



DelphiDay

italian conference

Analisi statica

Migliorare il codice con hint, lint e static analysis



LUCA MINUTI



lucominuti.it



luca.minuti@gmail.com



dev.to/lminuti



github.com/lminuti



www.linkedin.com/in/lucominuti



DelphiDay

italian conference

19-20 Giugno 2025
Piacenza



wintech
italia

OPEN-SOURCE PROJECTS

github.com/Iminuti

WiRL

github.com/delphi-blocks/WiRL

Delphi SAML

github.com/EtheaDev/Delphi-SAML

OpenSSL

github.com/Iminuti/Delphi-OpenSSL



19-20 Giugno 2024
Piacenza





AGENDA

1. Analisi statica e dinamica
2. Errori, warnings e hints
3. DelphiLint, SonarDelphi
4. PascalAnalyzer
5. GraphViz



STATICA E DINAMICA

- Analisi statica
 - Compilazione del codice
 - Code reading / Code review
 - Analisi del flusso (GraphViz)
- Analisi dinamica (a runtime)
 - Comportamento atteso vs comportamento osservato
 - Unit test, test di integrazione, penetration testing, performance, ...



PERCHÉ?

- Individuazione precoce dei bug
- Evitare problemi noti legati alla sicurezza
- Conformità agli standard
- Ottimizzazione delle prestazioni
- Valutazione delle complessità



WARNING E
HINT

1



HINT, WARNING, ERROR

- Tutti i compilatori, durante la fase di analisi del codice producono dei messaggi.
- Questi messaggi possono essere sia errori che informazioni
- I linguaggi interpretati, che non hanno una fase di compilazione, producono messaggi simili solo a runtime
- Inoltre questi linguaggi di solito sono più permissivi: debole controllo sui tipi, dichiarazione delle variabili, ecc.



ERRORI

- Sono problemi nel codice che **impediscono al compilatore di generare un programma eseguibile.**
- Sintassi errata, tipi incompatibili, funzioni mancanti, riferimenti a variabili non dichiarate.
- Il codice non si compila fino a quando tutti gli errori non vengono risolti



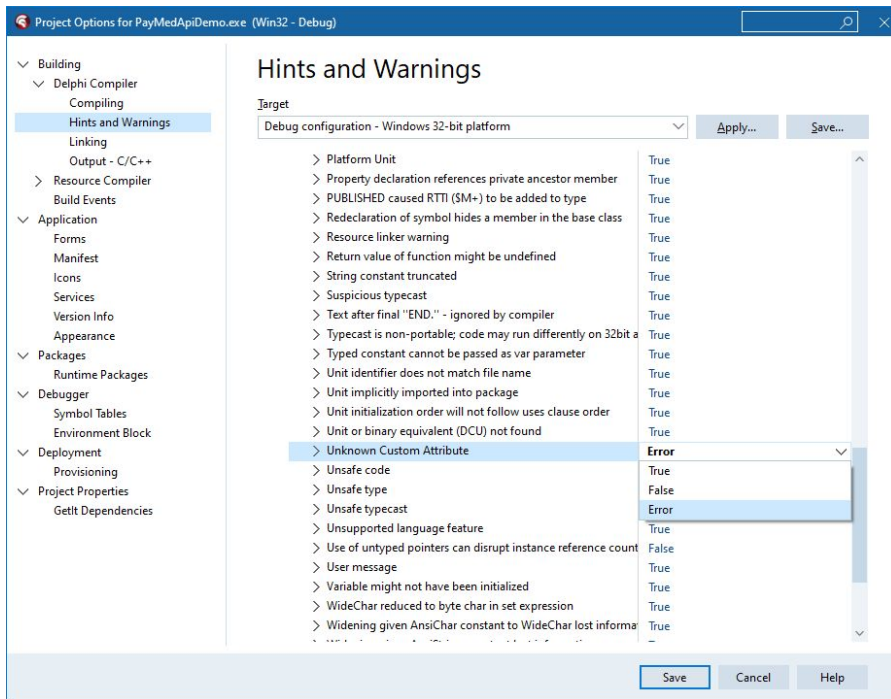
WARNING

- Indicano **potenziali problemi** che potrebbero causare comportamenti indesiderati o bug durante l'esecuzione.
- Conversioni implicite di tipi, variabili non inizializzate, possibili divisioni per zero, utilizzo di funzionalità obsolete.
- Il codice compila, ma gli avvisi dovrebbero essere **esaminati con cura e risolti**.



