

IOSP

Integration Operation Segregation Principle

**WE PLAN
AND OPTIMISE
EVERYTHING
THAT MOVES PEOPLE
AND GOODS
WORLDWIDE.**



the mind of movement

PTV Map&Guide internet

https://mginter.mapandguide.com/v5.6/app.html?language=de&user=stefan.hock@ptvgroup.com#sunrise:start

Drucken Mein Konto Einstellungen System Hilfe in Hilfe suchen Abmelden

Routenplanung 1

Route 2 Optimieren

Neu Öffnen Speichern Frachten PTV Navigator Transcamion

Anwendungsfall Route Finde Navigation Fahren

Dornach - Dornach

Planung Wegliste Kosten Emissionen Frachtenbörse

Land Adresse Suchbegriff

Aschheim, Humboldtstraße 12 z.B. "Truck Wash"

Geben Sie die Straße mit einem Komma getrennt ein.

Stationsliste

PLZ	Ort	Straße Hausnummer	A ...
1 85609	Aschheim Dornach	Humboldtstraße 12	C
2 85609	Aschheim Dornach	Einsteinring	K

Favoriten Importieren Exportieren Bearbeiten

Markierte Stationen: 0/2

Einstellungen

Fahrzeug

Eigener PKW

Zeiten

Abfahrt 17.05.2017 16:21

Routingparameter

Ladung Verkehr Beschränkungen Wegkorridor Zeitfenster Lenk- und Ruhezeiten

Ergebnis der Wegberechnung

Sichtbarkeit	Info	Wegstrecke	Mautstrecke	Abfahrt	Ankunft	Reisezeit	Fahrzeit	Wegkosten Σ	Mautkosten...	Fixkosten	Gesamtkos...	Fahren-Tun...	Autobahn	CO ₂ e
<input checked="" type="checkbox"/>	Route 1	-	0,92 km	0,00 km	16:21	16:23	0:01 h	0:01 h	-	-	-	-	-	0,18 kg

