret;
    }

    private void AddConnections(AffectedGraph to, IEnumerable<NodeConnection> allConnections)
    {
        foreach(var c in allConnections) to.AddConnection(c.From, c.To, c.IsForward);
    }
```



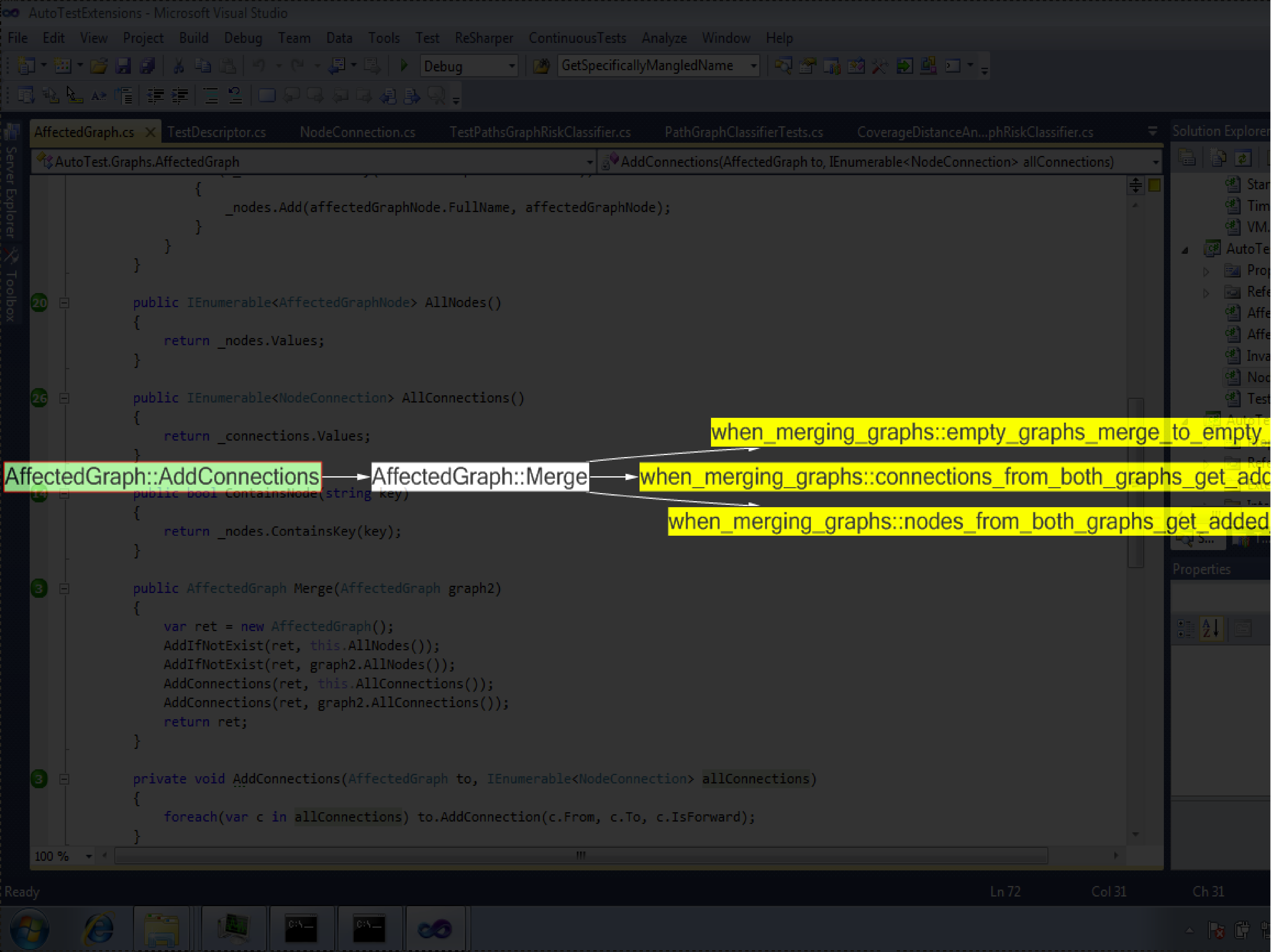
```
public class AffectedGraph
{
    private readonly Dictionary<string, AffectedGraphNode> _nodes = new Dictionary<string, AffectedGraphNode>();
    private readonly Dictionary<string, NodeConnection> _connections = new Dictionary<string, NodeConnection>();

    public void AddConnection(string from, string to, bool isForward)
    {
        if (from == null || to == null) return;
        lock (this)
        {
            if (!_nodes.ContainsKey(from) || !_nodes.ContainsKey(to))
            {
                return;
            }
            var key = from + "!" + to;
            if(!_connections.ContainsKey(key))
                _connections.Add(key, new NodeConnection(from, to, isForward));
        }
    }
```