WARNING

- Dalle opzioni di progetto è possibile ignorare degli warning (ha senso solo per quelli relativi alla piattaforma) e quali trasformare in errore





WARNING

- W1074 Unknown Custom Attribute (Delphi)
 - Consiglio vivamente di trasformarlo in un errore
- W1057 Implicit string cast from '%s' to '%s' (Delphi)
 - Questo warning e altri simili riguardano la gestione di Unicode. In generale si possono eliminare con dei cast espliciti (da String a AnsiString e viceversa). Ma spesso sarebbe saggio gestire esplicitamente il CharSet.



HINT

- Sono **messaggi informativi** che il compilatore fornisce per migliorare il codice. Non indicano errori, ma piuttosto suggeriscono modi per ottimizzare o pulire il codice.
- Variabili dichiarate ma mai utilizzate, funzioni che potrebbero essere ottimizzate, uso di caratteristiche deprecated.
- Il **codice compila** e viene eseguito correttamente



HINT

- H2077 Value assigned to '%s' never used (Delphi)
- H2135 FOR or WHILE loop executes zero times - deleted (Delphi)
- H2164 Variable '%s' is declared but never used in '%s' (Delphi)
- H2219 Private symbol '%s' declared but never used (Delphi)
- H2235 Package '%s' does not use or export '%s.%s' (Delphi)
- H2244 Pointer expression needs no Initialize/Finalize - need ^ operator? (Delphi)
- H2365 Override method %s.%s should match case of ancestor %s.%s (Delphi)
- H2368 Visibility of property accessor method %s should match property %s.%s (Delphi)
- H2369 Property accessor %s should be %s (Delphi)
- H2384 CLS: overriding virtual method '%s.%s' visibility (%s) must match base class '%s' (%s) (Delphi)
- H2440 Inline method visibility is not lower or same visibility of accessing member '%s.%s' (Delphi)
- H2443 Inline function '%s' has not been expanded because unit '%s' is not specified in USES list (Delphi)
- H2444 Inline function '%s' has not been expanded because accessing member '%s' is inaccessible (Delphi)
- H2445 Inline function '%s' has not been expanded because its unit '%s' is specified in USES statement of IMPLEMENTATION section (Delphi)
- H2456 Inline function '%s' has not been expanded because contained unit '%s' uses compiling unit '%s' (Delphi)
- H2457 Inline function '%s' has not been expanded because contained unit '%s' uses compiling unit '%s' indirectly (Delphi)
- H2505 'Self' is uninitialized. An inherited constructor must be called before entering a try block (Delphi)
- H2509 Identifier '%s' conflicts with type parameters of container type (Delphi)
- H2596 '%s' (Delphi)



HINT

- H2077 Value assigned to '%s' never used (Delphi)
 - Spesso nasconde delle insidie o interi blocchi di codice non più usato
- H2443 Inline function '%s' has not been expanded because unit '%s' is not specified in USES list (Delphi)
 - Capita spesso aggiornando da una versione di Delphi alla successiva. Richiede l'inserimento della unit indicata

demo time





LINT

2



LINT

- Strumenti che analizzano il codice sorgente
- Errori di programmazione
- Errori stilistici
- Costrutti sospetti
- Ambiguità



DelphiLint

- Permette di integrare SonarDelphi, con l'IDE in modo da testare oltre 100 regole
- Oltre al problema segnala anche il motivo e la possibile soluzione
- Le regole sono configurabili
- Due modalità:
 - Standalone - integrato nell'IDE con regole di default
 - Connected - collegato a SonarDelphi, con regole personalizzate e la possibilità di sopprimere regole già testate



