

usings

namespace Codice.Client.BaseCommands

Public class AclInfoPrinter

public void CalcAclInfo()

public string CalcExtendedAclInfo()

public statoc string GSeidName()

public string GetString()

private void GetEntries()

A

internal void CalculateOperations()

public void CalcAclEntry()

private void CalcPermissions()

private void Add()

private string GetString()

private void GetEntryString()

private const string STR_FORMAT

private SortedList mTable

Public class PrintableEntry

public string user

public string persmission

public string granted

public string denied

public string inherited

public PrintableEntry()

A

public enum NodeImageEnum

usings

namespace Codice.Client.BaseCommands

Public class AclInfoPrinter

public void CalcAclInfo()

public string CalcExtendedAclInfo()

public statoc string GSeidName()

public string GetString()

private void GetEntries()

private void CalcAclEntry()

private void CalcPermissions()

private void Add()

private string GetString()

private void GetEntryString()

private const string STR_FORMAT

private SortedList mTable

Public class PrintableEntry

public string user

public string persmission

public string granted

public string denied

public string inherited

public PrintableEntry()

usings

namespace Codice.Client.BaseCommands

Public class AclInfoPrinter

public void CalcAclInfo()

public string CalcExtendedAclInfo()

public statoc string GSeidName()

public string GetString()

private void GetEntries()

private void CalcAclEntry()

private void CalcPermissions()

private void Add()

private string GetString()

private void GetEntryString()

A

internal void CalculateOperations()

private const string STR_FORMAT

private SortedList mTable

Public class PrintableEntry

public string user

public string persmission

public string granted

public string denied

public string inherited

A

public enum NodeImageEnum

public PrintableEntry()

Standard layout. The unchanged methods are shown in gray color, the new methods (added) are shown in gree color, using an icon.

usings

namespace Codice.Client.BaseCommands

Public class AclInfoPrinter

public void CalcAclInfo()

public string CalcExtendedAclInfo()

public statoc string GSeidName()

public string GetString()

private void GetEntries()

A

 internal void CalculateOperations()

public void CalcAclEntry()

private void CalcPermissions()

private void Add()

private string GetString()

private void GetEntryString()

private const string STR_FORMAT

private SortedList mTable

Public class PrintableEntry

public string user

public string permission

public string granted

public string denied

public string inherited

A

 public PrintableEntry()

public enum NodeImageEnum

usings

namespace Codice.Client.BaseCommands

Public class AclInfoPrinter

public void CalcAclInfo()

public string CalcExtendedAclInfo()

public statoc string GSeidName()

public string GetString()

private void GetEntries()

private void CalcAclEntry()

private void CalcPermissions()

private void Add()

private string GetString()

private void GetEntryString()

private const string STR_FORMAT

private SortedList mTable

Public class PrintableEntry

public string user

public string permission

public string granted

public string denied

public string inherited

public PrintableEntry()

usings

namespace Codice.Client.BaseCommands

Public class AclInfoPrinter

public void CalcAclInfo()

public string CalcExtendedAclInfo()

public statoc string GSeidName()

public string GetString()

private void GetEntries()

private void CalcAclEntry()

private void CalcPermissions()

private void Add()

private string GetString()

private void GetEntryString()

A

 internal void CalculateOperations()

private const string STR_FORMAT

private SortedList mTable

Public class PrintableEntry

public string user

public string permission

public string granted

public string denied

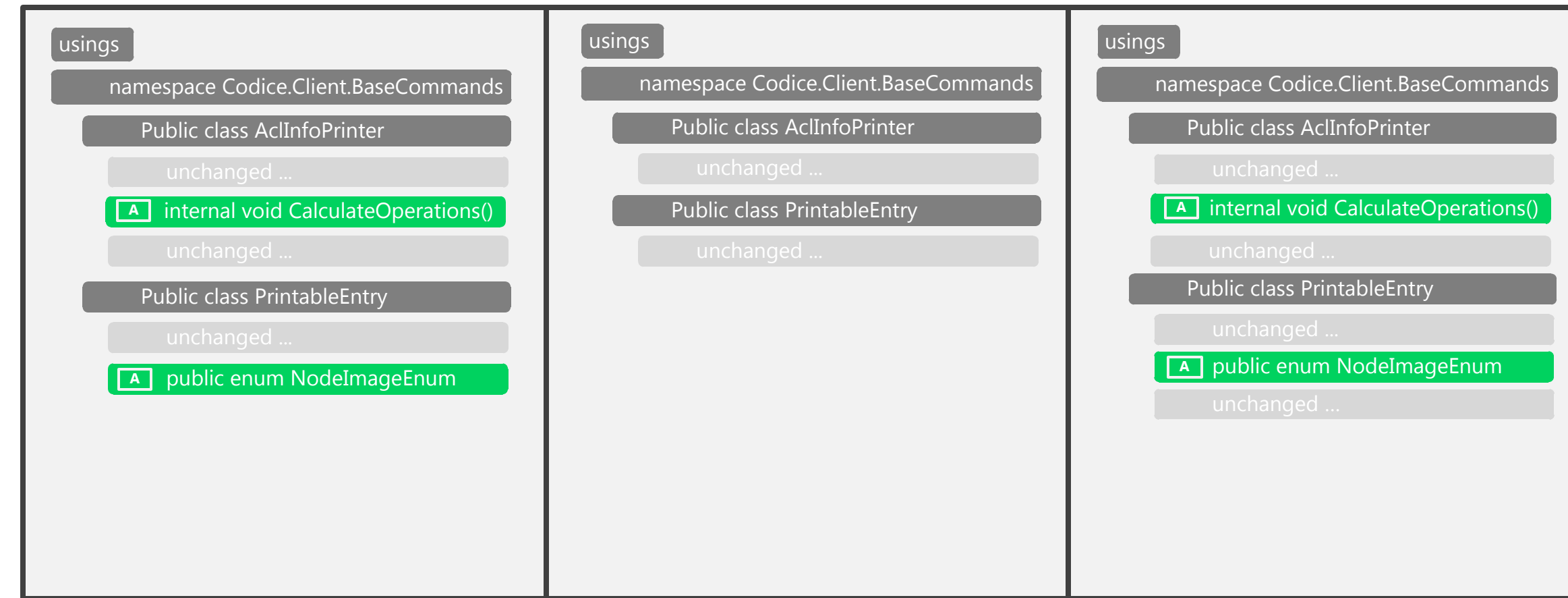
public string inherited

A

 public enum NodeImageEnum

public PrintableEntry()