```
public class Order
 {
    public uint DetermineCommercialCut(uint aTotalOrderAmount)
    {

        if (aTotalOrderAmount <= 1500)
        {
            return 0;
        }
        else if (aTotalOrderAmount > 1500 & aTotalOrderAmount <= 5000)
        {
            return 2;
        }
        else if (aTotalOrderAmount > 5000 & aTotalOrderAmount <= 10000)
        {
            return 3;
        }
        else if (aTotalOrderAmount > 10000 & aTotalOrderAmount <= 25000)
        {
            return 4;
        }
        else
        {
            return 5;
        }
    }
```



```
private readonly Dictionary<string, NodeConnection> _connecti

public void AddConnection(string from, string to, bool isForw
{
    if (from == null || to == null) return;
    lock (this)
    {
        if (!_nodes.ContainsKey(from) || !_nodes.ContainsKey(
        {
            return;
        }
        var key = from + "!" + to;
        if(!_connections.ContainsKey(key))
            _connections.Add(key, new NodeConnection(from, to
    }
}

public AffectedGraphNode GetNode(string name)
{
    AffectedGraphNode node;
```


AutoTestExtensions - Microsoft Visual Studio
File Edit View Project Build Debug Team Data Tools Test ReSharper ContinuousTests Analyze Window Help
Debug
GetSpecificallyMangledName
AffectedGraph.cs
TestDescriptor.cs
NodeConnection.cs
TestPathsGraphRiskClassifier.cs
PathGraphClassifierTests.cs
CoverageDistanceAn...phRiskClassifier.cs
Solution Explorer
AutoTest.Graphs.AffectedGraph
AddConnections(AffectedGraph to, IEnumerable<NodeConnection> allConnections)
Properties
100 %
Ready
Ln 72
Col 31
Ch 31

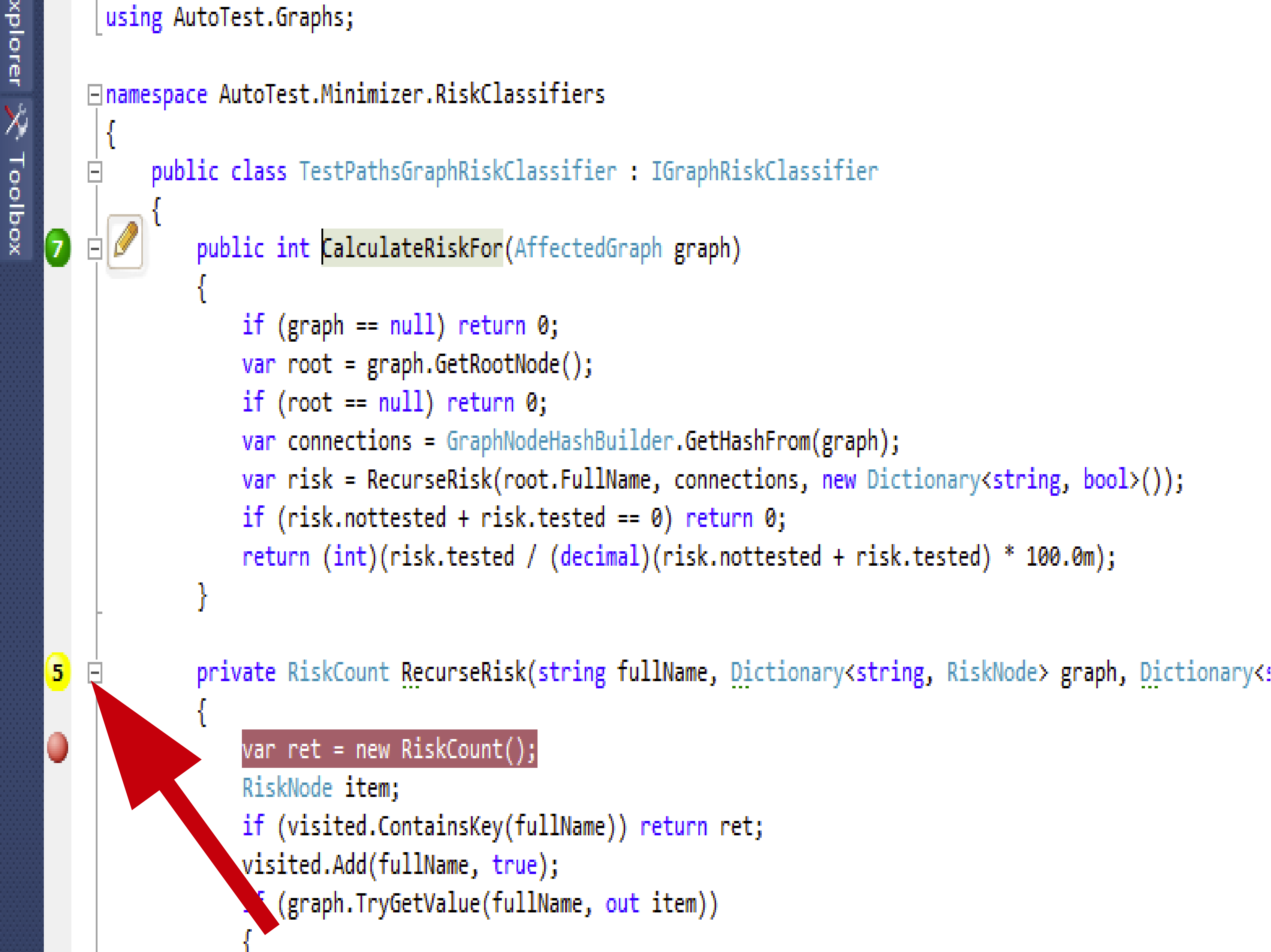
```
            {
                _nodes.Add(affectedGraphNode.FullName, affectedGraphNode);
            }
        }
    }