DelphiLint - Requisiti

- Microsoft Edge 79.0.309+
 - EdgeView2 SDK
- Delphi 11+
- Java 11+
- SonarDelphi (opzionale)



<https://github.com/integrated-application-development/delphlint>



DelphiLint

```
var
  LStartTime: Integer;
  C: AnsiChar; Give this short identifier a more meaningful name.
begin
  try
    Result := False;
    LStartTime := GetTickCount;
    while Integer(GetTickCount) - LStartTime < FTimeout do Avoid mixing names (found: "FTimeout" expected: "FTimeOut").
    begin
      if FComClient.CharReady then
      begin
        C := FComClient.GetChar;
        FBuffer := FBuffer + C;
        if FindChar(C, AChars) then
          Exit(True) Terminate this statement with a semicolon.
        end
      else
        Sleep(100);
    end;
  end;
```

demo time





SONARDELPHI 3



SonarQube

- Analizza il codice per più di 30 linguaggi
- Plugin per vari sistemi di CI
- SonarLint per vari IDE
- Self-managed (in house) / cloud based
- Prodotto commerciale ma esiste una “Community Edition” (set limitato di linguaggi, bugs & basic vulnerabilities)
- Static analysis, code smells, coding standards, unit tests, code coverage, code complexity, comments, bugs, and security recommendations.



<https://github.com/integrated-application-development/sonar-delphi>

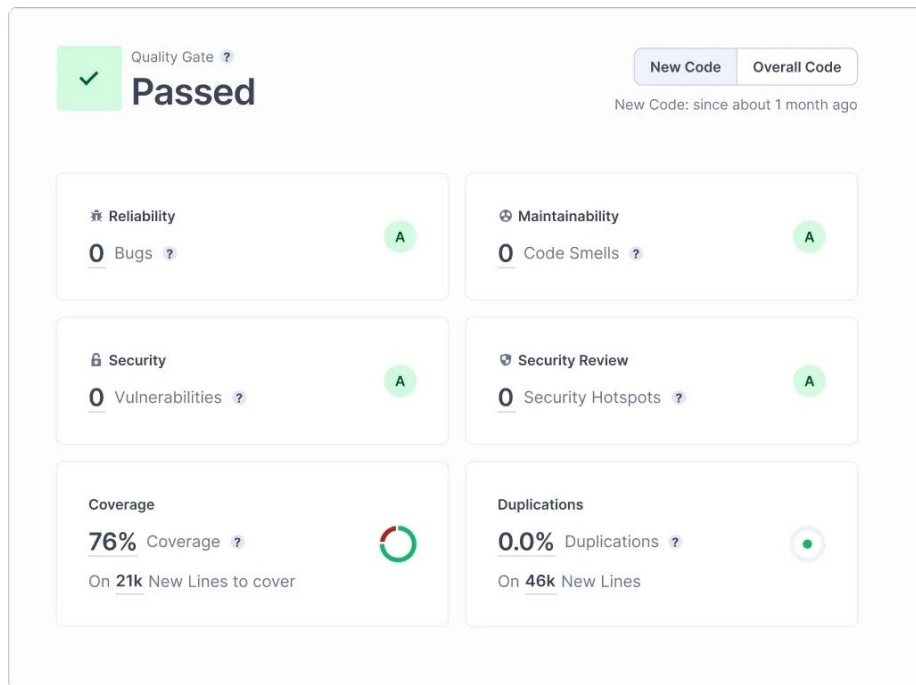


SonarQube

- **Scanner** (SonarScanner):
 - Scansiona i vari progetti ed esegue i plugin. Ogni plugin:
 - Legge i vari file
 - Calcola le metriche (e.g. lines of code, coverage, etc.)
 - Analizza il codice e applica le regole
- **Server** (SonarQube):
 - Permette di vedere e salvare i risultati della scansione ("issues" and "metrics")
 - Permette di cambiare lo stato delle issue aggiungere commenti, ecc.
 - Supporta vari plugin