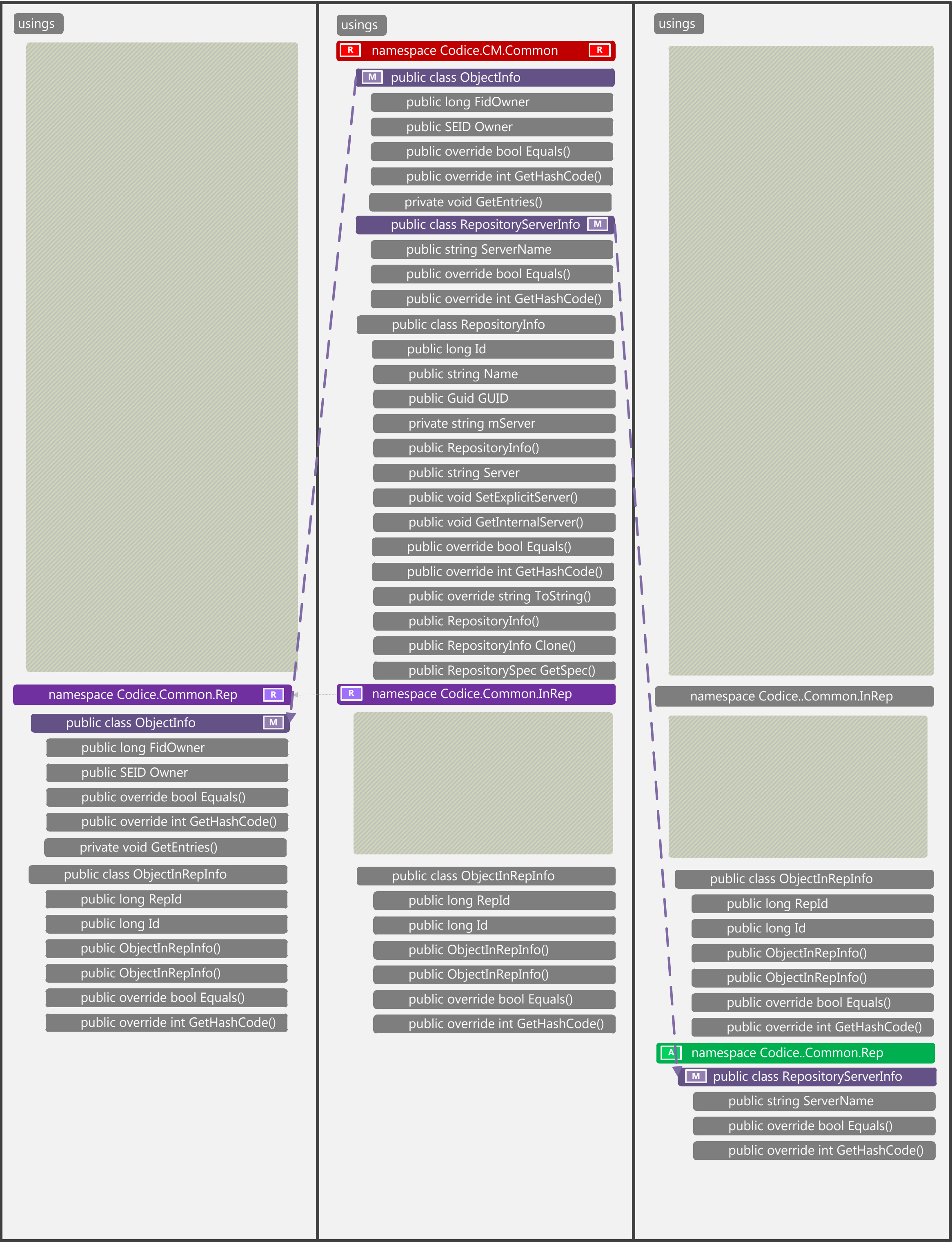
This layout adds an empty box to try to align methods, if you see a line, all methods are aligned, so they are in the same row.



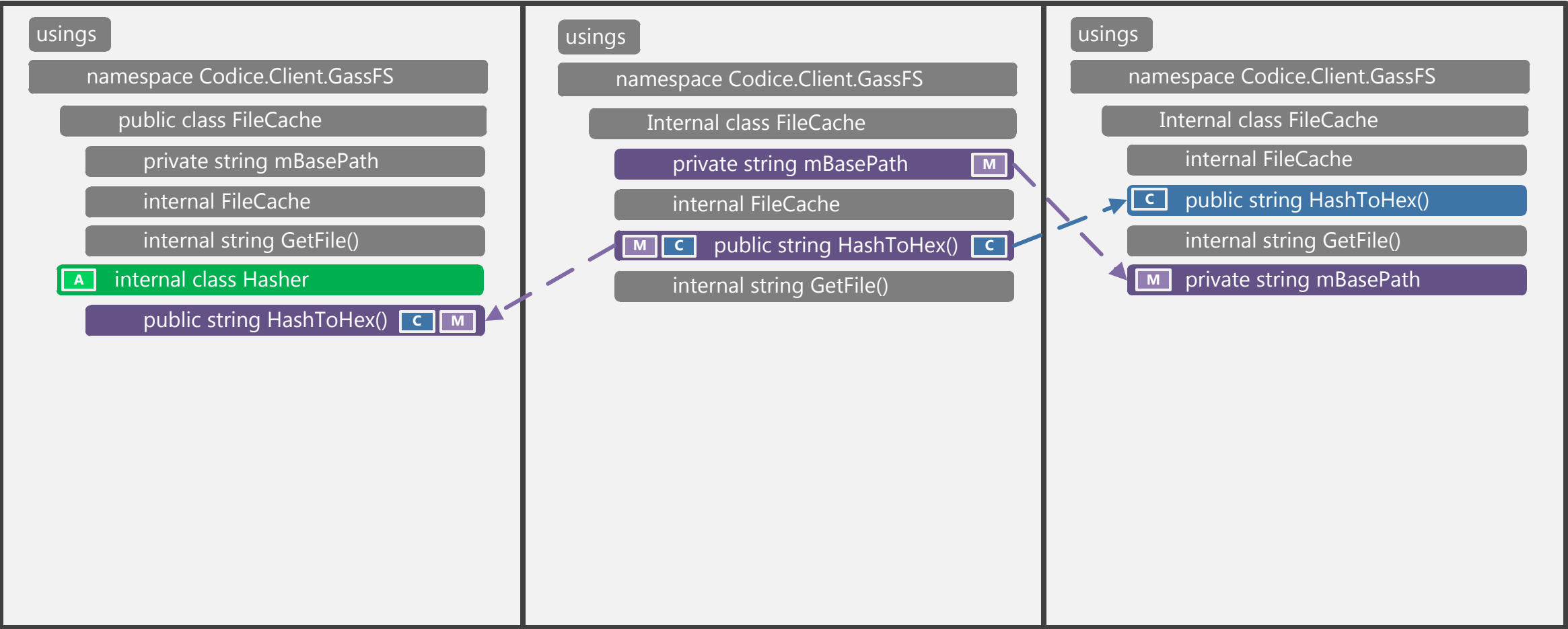
This is a way to hide/group unchanged methods, so the user can focus only in changed ones.



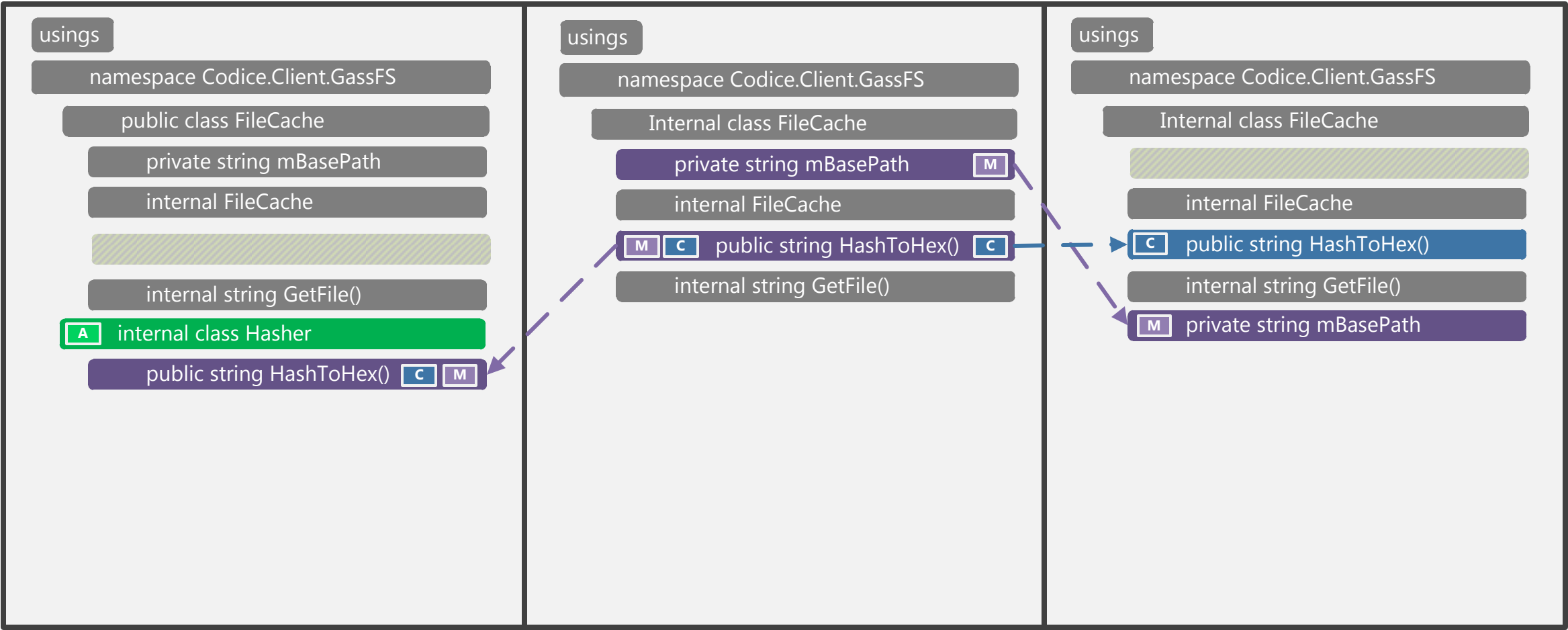
Other sample with the standard layout. Different colors used for moved and renamed.