    public IEnumerable<AffectedGraphNode> AllNodes()
    {
        return _nodes.Values;
    }

    public IEnumerable<NodeConnection> AllConnections()
    {
        return _connections.Values;
    }

    public bool ContainsNode(string key)
    {
        return _nodes.ContainsKey(key);
    }

    public AffectedGraph Merge(AffectedGraph graph2)
    {
        var ret = new AffectedGraph();
        AddIfNotExist(ret, this.AllNodes());
        AddIfNotExist(ret, graph2.AllNodes());
        AddConnections(ret, this.AllConnections());
        AddConnections(ret, graph2.AllConnections());
        return ret;
    }

    private void AddConnections(AffectedGraph to, IEnumerable<NodeConnection> allConnections)
    {
        foreach(var c in allConnections) to.AddConnection(c.From, c.To, c.IsForward);
    }
```

AffectedGraph::AddConnections → AffectedGraph::Merge

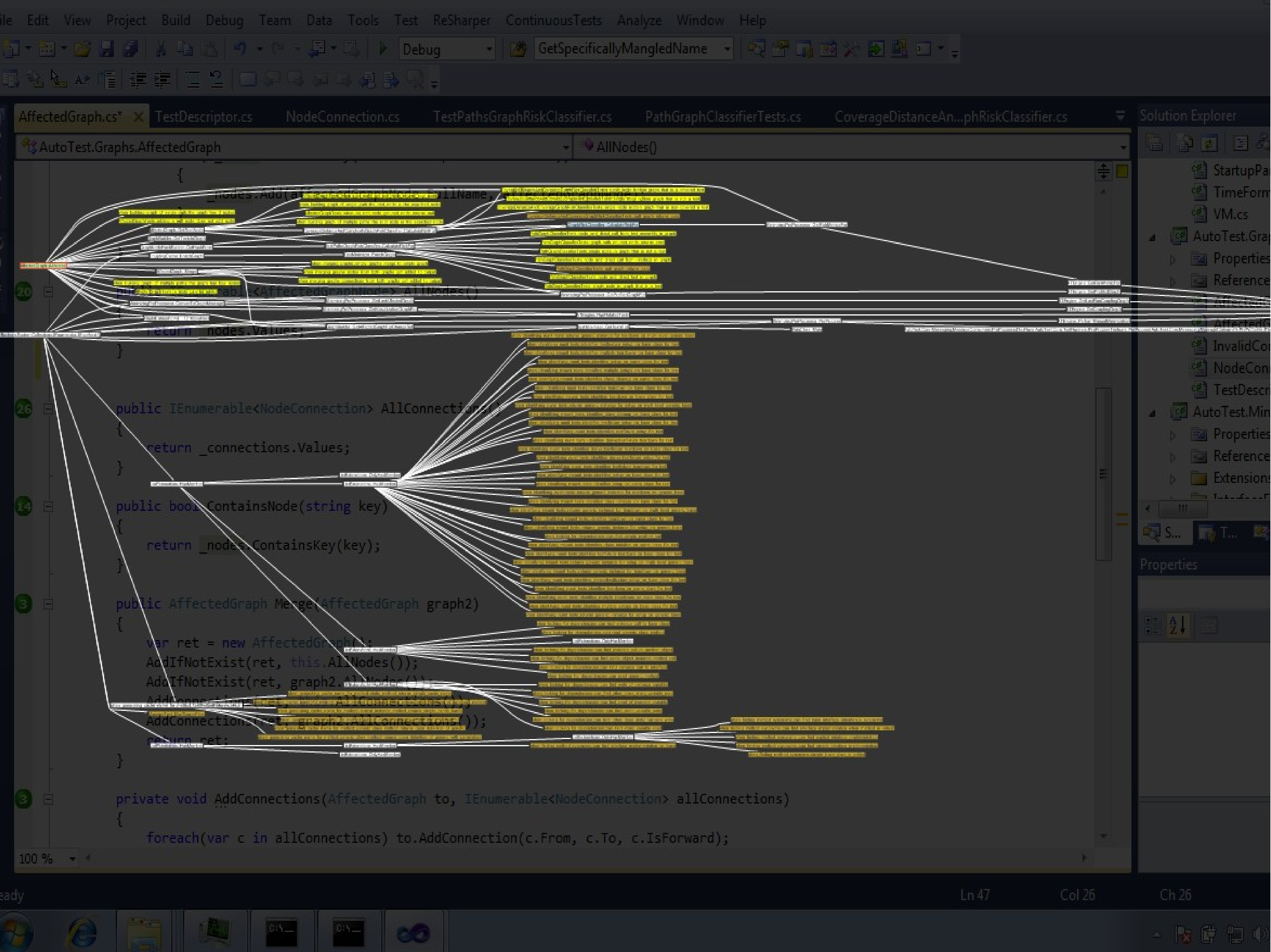
when_merging_graphs::empty_graphs_merge_to_empty_

when_merging_graphs::connections_from_both_graphs_get_adc

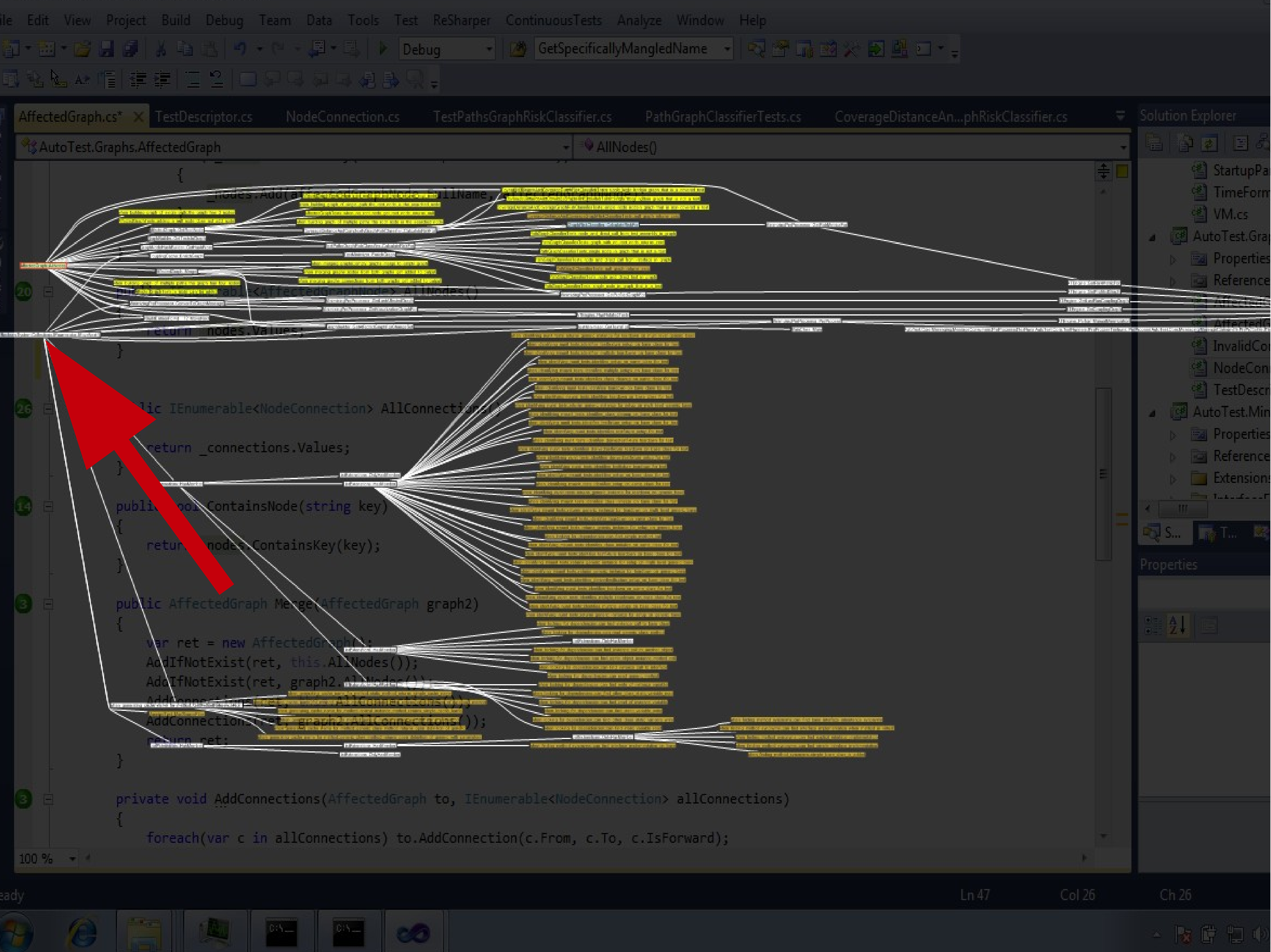
when_merging_graphs::nodes_from_both_graphs_get_added



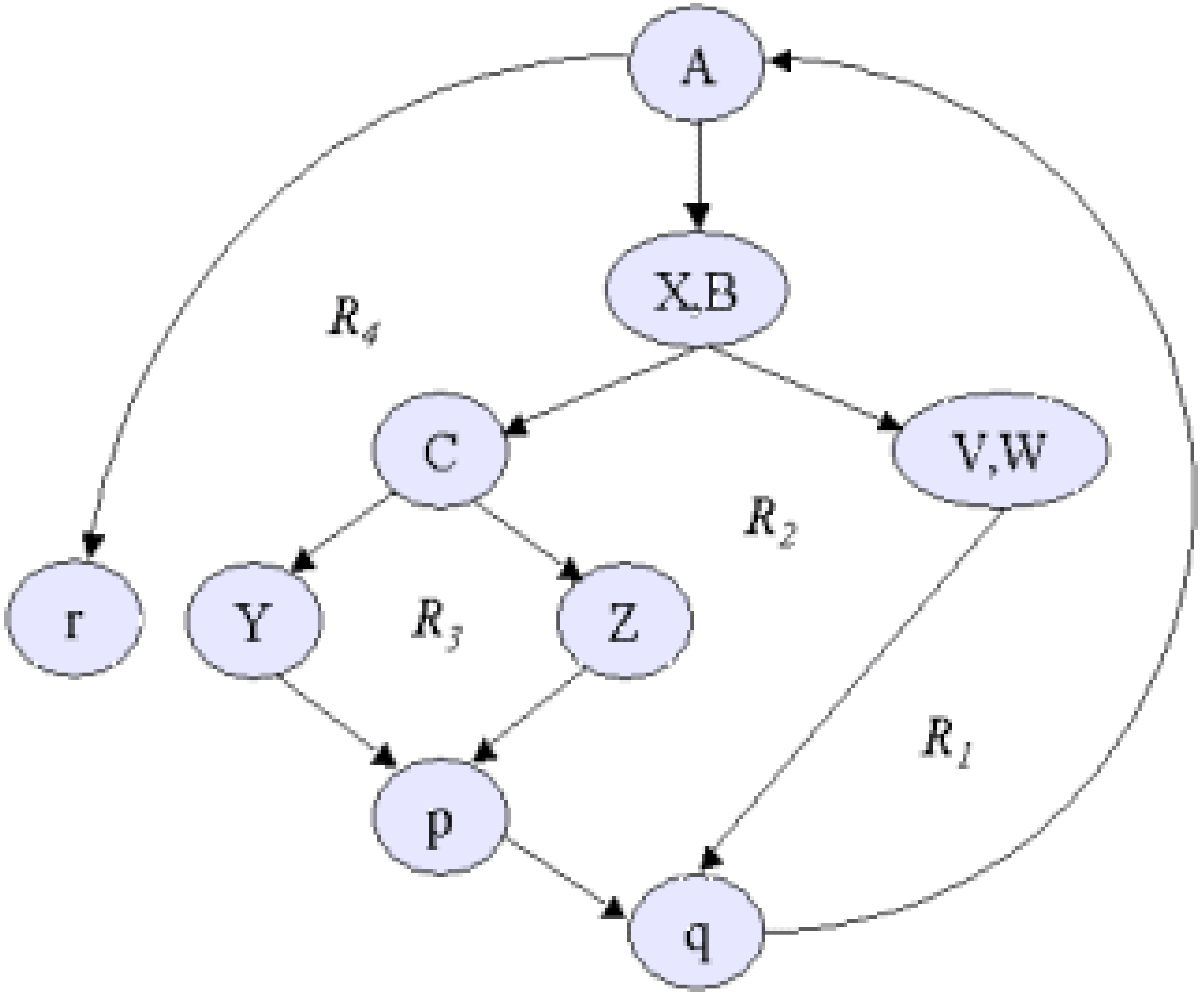
```
using AutoTest.Graphs;