SonarQube





SonarQube

sonarqube

Projects Issues Rules Quality Profiles Quality Gates Administration More

trunk

Version not provided

Overview Issues Security Hotspots Measures Code Activity

Project Settings Project Information

Issues in new code

Type

- Bug 144
- Vulnerability 0
- Code Smell 10k

Add to selection Ctrl + click

Severity

- Blocker 2.1k Minor 2.2k
- Critical 535 Info 2.3k
- Major 3k

Scope

Resolution

Status

Security Category

Bulk Change

Select issues Navigate to issue 3 / 10,199 issues 328d effort

/CSAttributes.pas

ClassNameRule

- Code Smell Open Minor Not assigned 15min effort 4 years ago

ConstructorWithoutInheritedStatementRule

- Code Smell Open Major Not assigned 5min effort 1 year ago

/CSConfig.pas

GetIniFileName - unused function/procedure. No other function and procedure in a project refers to it. Probably you could remove it. No tags

- Code Smell Open Info Not assigned 15min effort 7 years ago

Read - unused function/procedure. No other function and procedure in a project refers to it. Probably you could remove it. No tags

- Code Smell Open Info Not assigned 15min effort 7 years ago

Write - unused function/procedure. No other function and procedure in a project refers to it. Probably you could remove it. No tags

- Code Smell Open Info Not assigned 15min effort 7 years ago



SonarQube

sonarqube

Projects Issues Rules Quality Profiles Quality Gates Administration More Q

☆ [redacted] / trunk [green checkmark] [dropdown arrow]

Version not provided [lock icon]

Overview **Issues** Security Hotspots Measures Code Activity

Project Settings ▾ [Project Information](#)

3 / 10,199 issues

CarSharing/CSConfig.pas

GetIniFileName - unused function/procedure. No other function and procedure in a project refers to it. Probably you could remove it.

Code Smell

Read - unused function/procedure. No other function and procedure in a project refers to it. Probably you could remove it.

Code Smell

Write - unused function/procedure. No other function and procedure in a project refers to it. Probably you could remove it.

Code Smell

ConstructorWithoutInheritedStatementRule

Code Smell

CarSharing/CSConsole.pas

FieldNameRule

Code Smell

GetIniFileName - unused function/procedure. No other function and procedure in a project refers to it. Probably you could remove it. [delphi:UnusedFunctionRule](#)

No tags +

Code Smell ▾ Open ▾ Info ▾ Not assigned ▾

15min effort • 7 years ago

Where is the issue? Why is this an issue? Activity

[redacted] > [redacted]/CSConfig.pas [See all issues in this file](#) [dropdown arrow]

```
13  ininuti      FParamNameSuffix: string;
14              FParamPort: Integer;
15              FPort: Integer;
16              FLogRequest: Boolean;
17  ininuti      FNameSuffix: string;
18              function GetIniFileName :string;

ininuti

19
20
21  procedure ReadParams;
public
  procedure Read;
```

GetIniFileName - unused function/procedure. No other function and procedure in a project refers to it. Probably you could remove it.

Read - unused function/procedure. No other function and procedure in a project refers to it. Probably you could remove it.



SonarQube

sonarqube

Projects Issues Rules Quality Profiles Quality Gates Administration More

Bicta / trunk

Version not provided

Overview Issues Security Hotspots Measures Code Activity

Project Settings Project Information

ConstructorWithoutInheritedStatementRule

Code Smell

CarSharing/CSConfig.pas

GetIniFileName - unused function/procedure. No other function and procedure in a project refers to it. Probably you could remove it.

Code Smell

Read - unused function/procedure. No other function and procedure in a project refers to it. Probably you could remove it.

Code Smell

Write - unused function/procedure. No other function and procedure in a project refers to it. Probably you could remove it.