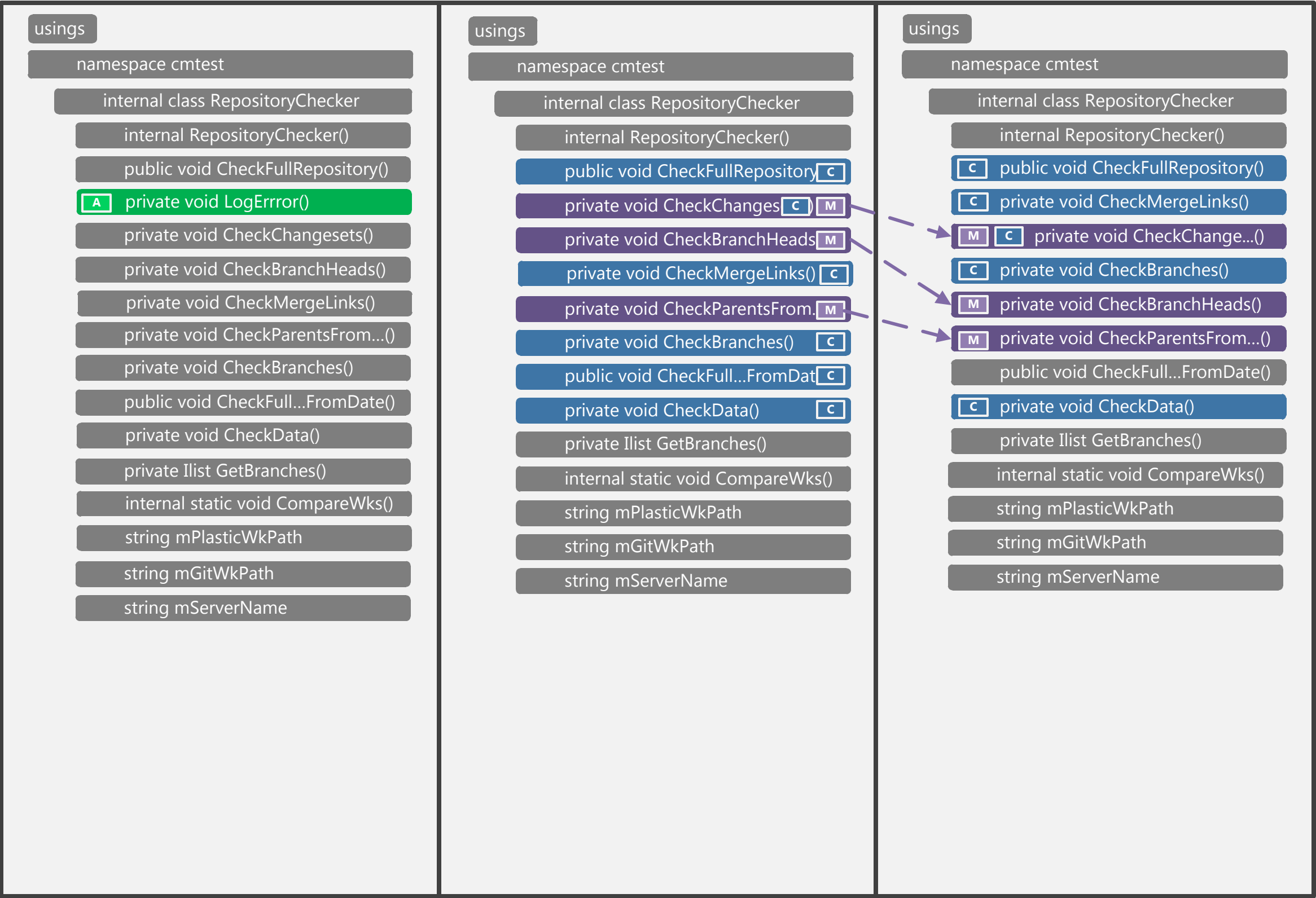
Other sample using blanks, in order to align methods in the same row.