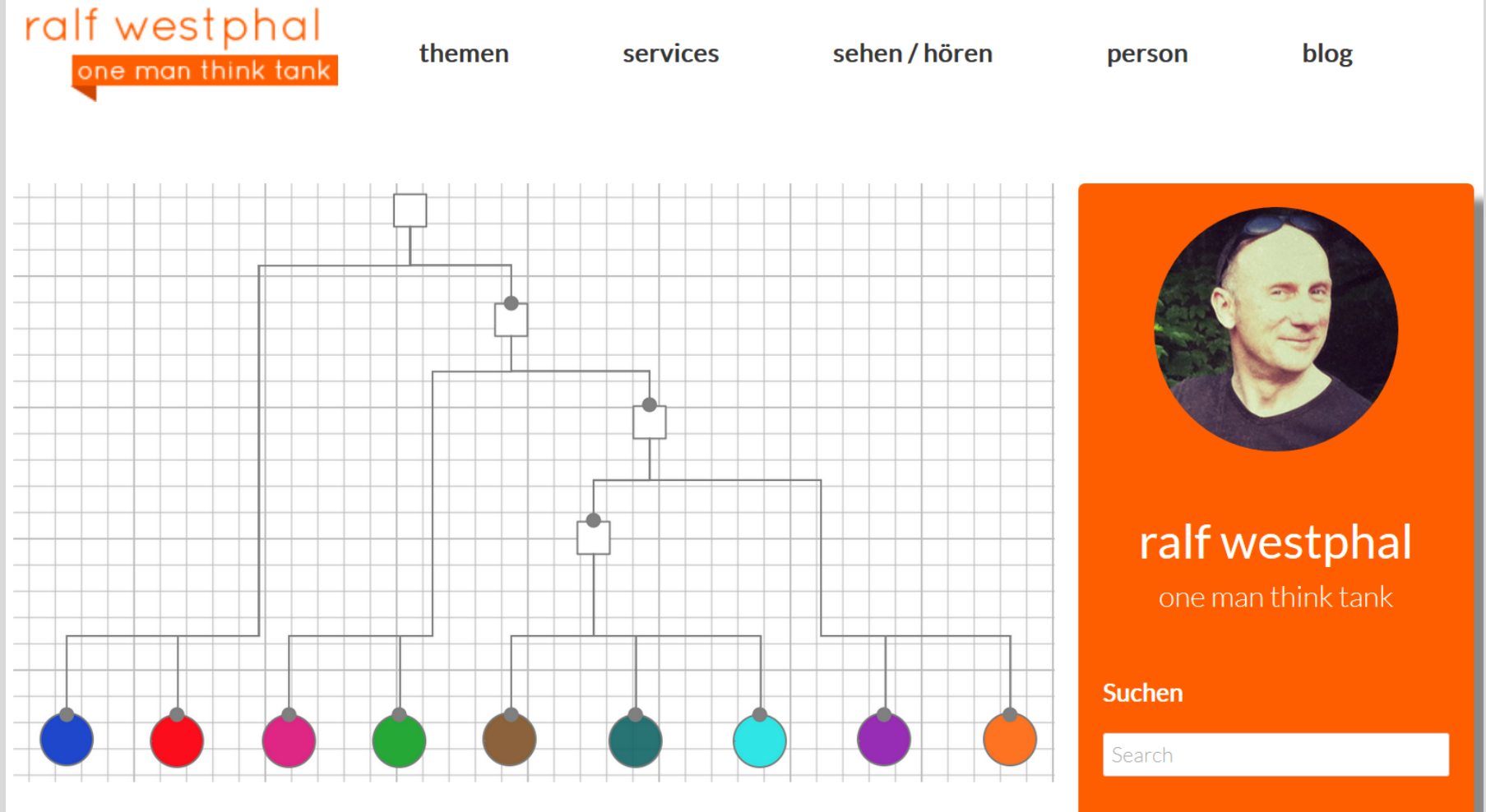
Feedback geben

HOW DO PEOPLE REACH THEIR DESTINATIONS?

**How do we implement several
CCD principles by only one?**

INTEGRATION OPERATION SEGREGATION PRINCIPLE

➔ Invented by



INTEGRATION OPERATION SEGREGATION PRINCIPLE

PoMO

DIP

IOSP

SRP

SoC

SLA

INTEGRATION OPERATION SEGREGATION PRINCIPLE



INTEGRATION OPERATION SEGREGATION PRINCIPLE

➡ Operator

```
public DateTime UnixTimestampToDateTime(long timestamp)
{
    try
    {
        if (timestamp == 0)
        {
            return DateTime.Now;
        }
        else
        {
            DateTime dateTime = new DateTime(1970, 1, 1, 0, 0, 0, 0, DateTimeKind.Utc);
            dateTime = dateTime.AddSeconds(timestamp).ToUniversalTime();
            return dateTime;
        }
    }
    catch (Exception ex)
    {
        logger.Error("UnixTimestampToDateTime", ex);
        throw;
    }
}
```

