



Eleve o nível de sua arquitetura com Delphi e AOP

Francisco Thiago de Almeida

Embarcadero

Conference

Oi! Eu sou o FT!



- Desenvolvedor há 14 anos
- Formado em Teologia (UMESP)
- Estudante de SI (UFMS)
- Pós-graduando em Arquiteturas de Soluções na Plataforma .Net

Oi! Eu sou o FT!



- Desenvolvedor há 14 anos
- Formado em Teologia (UMESP)
- Estudante de SI (UFMS)
- Pós-graduando em Arquiteturas de Soluções na Plataforma .Net

Oi! Eu sou o FT!



- Sistemas de Indústria
- Comércio
- Transporte de cargas
- Gestão empresarial
- Justiça
- Web, Client/Server, Multitier

Oi! Eu sou o FT!



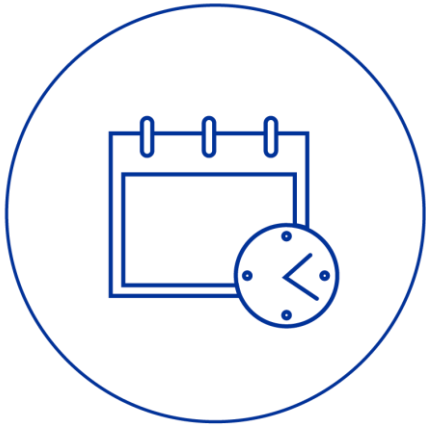
- Atualmente integro o setor de IT Services, da DB1 Global Software

DB1
IT SERVICES



DB1
global software

Qual a nossa agenda?



- Por que AOP?
- Como funciona a AOP?
- Como ela seria implementável no Delphi?
- Criando aspectos
- Limitações;
- Boas práticas!

Por que AOP?

Embarcadero

Conference

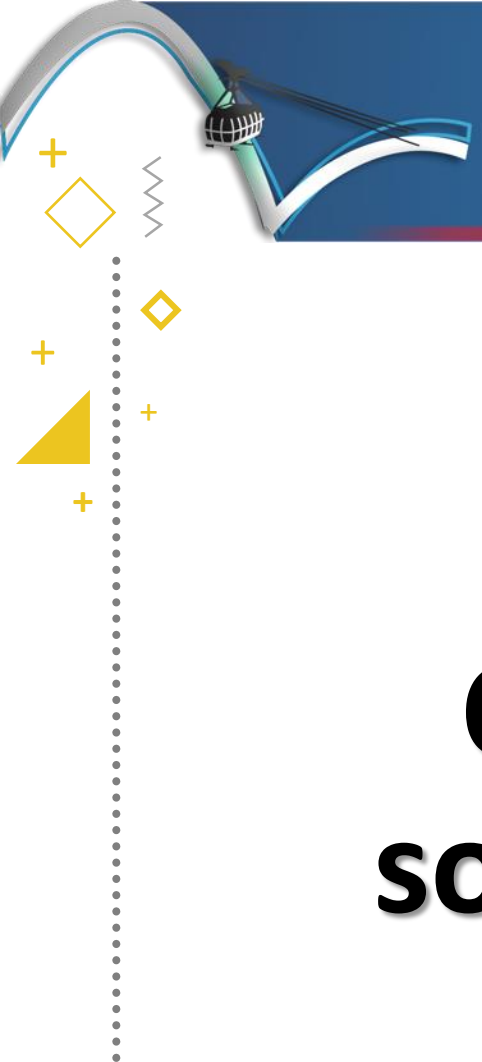
Por que AOP?



- O mundo mudou...
 - Requisitos básicos;
 - Requisitos para além do avançado;
- Sistemas inteligentes

Por que AOP?

Como a arquitetura do nosso software irá suportar tudo isso?



Por que AOP?



- Requisitos:
 - Gerar log de todos os Form acessados na aplicação;
 - Armazenar nome do Form;
 - Armazenar usuário;
 - Armazenar permissões;
 - Armazenar período;

Por que AOP?

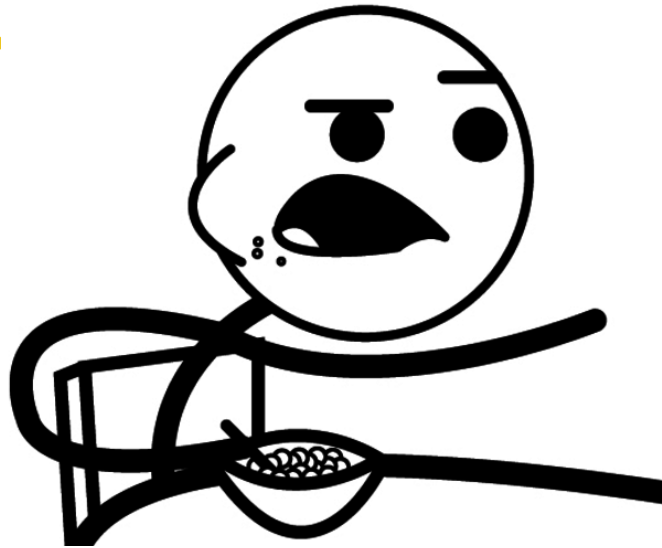
- Requisitos:

- Gerar log de todos os Form acessados na aplicação;
- Armazenar nome do Form;
- Armazenar usuário;
- Armazenar permissões;
- Armazenar período;

- Sem quebrar o SRP.