Code Smell

ConstructorWithoutInheritedStatementRule

Constructor without 'inherited' statement. [delphi:ConstructorWithoutInheritedStatementRule](#)

No tags

Code Smell Open Major Not assigned

5min effort • 2 years ago

Where is the issue? Why is this an issue? Activity

Bicta > CarSharing/CSAttributes.pas

See all issues in this file

```
18 1minuti
19
20 implementation
21 { RemotableAttribute }
22
23 constructor RemotableAttribute.Create(AHttpMethods: THttpMethods);
24
25 begin
26     FHttpMethods := AHttpMethods;
27 end;
```

ConstructorWithoutInheritedStatementRule



SonarQube

sonarqube

Projects Issues Rules Quality Profiles Quality Gates Administration More

Bulk Change

1 / 3,005 rules

Filters

Search for rules...

Language

Search for languages...

Java 626

C# 424

TypeScript 331

JavaScript 324

PHP 253

Python 242

VB.NET 196

Kotlin 126

Flex 76

HTML 65

Terraform 51

Delphi 43

Ruby 42

Scala 41

Go 38

15 shown [Show More](#)

Type

Bug 620

Code Smell 423

Table

1 / 3,005 rules

"important" should not be used on "keyframes" CSS Bug

"\$this" should not be used in a static context PHP Bug

"&&" and "||" should be used PHP Code Smell suspicious

".equals()" should not be used to test the values of "Atomic" classes Java Bug multi-threading

"<IDOCETYPE>" declarations should appear before "<html>" tags HTML Bug user-experience

"<>" should not be used to test inequality Python Code Smell obsolete

"<?php" and "<?=" tags should be used PHP Code Smell convention, psr1

"<fieldset>" tags should contain a "<legend>" HTML Bug accessibility

"<frames>" should have a "title" attribute HTML Bug accessibility

"<html>" element should have a language attribute HTML Bug accessibility, wcag2-a

"" and "<dt>" item tags should be in "", "" or "<dl>" container tags HTML Bug

"<object>" tags should provide an alternative content HTML Code Smell accessibility, wcag2-a

"" and "" tags should be used DEPRECATED HTML Bug

"<table>" tags should have a description HTML Bug accessibility, wcag2-a

demo time





PascalAnalyzer

4



PASCAL ANALYZER

- Commerciale (c'è una versione Lite), prima versione del 2001
- Scansiona il codice alla ricerca di potenziali errori
- C'è un help che spiega il senso delle anomalie rilevate
- Produce decine di tipologie di report, per esempio:
 - Warning: potenziali problemi
 - Performance: segnala situazioni che possono degradare le performance
 - Dipendenza tra classi e componenti terze parti
 - Memoria: aiuta ad identificare memory leak (bilanciamento create/free)



Pascal Analyzer

```
// STWA4 - Index Error

function GetResult: Integer;
var
  Arr: array[0.. 10] of Integer;
  Idx: Integer;
begin
  if Random(3) = 1 then
    Idx := 1;
  else
    Idx := 11;

    if Arr[Idx] = 2 then
      begin
        ...
      end;
      ...
    end;
end;
```



Pascal Analyzer

```
// STWA5 - Possibile bad pointer usage
```

```
function GetObj: TMyClass;  
begin  
    if SomeCondition then  
        Result := TMyClass.Create  
    else  
        Result := nil;  
end;
```

```
procedure MainProc;  
var  
    Obj: TMyClass;  
begin  
    Obj := GetObj;  
    try  
        Obj.MyMethod;  
    finally  
        Obj.Free;  
    end;  
end;
```



Pascal Analyzer

```
// STWA7 - For loop with possibile bad-condition
```

```
procedure MainProc;  
var  
    I: Integer;  
begin  
    for I := 0 to List.Count do  
        begin  
            DoSomething(List[I]);  
        end;  
    end;
```



Pascal Analyzer

```
// STWA9 - Var parameter that are used but never set

function CalcNetResult (var InitVal: Integer): Integer;
var
    I, Sum: Integer;
begin
    Sum := InitVal;
    for I := 1 to 2 do
        Inc(Sum, GetPartAmount(I));
    Result := Sum;
end;
```