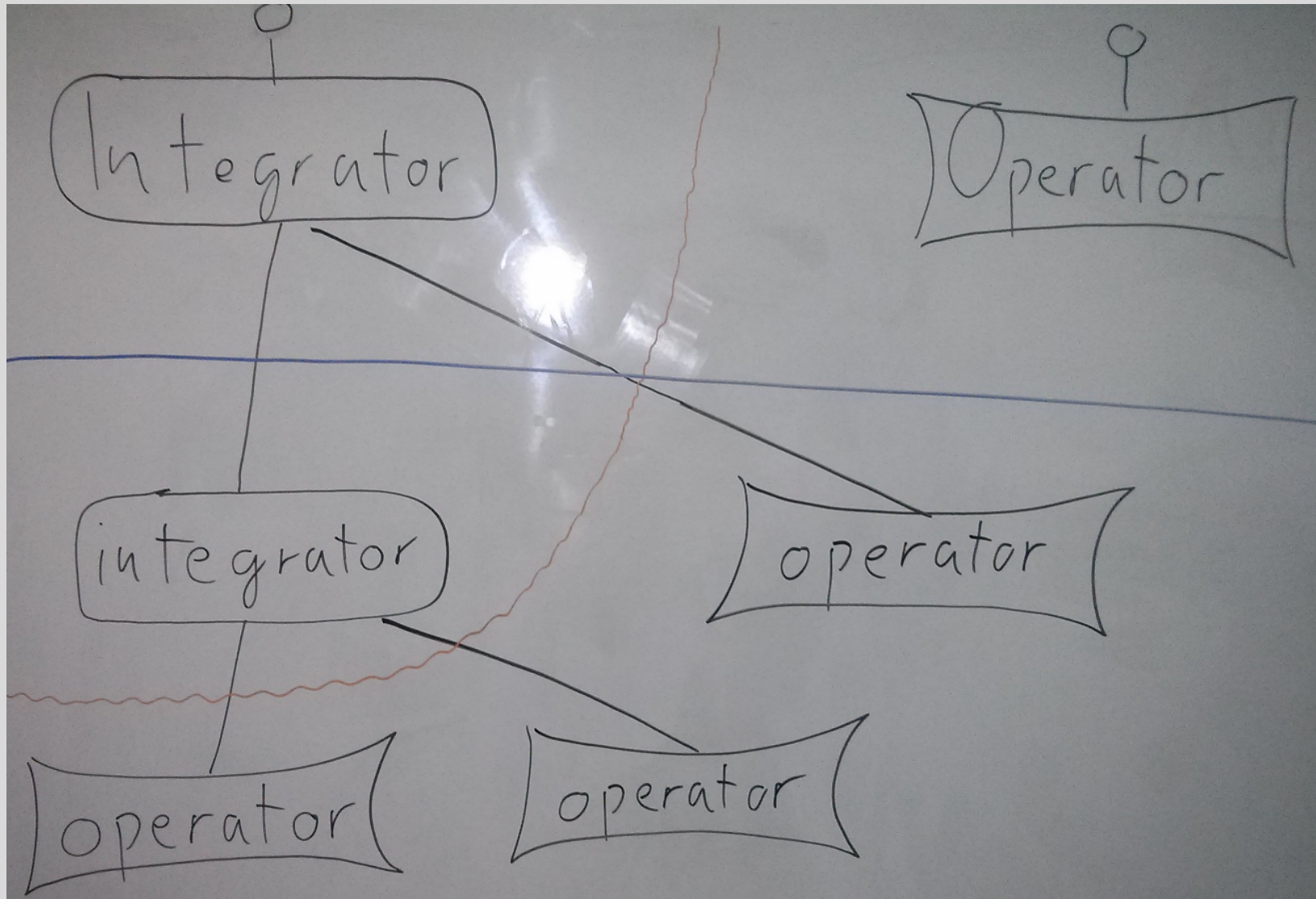
INTEGRATION OPERATION SEGREGATION PRINCIPLE

➡ Integrator

```
private EmissionHBEFA buildEmissionHBEFA(double distance, Generated.EmissionValues_HBEFA_3_2 values, FuelType fuelType)
{
    EmissionHBEFA result      = buildEmissionHBEFABody(distance, values);
    double consumption        = buildConsumption(values.fuelConsumption, fuelType);
    result.Consumption         = roundConsumption(consumption);
    result.StandardConsumption = buildStandardConsumptionHBEFA(distance, consumption);
    result.CO2e                = buildCO2e(values);

    return result;
}
```

INTEGRATION OPERATION **SEGREGATION** PRINCIPLE



HOW TO TEST?

- integrator + public => integration test
- integrator + private => no test
- operator => unit test
 - No mockups needed!



IN PRACTISE!?

[Integrator]

4 references | 0/3 passing

```
public List<string> Do()
{
    try
    {
        foreach (SyntaxTree unit in syntaxTrees)
        {
            textBuilder.InitFilename(unit.FilePath);
            if (codeCoverage.CheckTestProject())
            {
                nodes(unit);
            }
            else
            {
                textBuilder.ErrorIOSP005();
            }
        }
        textBuilder.Counter();
        return textBuilder.Result();
    }
    catch (Exception ex)
    {
        logger.Error("CodeAnalysis.Units", ex);
        throw;
    }
}
```

```
[AttributeUsage(AttributeTargets.Method | AttributeTargets.Constructor)]
```

0 references

```
public class Integrator : Attribute  
{  
}
```

```
[AttributeUsage(AttributeTargets.Method)]
```

0 references

```
public class Operator : Attribute  
{  
}
```

```
[AttributeUsage(AttributeTargets.Method)]
```

```
//Methods with this flag are not yet marked as Integrator or Operator
```

0 references

```
public class Todo : Attribute  
{  
}
```

```
[AttributeUsage(AttributeTargets.Method)]
```

```
//Methods with this flag are to be deleted as soon as possible
```