namespace AutoTest.Minimizer.RiskClassifiers
{
    public class TestPathsGraphRiskClassifier : IGraphRiskClassifier
    {
        public int CalculateRiskFor(AffectedGraph graph)
        {
            if (graph == null) return 0;
            var root = graph.GetRootNode();
            if (root == null) return 0;
            var connections = GraphNodeHashBuilder.GetHashFrom(graph);
            var risk = RecurseRisk(root.FullName, connections, new Dictionary<string, bool>());
            if (risk.nottested + risk.tested == 0) return 0;
            return (int)(risk.tested / (decimal)(risk.nottested + risk.tested) * 100.0m);
        }

        private RiskCount RecurseRisk(string fullName, Dictionary<string, RiskNode> graph, Dictionary<s
        {
            var ret = new RiskCount();
            RiskNode item;
            if (visited.ContainsKey(fullName)) return ret;
            visited.Add(fullName, true);
            if (graph.TryGetValue(fullName, out item))
            {
```

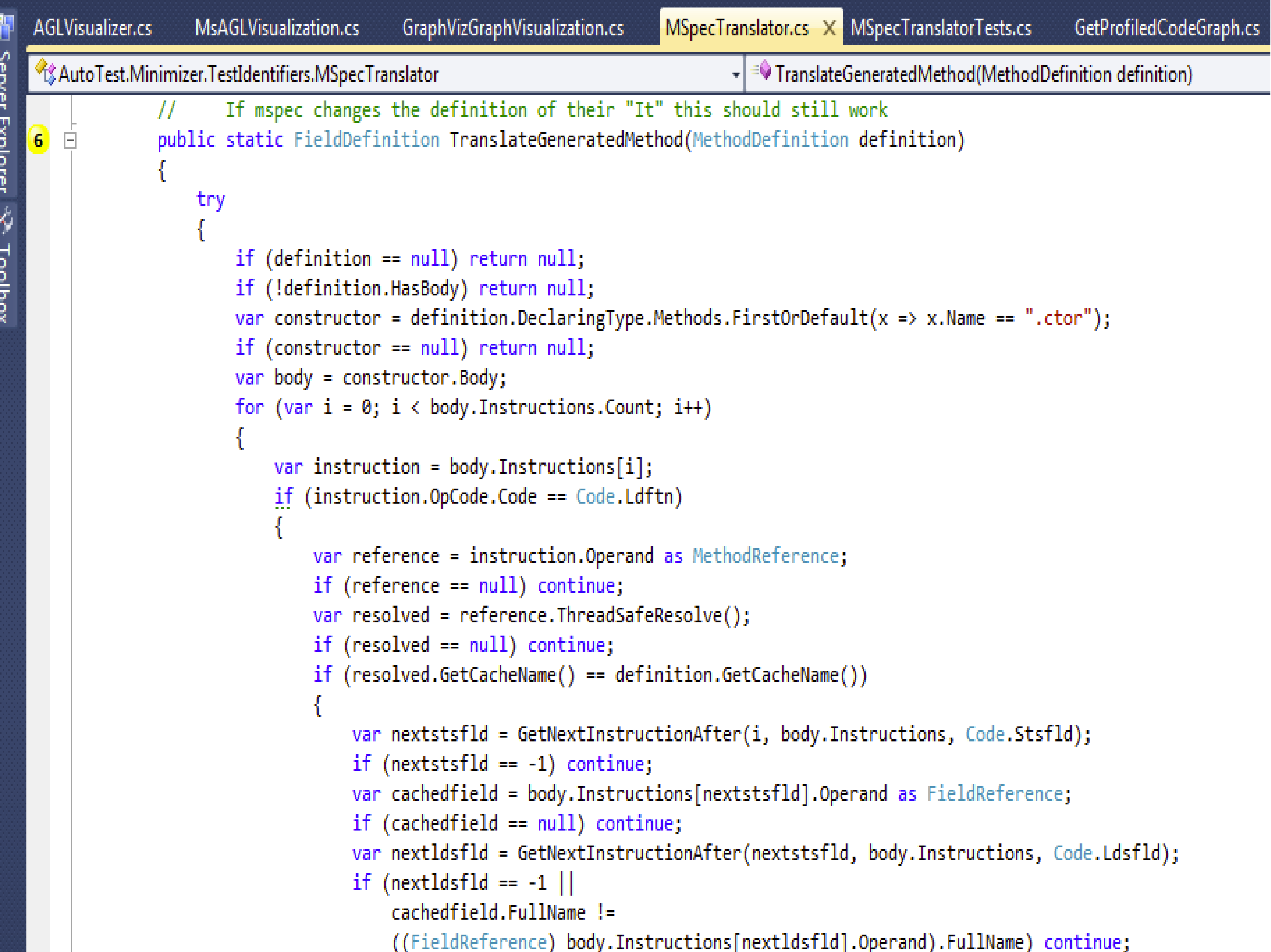

File Edit View Project Build Debug Team Data Tools Test ReSharper ContinuousTests Analyze Window Help
Debug
GetSpecificallyMangledName
AffectedGraph.cs*
TestDescriptor.cs
NodeConnection.cs
TestPathsGraphRiskClassifier.cs
PathGraphClassifierTests.cs
CoverageDistanceAn...phRiskClassifier.cs
AutoTest.Graphs.AffectedGraph
AllNodes()
Solution Explorer
StartupPa
TimeForm
VM.cs
AutoTest.Gra
Properties
Reference
InvalidCo
NodeCon
TestDescr
AutoTest.Min
Properties
Reference
Extension
Properties
100 %
Ready
Ln 47
Col 26
Ch 26