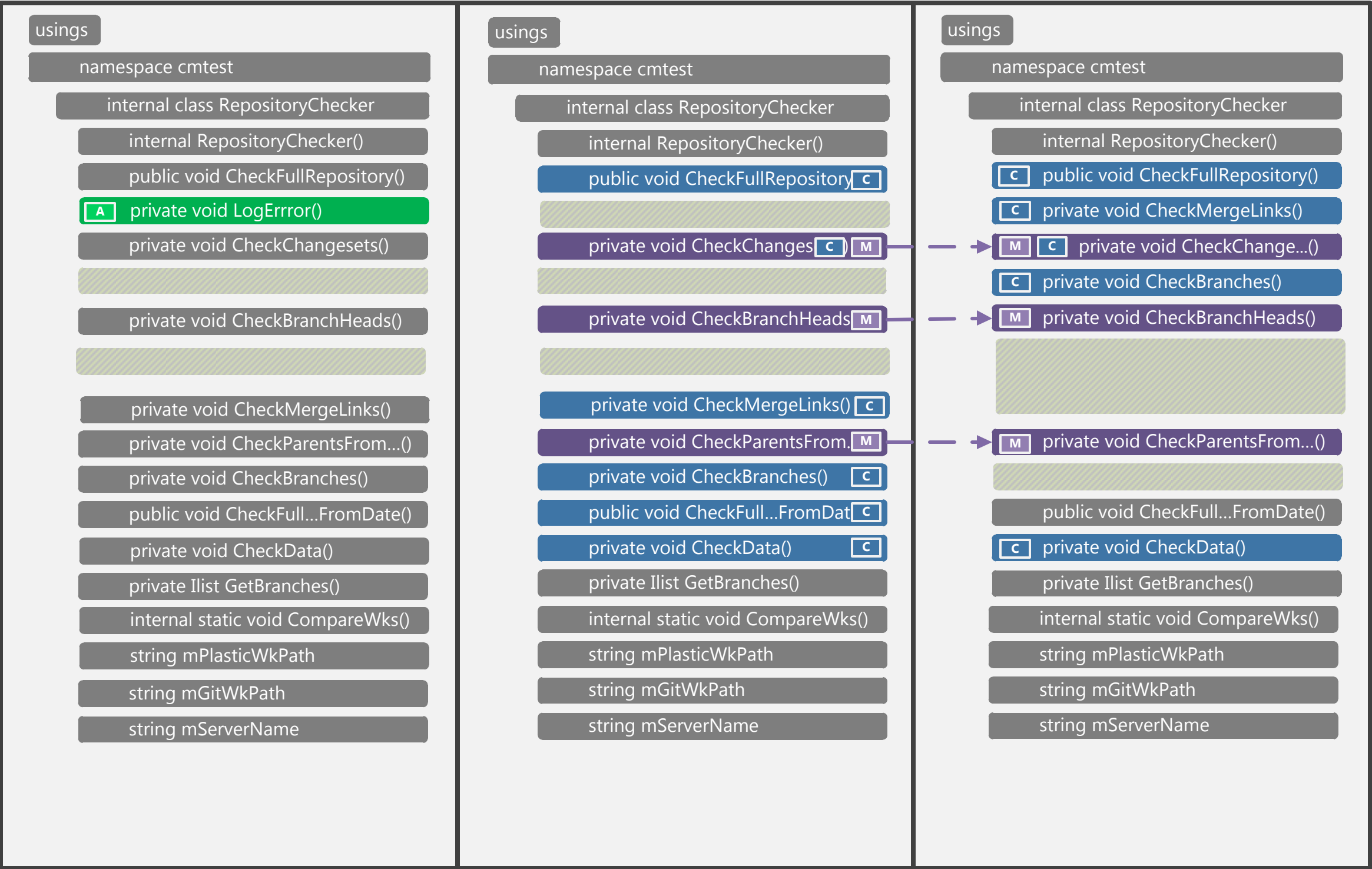
FileCache sample: Standard layout



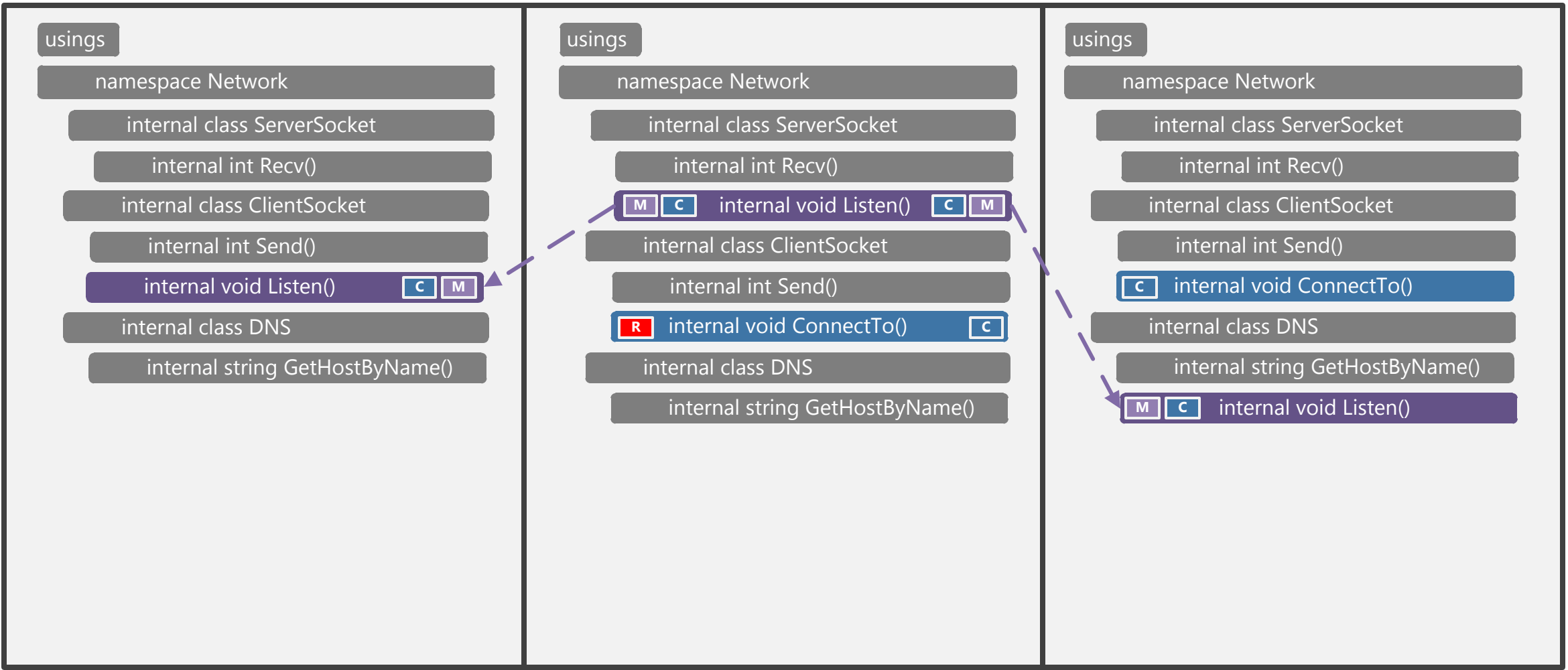
FileCache sample: Standard layout



RepositoryChecker sample:
Standard layout