0 references

```
public class Deprecated : Attribute  
{  
}
```

```
[AttributeUsage(AttributeTargets.Method)]
```

```
//Methods with this flag have no implementation or only a default return value.
```

```
//They need no classification and no tests
```

0 references

```
public class Empty : Attribute  
{  
}
```

[Todo]

0 references

```
public Dictionary<string, object> FullTextPoiSearch(SRAdvancedSearchRequest request)
{
    logger.Debug("User " + SessionManager.GetTenantId() + ";" + SessionManager.GetUserId() + " starts FullTextPoiSearch");
    bool logFullTextPoiSearch = Convert.ToBoolean(ConfigurationManager.AppSettings["LogFullTextPoiSearch"]);
    Stopwatch stopwatch = new Stopwatch();
    stopwatch.Start();
    Dictionary<string, object> response = null;
    if (request.Address == null || Regex.Match(request.Address, @"^[!-\/\:\-@\[-\`\\{\|\}\~\€\s]*$").Success)
    {
        request.Address = string.Empty;
    }
    else
    {
        request.Address = request.Address.Trim();
    }

    # region special_search

    if (request.Address.StartsWith(ShortCuts.SHORTCUT_BORDER_CROSSINGS, StringComparison.InvariantCultureIgnoreCase))
    {
        response = SpecialSearchBorderCrossings(request);
        stopwatch.Stop();
        if (logFullTextPoiSearch && SessionManager.GetBool(SessionManager.LOGGING_ENABLED_FOR_USER))
        {
            SaveFullTextPoiSearchLog(request, stopwatch.ElapsedMilliseconds, response.Count);
        }
        return response;
    }

    if (request.Address.StartsWith(ShortCuts.SHORTCUT_FERRIES, StringComparison.InvariantCultureIgnoreCase) ||
        (request.Address.StartsWith(ShortCuts.SHORTCUT_DATAMANAGER, StringComparison.InvariantCultureIgnoreCase))) //Rec
    {
        Dictionary<string, object> mergedResult = new Dictionary<string, object>();

        if (GlueCode.IsDataManagerEnabled())
        {
            Dictionary<string, object> dmLayerResult = SpecialSearchDataManagerLayerByText(request);
            dmLayerResult.ToList().ForEach(x => mergedResult.Add(x.Key, x.Value));
        }

        if (request.Address.StartsWith(ShortCuts.SHORTCUT_FERRIES, StringComparison.InvariantCultureIgnoreCase))
        {
            Dictionary<string, object> ferryResult = combinedTransport.SpecialSearchByText(request);
            ferryResult.ToList().ForEach(x => mergedResult.Add(x.Key, x.Value));
        }
    }
}
```

```
[Operator]  
public bool CanBeEdited() => true;
```

```
[Todo]  
public string Date() => iReadWriteExif.Date();
```

```
[Todo]  
public string Time() => iReadWriteExif.Time();
```

```
[Empty]  
public void Dispose()  
{  
    thumbnail.Dispose();  
    iReadWriteExif.Dispose();  
    iReadWriteExif = null;  
}
```

[Deprecated]

[JsonRpcMethod(Name = "GetObjectInfos", Idempotent = true)]

0 references

```
public JsonObject[] GetObjectInfos(int timezoneOffset)
{
    try
    {
        RenderObjectInfo[] objectInfos = (RenderObjectInfo[])SessionManager.GetObject(SessionManager.LAYER_O_INFOS);

        if (objectInfos == null)
        {
            return new JsonObject[0];
        }
        JsonObject[] smallObjectInfos = new JsonObject[objectInfos.Length];

        for (int i = 0; i < objectInfos.Length; i++)
        {
            if (objectInfos[i].DataSource.StartsWith(DMLayerConfigFactory.DATA_MANAGER_RENDERING_PREFIX))
            {
                double minuteOffset = Convert.ToDouble(timezoneOffset);

                if (!string.IsNullOrEmpty(objectInfos[i].Description1) && objectInfos[i].Description1.Contains("|"))
                {
                    string[] multiDesc = objectInfos[i].Description1.Split("|".ToCharArray(), StringSplitOptions.RemoveEmptyEntries)
                    for (int ii = 0; ii < multiDesc.Length; ii++)
                    {
                        multiDesc[ii] = (string)DataManagerUtils.GetLocalTime(minuteOffset, multiDesc[ii]);
                    }
                    objectInfos[i].Description1 = string.Join("|", multiDesc) + "|";
                }
            }
            else
            {
                if (!string.IsNullOrEmpty(objectInfos[i].Description1))
                {
                    objectInfos[i].Description1 = (string)DataManagerUtils.GetLocalTime(minuteOffset, objectInfos[i].Description1);
                }
                if (!string.IsNullOrEmpty(objectInfos[i].Description2))
                {
                    objectInfos[i].Description2 = (string)DataManagerUtils.GetLocalTime(minuteOffset, objectInfos[i].Description2);
                }
                if (!string.IsNullOrEmpty(objectInfos[i].Description3))
                {