Pascal Analyzer

```
// WARN15 - Constructor/destructor without call to in inherited
```

```
TBaseClass = class
public
    constructor Create; virtual;
end;

TChildClass = class(TBaseClass)
public
    constructor Create; override;
end;

constructor TChildClass.Create;
begin
    //
end;
```



Pascal Analyzer

```
// WARN30 - Short-circuited if/case statements
```

```
function IsValid(Age: Integer; Check: Boolean): Boolean;  
begin  
    Result := False;  
    if Check then  
        begin  
            if Age >= 18 then;  
            Result := True;  
        end;  
    end;
```




Pascal Analyzer

```
// MEMO1 - Local object with unprotected calls to free
```

```
procedure CalcResult;  
var  
    L: TList;  
begin  
    L := TList.Create;  
    ...  
    L.Free;  
end;
```



Pascal Analyzer

```
// REDU4 - Local identifiers that are set and referenced once
```

```
function GetResult: string;  
var  
    Idx: Integer;  
begin  
    Result := '';  
    Idx := Random(3);  
  
    if Idx = 3 then  
        Result := 'Ok';  
end;
```



PASCAL ANALYZER

Start ☒ | Status.txt ☒ | Identifiers.txt ☒ | Duplicate Identifiers.txt ☒ | Third-party Dependencies.txt ☒ | Most Called.txt ☒

```
*****
*                               *
*      Most Called Report for   *
*      C:\Corsi\ModernAppDemo\Anonimizer\Anonymizer.dpr                *
*      03/06/2024 16:22:35                                           *
*****
```

These subprograms are called from two or more locations (7, was unknown):

```
-----
```

13	ServiceLocator	Func, Interfaced	Demo.Core.ServiceLocator (45)
8	RegisterClass	Proc, Method	Demo.Core.ServiceLocator\TServiceLocator (31)
6	Create	Constructor	Demo.Core.Rtti\AliasAttribute (14)
3	GetClass<T>	Func, Method	Demo.Core.ServiceLocator\TServiceLocator (26)
2	CreateInstance	Func, Method	Demo.Core.Rtti\TRttiUtils (35)
2	ShowHelp	Proc, Method	Anon.Core.Console\TConsole (15)
2	TryGetClass<T>	Func, Method	Demo.Core.ServiceLocator\TServiceLocator (30)

demo time





GraphViz

5



GraphViz

- Con Delphi 12 è stato introdotto il supporto a GraphViz
- GraphViz permette di creare grafici di dipendenze tra elementi
- È possibile visualizzare (anche con tool esterni) le dipendenze tra le unit
- In particolare le dipendenze circolari che rallentano molto la fase di compilazione
- Tramite il compilatore a linea di comando è possibile creare un file gv

```
dcc32 --graphviz --graphviz-exclude=System.*;Vcl.*;WinApi.* GraphTest.dpr
```

https://docwiki.embarcadero.com/RADStudio/Athens/en/What%27s_New#Delphi_Exporting_the_uses_Graph_in_a_GraphViz_file



GraphViz

```
digraph Anonymizer {  
    "Anon.Connection.FireDAC" -> "Anon.Interfaces.Database"  
    "Anon.Connection.Oracle" -> "Anon.Interfaces.Database"  
    "Anon.Interfaces.Anonymizer"  
    "Anon.Interfaces.Database"  
    "Anon.Core.Anonymizer" -> "Anon.Interfaces.Anonymizer"  
    "Anon.Core.Anonymizer" -> { "Anon.Generators" "Anon.Core.Utills" } [arrowhead=open,style=dashed]  
    "Anon.Generators" -> "Anon.Interfaces.Anonymizer"  
    "Anon.Core.Utills" -> "Anon.Interfaces.Anonymizer"  
    "Anon.Core.Console" -> "Anon.Interfaces.Anonymizer"  
    "Anon.Generators.Date" -> "Anon.Interfaces.Anonymizer"  
    "Anon.Generators.FirstName" -> "Anon.Interfaces.Anonymizer"  
    "Anon.Generators.LastName" -> "Anon.Interfaces.Anonymizer"  
}
```


demo time





THANK YOU