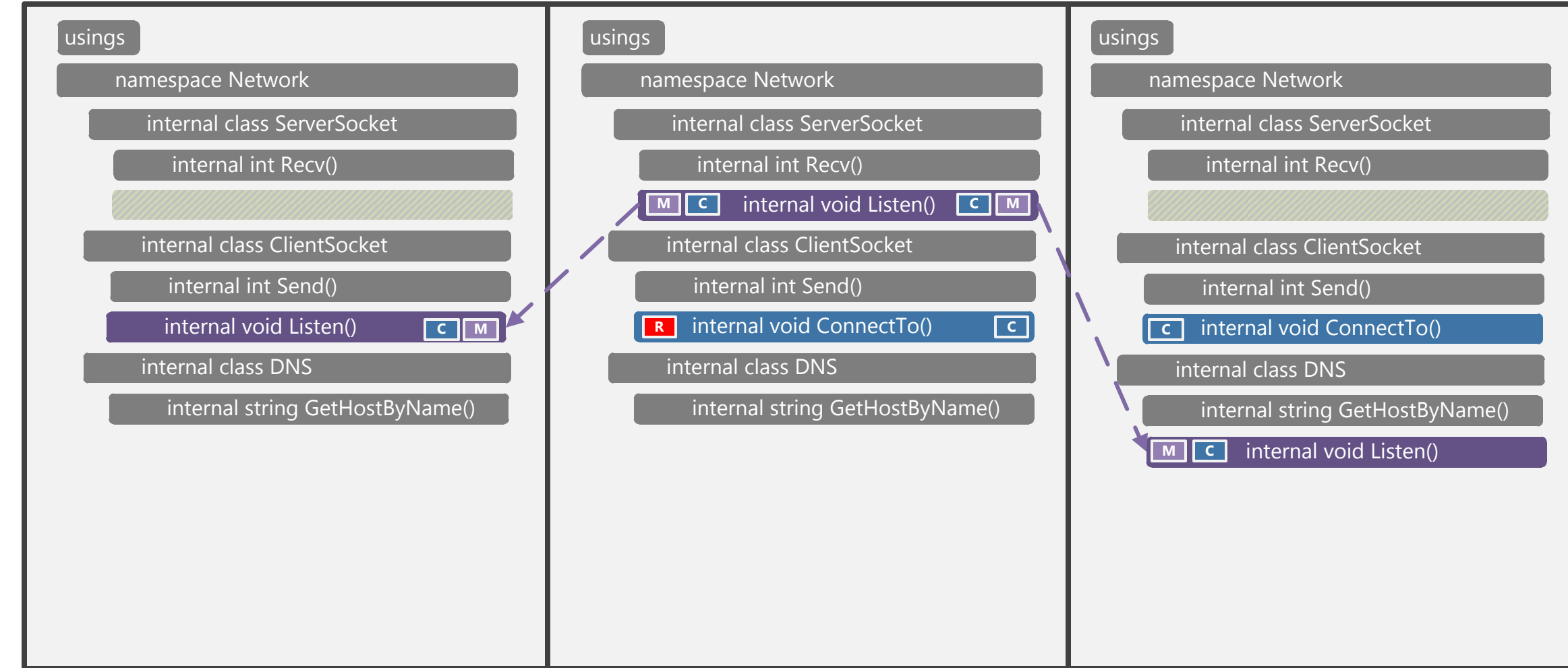


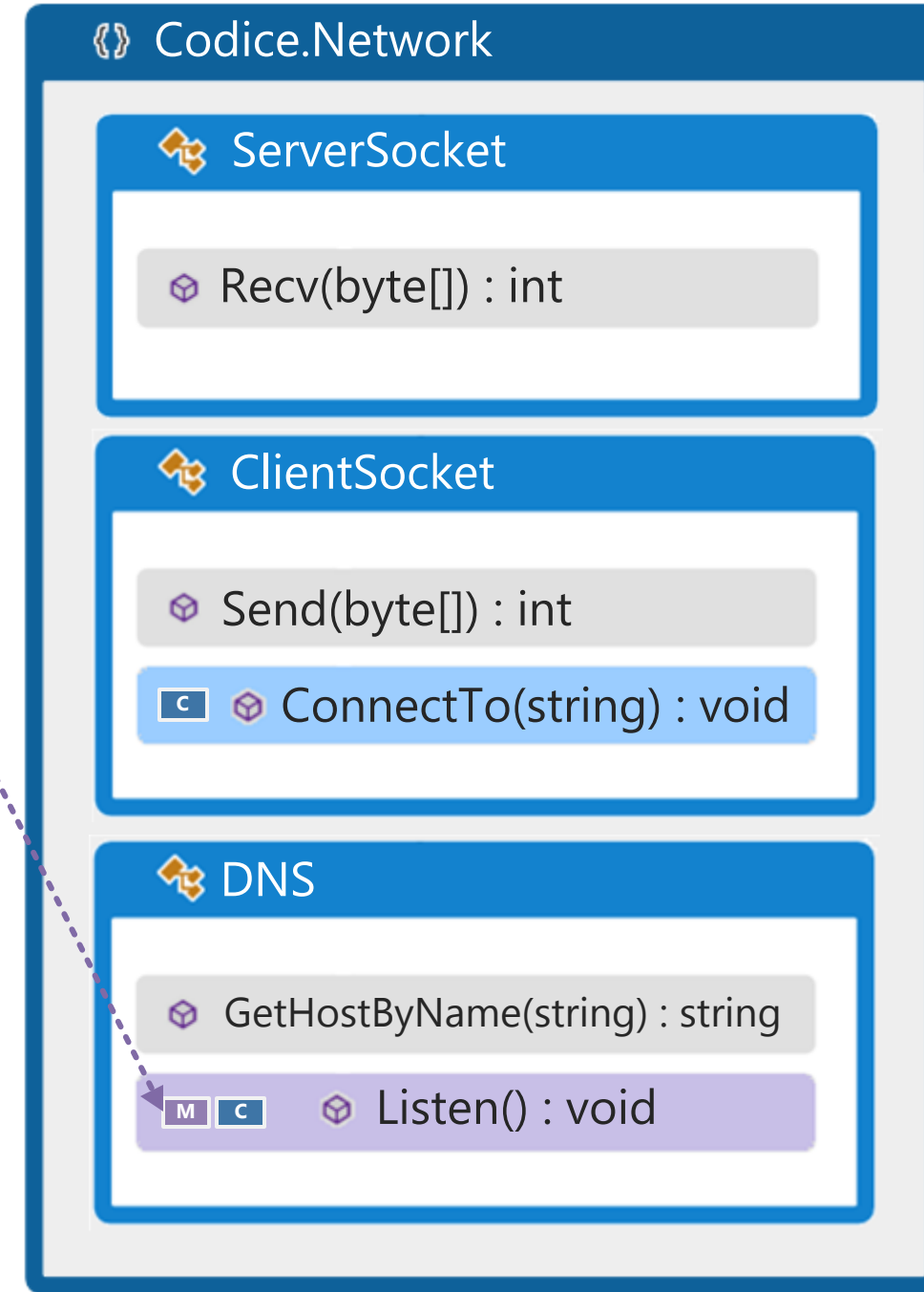
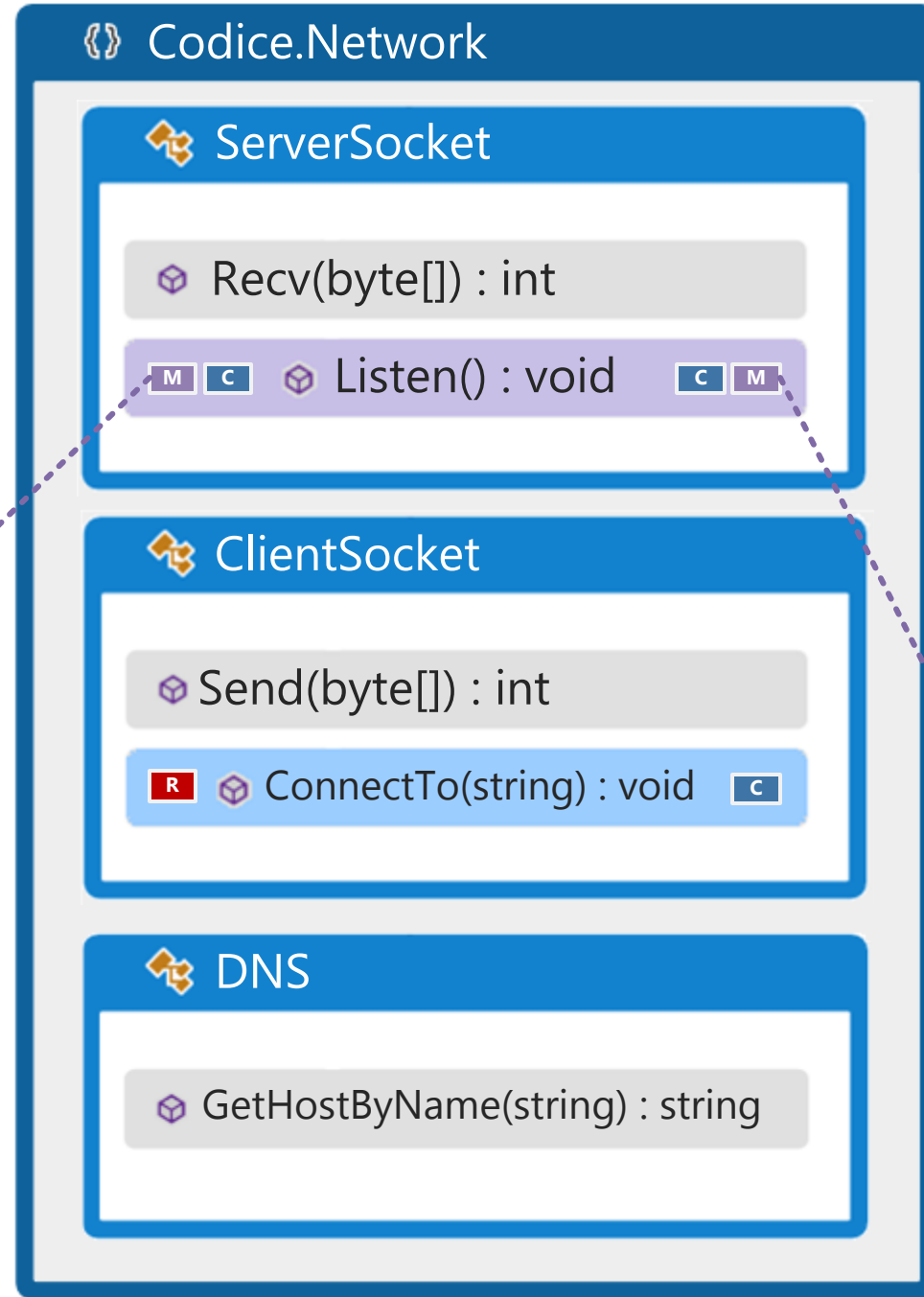
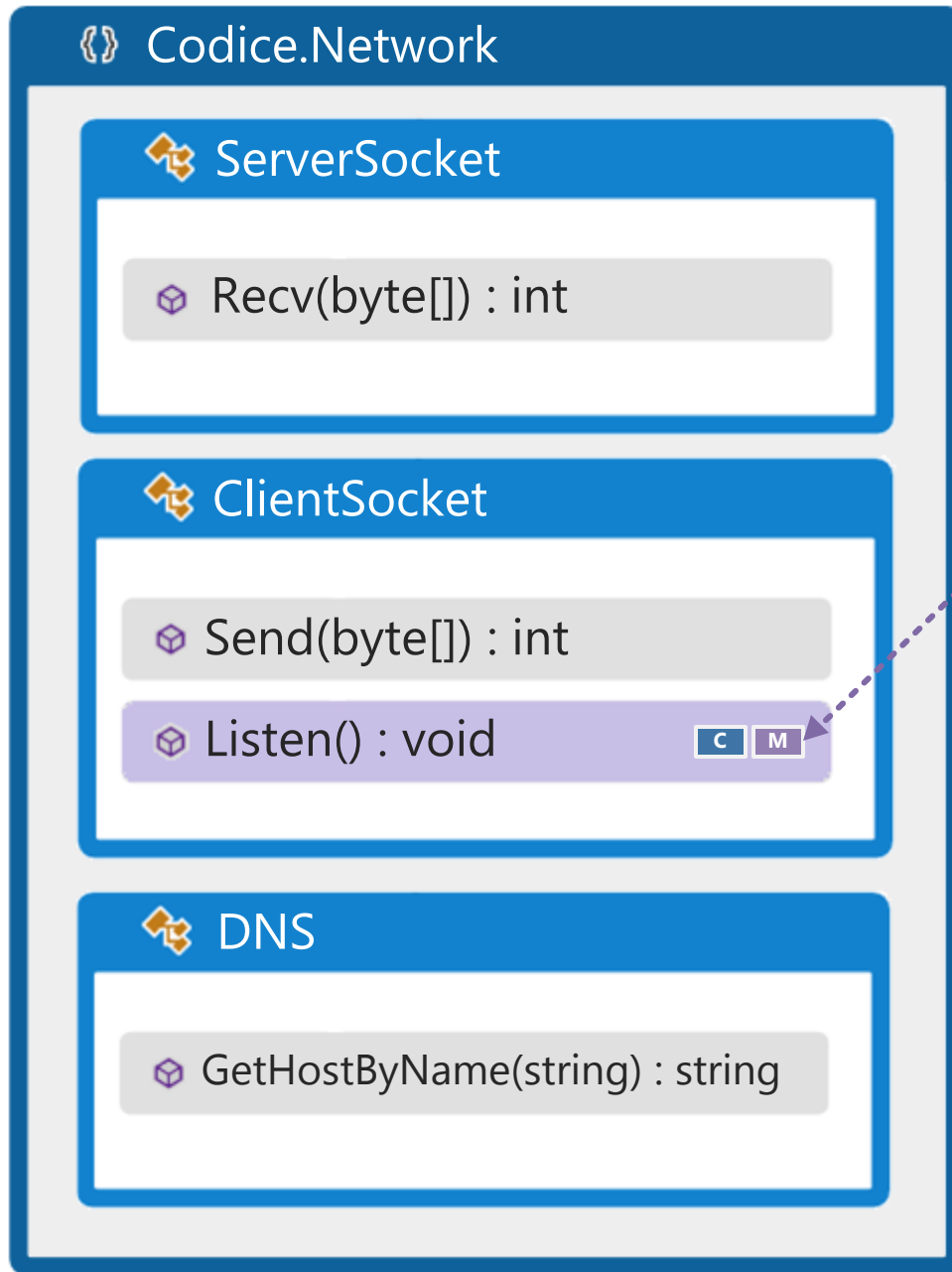
RepositoryChecker sample: using empty blocks to align methods.