```

IOSP FORCES OTHER PRINCIPLES

- Keep it simple, stupid (KISS)
- Refactoring
- Single Level of Abstraction (SLA)
- ~~➤ Single Responsibility Principle (SRP)~~
- ~~➤ Separation of Concerns (SoC)~~
- Dependency Inversion Principle (DIP)
- Mockups
- Know-how transfer
- Code Coverage Analysis

SINGLE LEVEL OF ABSTRACTION (SLA) DONE BY IOSP

[Integrator]

3 references

```
private EmissionHBEFA buildEmissionHBEFA(double distance, Generated.EmissionValues_HBEFA_
{
    EmissionHBEFA result      = buildEmissionHBEFABody(distance, values);
    double consumption        = buildConsumption(values.fuelConsumption, fuelType);
    result.Consumption         = roundConsumption(consumption);
    result.StandardConsumption = buildStandardConsumptionHBEFA(distance, consumption);
    result.CO2e                = buildCO2e(values);

    return result;
}
```

```
/// <param name="distance">[m]</param>
/// <param name="consumption">[l]</param>
/// <returns>1/100km</returns>
```

[Operator]

2 references

```
private double buildStandardConsumptionHBEFA(double distance, double consumption)
{
    return Math.Round((consumption / distance) * 100 * 1000, 2);
}
```

```
TextBuilder.cs* X CodeCoverage.cs CodeAnalysis.cs
C# IOSPCoverage IOSPCoverage.TextBuilder

21 }
22
23 [Integrator]
1 reference
24 public void WarningIOSP002(string modifier, object method)
25 {
26     oneLine("warning IOSP002", modifier, method, null, " and has no valid IOSP attribute");
27 }
28
29 [Integrator]
2 references
30 public void WarningIOSP003(string modifier, object method, string attribute)
31 {
32     oneLine("warning IOSP003", modifier, method, attribute, " and has no integration test");
33 }
34
35 [Integrator]
1 reference
36 public void WarningIOSP004(string modifier, object method, string attribute)
37 {
38     oneLine("warning IOSP004", modifier, method, attribute, " and has no unit test");
39 }
40
41 [Integrator]
1 reference
42 public void WarningIOSP005(string modifier, object method, string attribute)
43 {
44     oneLine("warning IOSP005", modifier, method, attribute, " and has no try..catch");
45 }
46
47 [Operator]
1 reference
48 public void ErrorIOSP006()
49 {
50     result.Add("error IOSP006: Project Units has no test project");
51 }
52
53 [Integrator]
1 reference
54 public void HintIOSP007(string modifier, object method, string attribute)
55 {
56     oneLine("hint IOSP007", modifier, method, attribute, "");
57 }
```

IOSPCoverage

```
1 reference
public void AddRoll(int pins)...
```

[Todo]
0 references

```
public Frame[] Frames() => frames.ToArray();
```

[Integrator]
0 references

```
public int TotalScore()
{
    int totalScore = 0;
    foreach (Frame frame in frames)
    {
        totalScore += frame.Score;
    }
    return totalScore;
}
```