```
    {
        _nodes.Add(affectedGraphNode.FullName, affectedGraphNode);
    }

public IEnumerable<AffectedGraphNode> AllNodes()
{
    return _nodes.Values;
}

public IEnumerable<NodeConnection> AllConnections()
{
    return _connections.Values;
}

public bool ContainsNode(string key)
{
    return _nodes.ContainsKey(key);
}

public AffectedGraph Merge(AffectedGraph graph2)
{
    var ret = new AffectedGraph();
    AddIfNotExist(ret, this.AllNodes());
    AddIfNotExist(ret, graph2.AllNodes());
    AddConnections(ret, this.AllConnections());
    AddConnections(ret, graph2.AllConnections());
    return ret;
}

private void AddConnections(AffectedGraph to, IEnumerable<NodeConnection> allConnections)
{
    foreach(var c in allConnections) to.AddConnection(c.From, c.To, c.IsForward);
```


ile Edit View Project Build Debug Team Data Tools Test ReSharper ContinuousTests Analyze Window Help
Debug
GetSpecificallyMangledName
AffectedGraph.cs*
TestDescriptor.cs
NodeConnection.cs
TestPathsGraphRiskClassifier.cs
PathGraphClassifierTests.cs
CoverageDistanceAn...phRiskClassifier.cs
Solution Explorer
AutoTest.Graphs.AffectedGraph
AllNodes()
StartupPa
TimeForm
VM.cs
AutoTest.Gra
Properties
Reference
InvalidCo
NodeCon
TestDescr
AutoTest.Min
Properties
Reference
Extension
Properties
100 %
eady
Ln 47
Col 26
Ch 26

```
            {
                _nodes.Add(affectedGraphNode.FullName, affectedGraphNode);
            }

        public IEnumerable<AffectedGraphNode> AllNodes()
        {
            return _nodes.Values;
        }

        public IEnumerable<NodeConnection> AllConnections()
        {
            return _connections.Values;
        }

        public bool ContainsNode(string key)
        {
            return _nodes.ContainsKey(key);
        }

        public AffectedGraph Merge(AffectedGraph graph2)
        {
            var ret = new AffectedGraph();
            AddIfNotExist(ret, this.AllNodes());
            AddIfNotExist(ret, graph2.AllNodes());
            AddConnections(ret, this.AllConnections());
            AddConnections(ret, graph2.AllConnections());
            return ret;
        }

        private void AddConnections(AffectedGraph to, IEnumerable<NodeConnection> allConnections)
        {
            foreach(var c in allConnections) to.AddConnection(c.From, c.To, c.IsForward);
```


A
X,B
C
V,W
r
Y
Z
p
q
R_4
R_3
R_2
R_1


AGLVisualizer.cs
MsAGLVisualization.cs
GraphVizGraphVisualization.cs
MSpecTranslator.cs
MSpecTranslatorTests.cs
GetProfiledCodeGraph.cs
AutoTest.Minimizer.TestIdentifiers.MSpecTranslator
TranslateGeneratedMethod(MethodDefinition definition)
Server Explorer
Toolbox
6

```
        //     If mspec changes the definition of their "It" this should still work
        public static FieldDefinition TranslateGeneratedMethod(MethodDefinition definition)
        {
            try
            {
                if (definition == null) return null;
                if (!definition.HasBody) return null;
                var constructor = definition.DeclaringType.Methods.FirstOrDefault(x => x.Name == ".ctor");
                if (constructor == null) return null;
                var body = constructor.Body;
                for (var i = 0; i < body.Instructions.Count; i++)
                {
                    var instruction = body.Instructions[i];
                    if (instruction.OpCode.Code == Code.Ldftn)
                    {
                        var reference = instruction.Operand as MethodReference;
                        if (reference == null) continue;
                        var resolved = reference.ThreadSafeResolve();
                        if (resolved == null) continue;
                        if (resolved.GetCacheName() == definition.GetCacheName())
                        {
                            var nextstsfld = GetNextInstructionAfter(i, body.Instructions, Code.Stsfld);
                            if (nextstsfld == -1) continue;
                            var cachedfield = body.Instructions[nextstsfld].Operand as FieldReference;
                            if (cachedfield == null) continue;
                            var nextldsfld = GetNextInstructionAfter(nextstsfld, body.Instructions, Code.Ldsfld);
                            if (nextldsfld == -1 ||
                                cachedfield.FullName !=
                                ((FieldReference) body.Instructions[nextldsfld].Operand).FullName) continue;
```