SocketDeeper sample: Standard layout

Socket deeper sample: Using blanks to align methods.





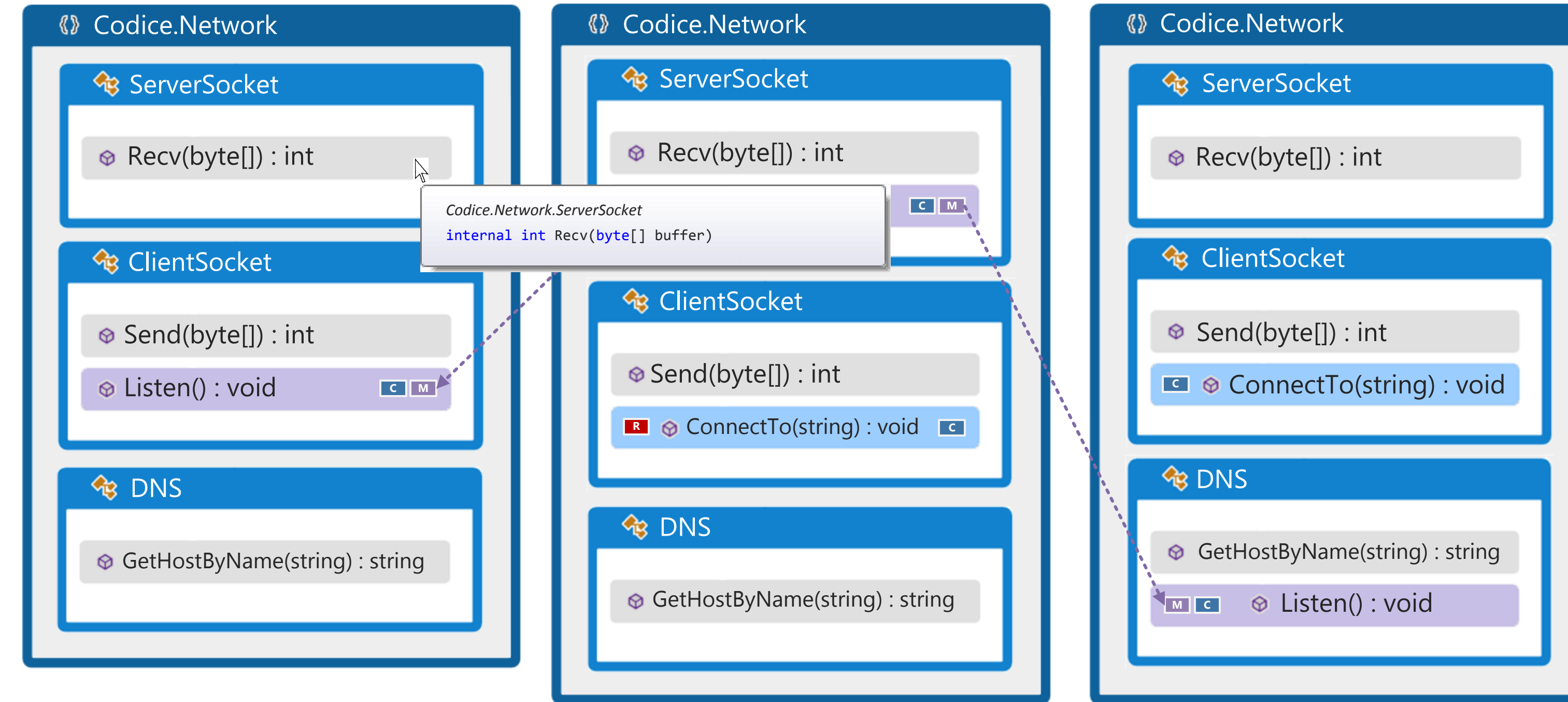
Alternative design. This is the SocketDeeper example, representing entities in other way. Different colors used by containers, for example, usings in black, namespaces in dark blue, and classes in blue. Containers groups its chids so its easy to see relationships. Also this model seems that doesn't need the blanks layout to align methods because it's easy to see relationships, using the arrows.