[Integrator]
0 references

```
public int TryTotalScore()
{
    try
    {
        int totalScore = 0;
        foreach (Frame frame in frames)
        {
            totalScore += frame.Score;
        }
        return totalScore;
    }
}
```

```
D:\Code\trunk2\IOSPCoverage\IOSPCoverageConsole\bin\Debug\IOSPCoverageConsole.exe
NamespacesToIgnore Generated,Interfaces,Attributes
warning IOSP001: Program.cs Bowling.Program.Main is static and has no valid modifier (private, protected, public)
warning IOSP002: Program.cs Bowling.Game.AddRoll is public and has no valid IOSP attribute
hint IOSP007: Program.cs Bowling.Game.Frames is [Todo] public
warning IOSP005: Program.cs Bowling.Game.TotalScore is [Integrator] public and has no try..catch
warning IOSP003: Program.cs Bowling.Game.TryTotalScore is [Integrator] public and has no integration test
warning IOSP005: Program.cs Bowling.Game.Over is [Operator] public and has no try..catch
warning IOSP004: Program.cs Bowling.Game.addPinsToLastFrame is [Operator] private and has no unit test
warning IOSP004: Program.cs Bowling.Game.addPinsToLastFrameBecauseOfStrike is [Operator] private and has no unit test
warning IOSP004: Program.cs Bowling.Game.isFirstRoll is [Operator] private and has no unit test
warning IOSP004: Program.cs Bowling.Game.isLastFrameAndRoll is [Operator] private and has no unit test
The project has 9 Warnings and Errors.
```

Jenkins Build

```
16:58:16
16:58:16 d:\Jenkins\workspace\mginter2beta_trunk2_ConnectXServer_BuildTestCommit>D:
16:58:16
16:58:16 d:\Jenkins\workspace\mginter2beta_trunk2_ConnectXServer_BuildTestCommit>cd \Buildtools\IOSPCoverage
16:58:16
16:58:16 d:\Buildtools\IOSPCoverage>IOSPCoverageConsole "D:\Jenkins\workspace\mginter2beta_trunk2_ConnectXServer_BuildTestCommit\ConnectXServer.sln"
"ConnectXServer"
16:58:16 NamespacesToIgnore Generated, Interfaces, Attributes
16:58:20 warning IOSPO04: ConnectXLocate.cs Ptvag.ConnectXServer.ConnectXLocate.buildRequest is [Operator] private and has no unit test
16:58:20 hint IOSPO06: RouteResponseBuilder.cs Ptvag.ConnectXServer.RouteResponseBuilder.buildBoundingBox is [Todo] private
16:58:20 hint IOSPO06: RouteResponseBuilder.cs Ptvag.ConnectXServer.RouteResponseBuilder.buildDummies is [Todo] private
16:58:20 The project has 1 Warnings and Errors.
16:58:20
16:58:20 d:\Buildtools\IOSPCoverage>exit 1
16:58:20 Build step 'Windows Batch-Datei ausführen' marked build as failure
16:58:21 Sending e-mails to: stefan.hock@ptvgroup.com stefan.hock@ptv.de
16:58:21 Started calculate disk usage of build
16:58:21 Finished Calculation of disk usage of build in 0 seconds
16:58:21 Started calculate disk usage of workspace
16:58:21 Finished Calculation of disk usage of workspace in 0 seconds
16:58:21 Finished: FAILURE
```

32
33
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109

1 reference

public void AddRoll(int pins)...

[Todo]
0 references

public Frame[] Frames() => frames.ToArray();

[Integrator]
0 references

public int TotalScore()
{
 int totalScore = 0;
 foreach(Frame frame in frames)
 {
 totalScore += frame.Score;
 }
 return totalScore;
}

[Integrator]
0 references

public int TryTotalScore()
{
 try
 {
 int totalScore = 0;

100 %

Error List

Entire Solution 0 Errors 0 Warnings 0 Messages Build + IntelliSense Search Error List

	Code	Description	Project	File	Line	Suppression St...
!	IOSP002:	Bowling.Game.AddRoll is public and has no valid IOSP attribute		Program.cs		
i	IOSP007	Bowling.Game.Frames is [Todo] public		Program.cs		
!	IOSP005	Bowling.Game.TotalScore is [Integrator] public and has no try..catch gram.cs				
!	IOSP003	Bowling.Game.TryTotalScore is [Integrator] public and has no integration test				

➡ Presentation

- CCD Dropbox

➡ Clean Code Developer

- <http://clean-code-developer.de/die-grade/roter-grad/>

➡ Ralf Westphal

- <http://ralfw.de/2015/11/tests-als-evolutionaere-kraft/>
- <http://blog.ralfw.de/2013/04/software-fraktal-funktionale.html>

➡ PTV Group

- <http://karriere.ptvgroup.com/de/karriere/>
- <http://compass.ptvgroup.com/2016/01/ptv-supergeil/>

➡ Stefan Hock

- <https://stefanhock.wordpress.com/iosp/>



the mind of movement