Method signature is not complete

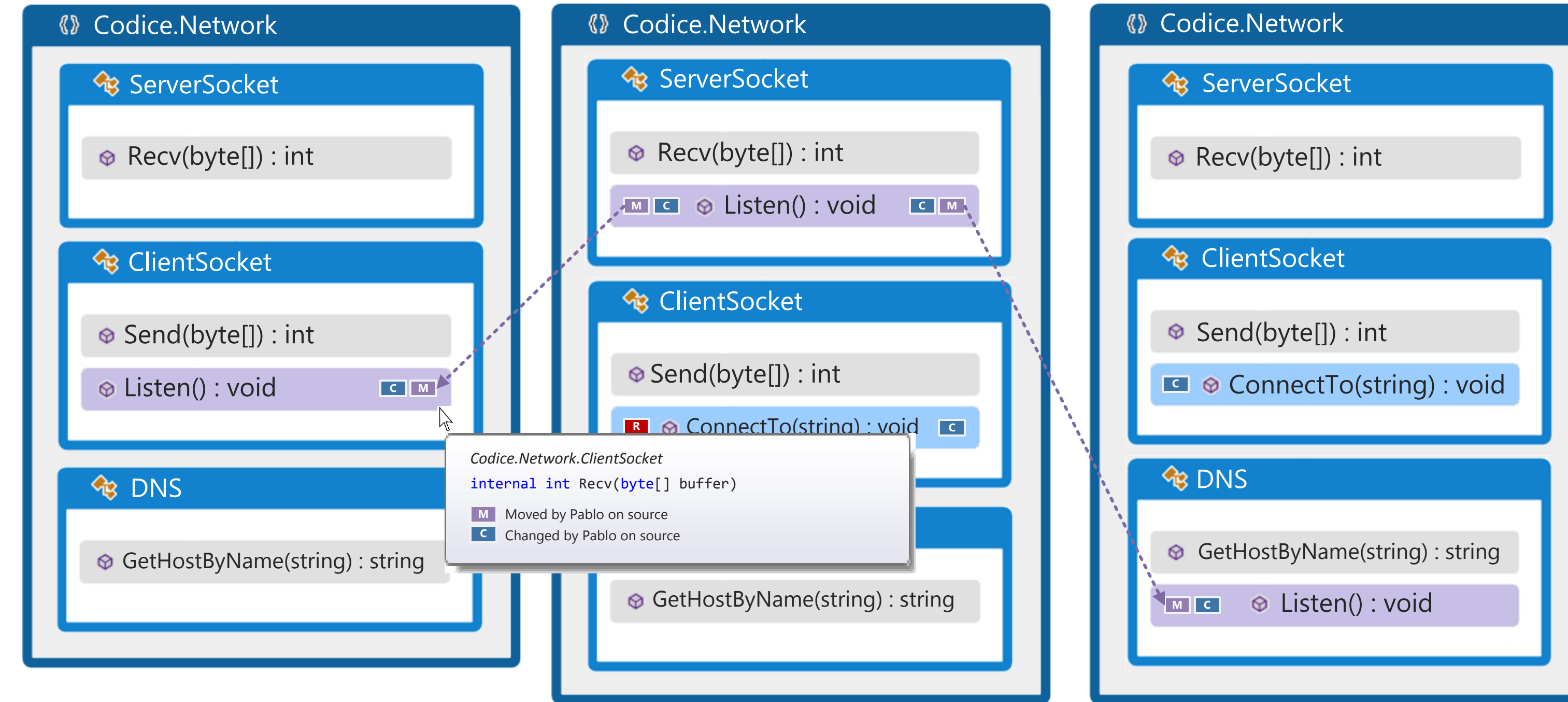
- 1) Method name
- 2) Optional parameters
- 3) : return type

Use icons for private, public, internal, sealed, method, enum, class, etc ...

The connectors between moves will go from the M source icon to the M dst icon



When you place the mouse hover a method, it will show a tooltip with the whole method signature.



Nodes that have operations involved, will show that information in their tooltip.

Codice.Client.BaseCommands

AclInfoPrinter

CalcAclInfo() : void

CalcExtendedAclInfo() : string

GetPrintableSEIDName() : string

GetString() : string

A

CalculateOperationsToApply() : void

CalcAclEntry() : void

CalcPermissions() : void

Add() : void

GetString() : void

GetEntryString() : void

STR_FORMAT

mTable

ClientSocket

user

permission

granted

denied

inherited

PrintableEntry()

A

NodeImageEnum

Codice.Client.BaseCommands

AclInfoPrinter

CalcAclInfo() : void

CalcExtendedAclInfo() : string

GetPrintableSEIDName() : string

GetString() : string

CalcAclEntry() : void

CalcPermissions() : void

Add() : void

GetString() : void

GetEntryString() : void

STR_FORMAT

mTable

ClientSocket

user

permission

granted

denied

inherited

PrintableEntry()

Codice.Client.BaseCommands

AclInfoPrinter

CalcAclInfo() : void

CalcExtendedAclInfo() : string

GetPrintableSEIDName() : string

GetString() : string

CalcAclEntry() : void

CalcPermissions() : void

Add() : void

GetString() : void

GetEntryString() : void

A

CalculateOperationsToApply() : void

STR_FORMAT

mTable

ClientSocket

user

permission

granted

denied

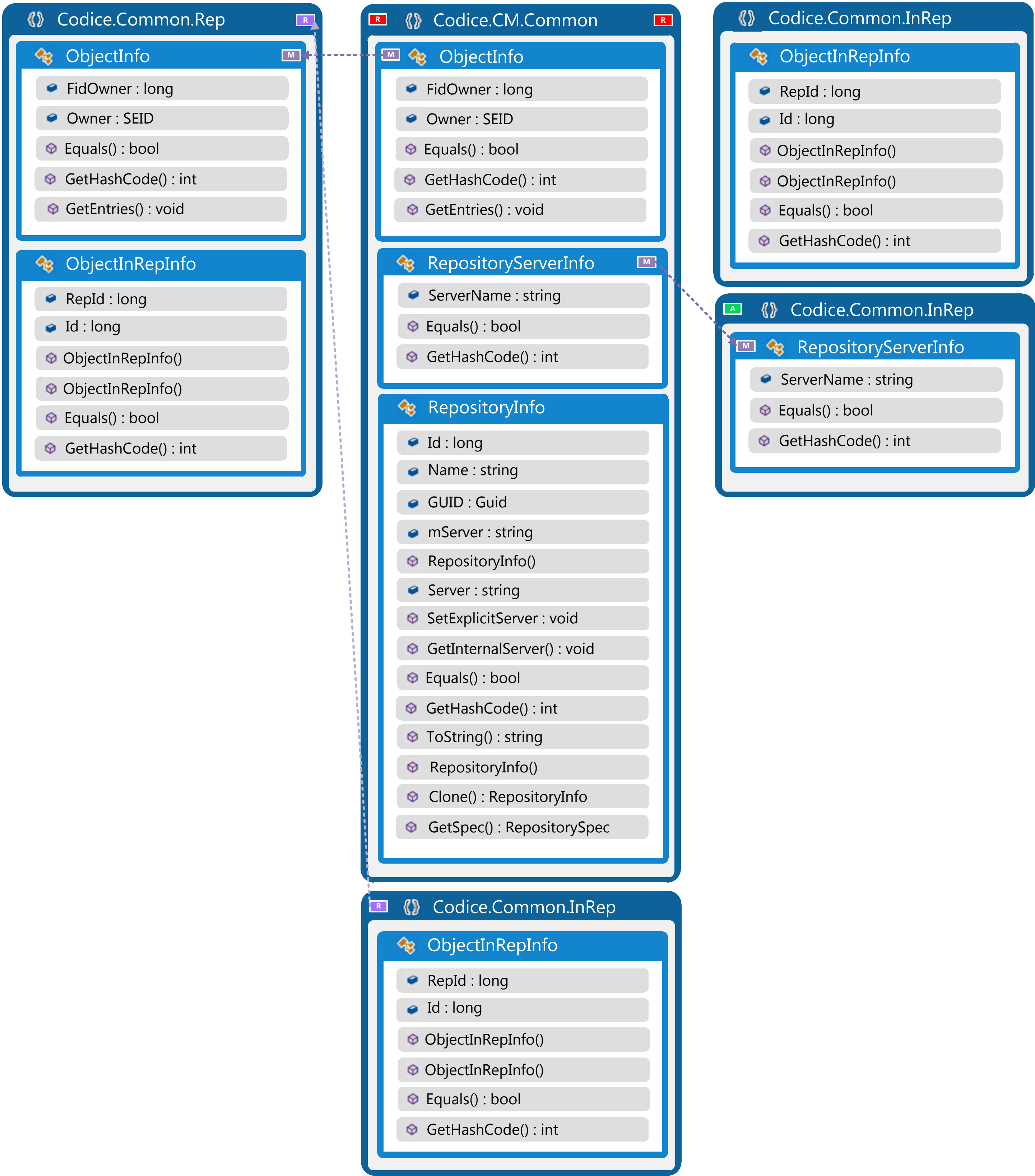
inherited

A

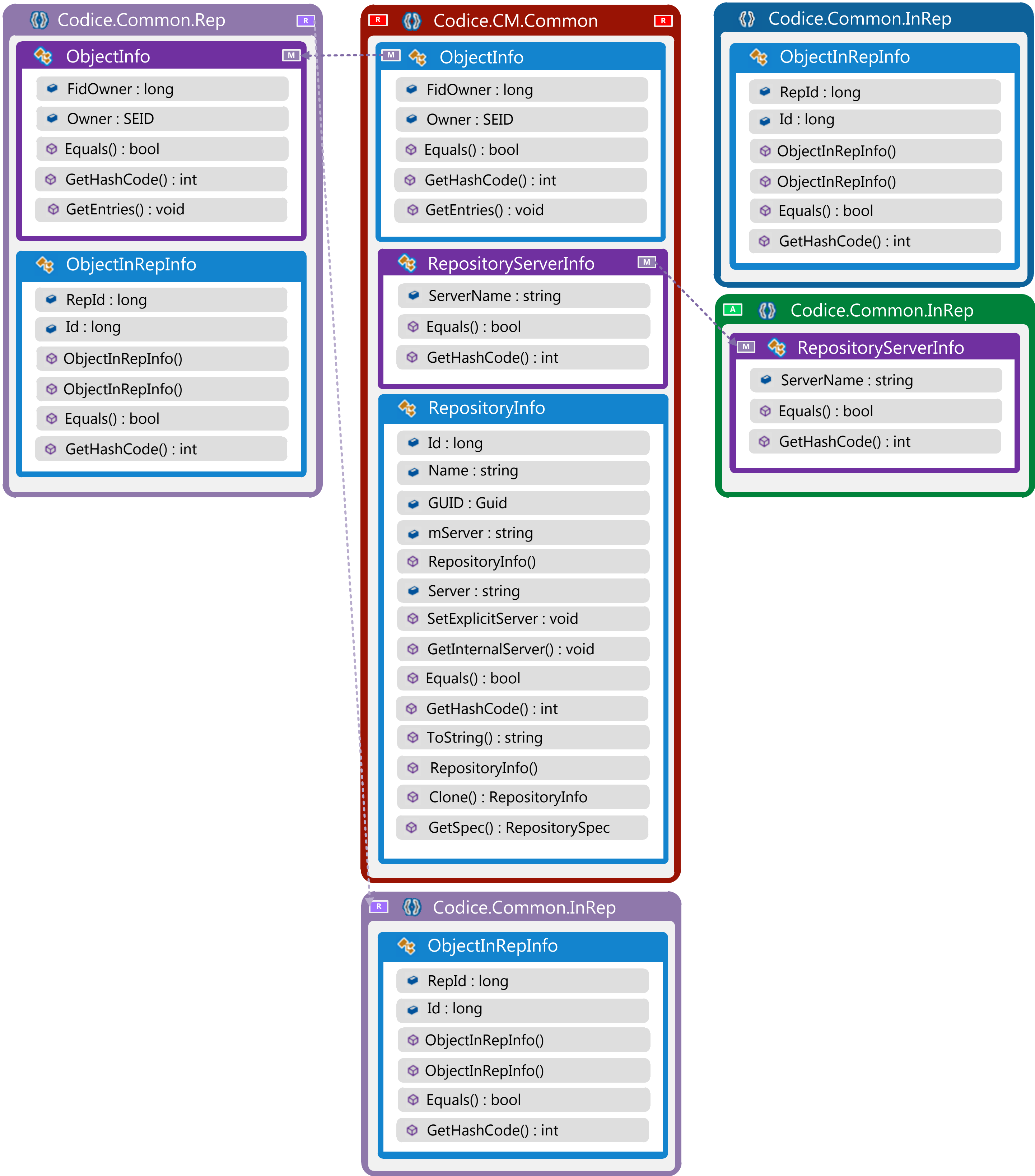
NodeImageEnum

PrintableEntry()

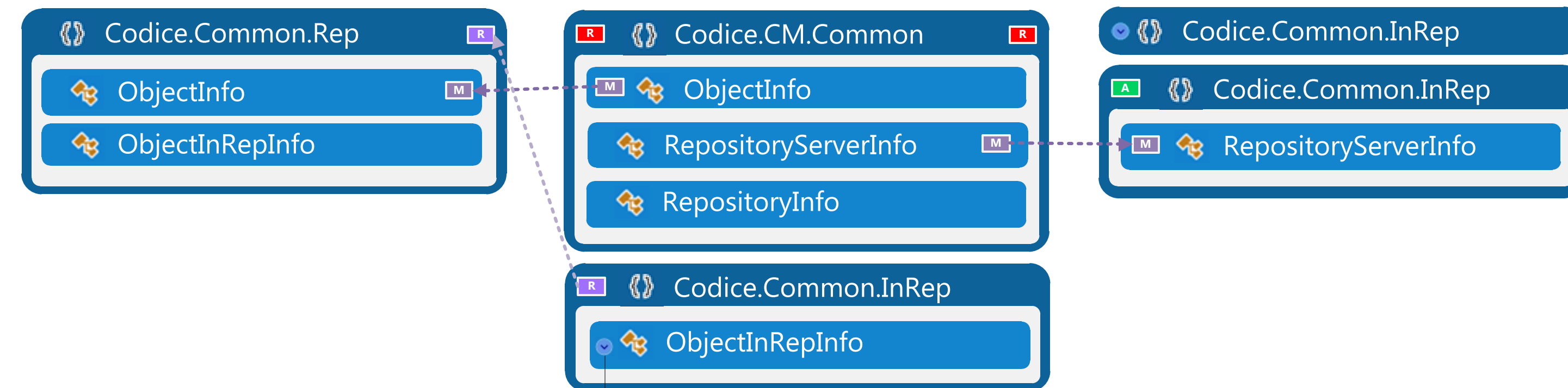
AclInfoPrinter sample showing the new design.



AdvancedObjectInfo showing the new design



When a namespace or class (in general a container) has been added, moved or renamed, we could change its color, but maybe it makes more difficult to understand the colors. Maybe a single rectangle in the text background is good enough



Unchanged nodes will be collapsed by default. All containers will be collapsable, using a button that will be shown when you place the mouse hover the container (see ObjectInRepInfo @ base file in the sample).