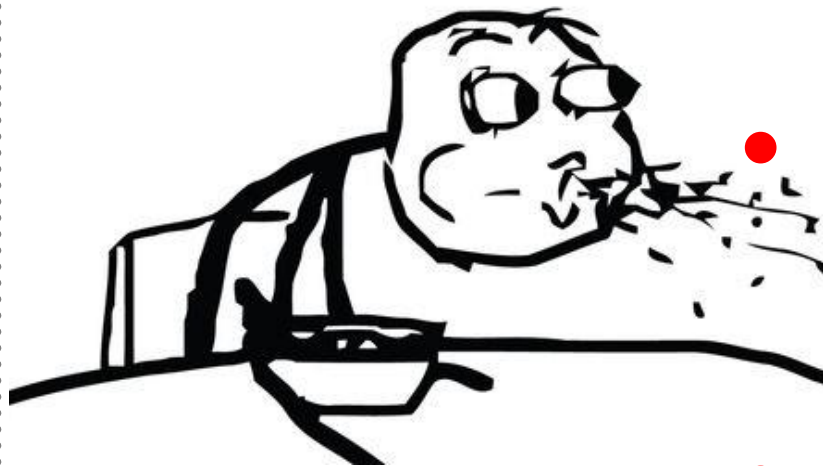


Por que AOP?



- Mapearia todos os botões/menus de abertura de formulários?
- Alterar a Fábrica de forms?
- Alterar a herança?

Por que AOP?



- E se tudo mudar?
- Fazer autorização do usuário?
- Fazer log do tempo gasto na execução das rotinas?
- ...

Por que AOP?

Percebe que em todos esses cenários, você estaria violando o SRP?



Por que AOP?

- Quais seriam as responsabilidades de um form?
- Apresentar os dados para os usuários;
- Oferecer elementos de interface para que o usuário possa interagir com os dados;

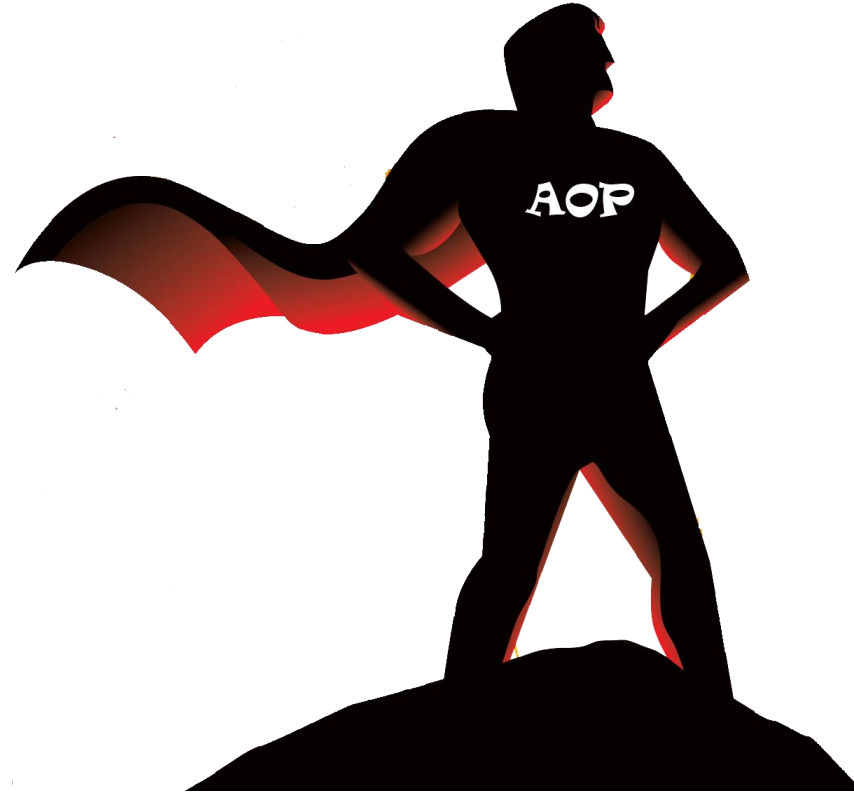


Por que AOP?



- Requisitos funcionais;
 - Propósito
- Requisitos não-funcionais/sistêmicos;
 - Acessório;

Por que AOP?



AOP – Aspect Oriented Programming

Por que AOP?



AOP te permite adicionar ao objeto os seus requisitos não funcionais sem adicionar dependência ou codificação. E isso de forma transparente!

Como funciona a AOP?

Embarcadero

Conference

Como funciona a AOP?



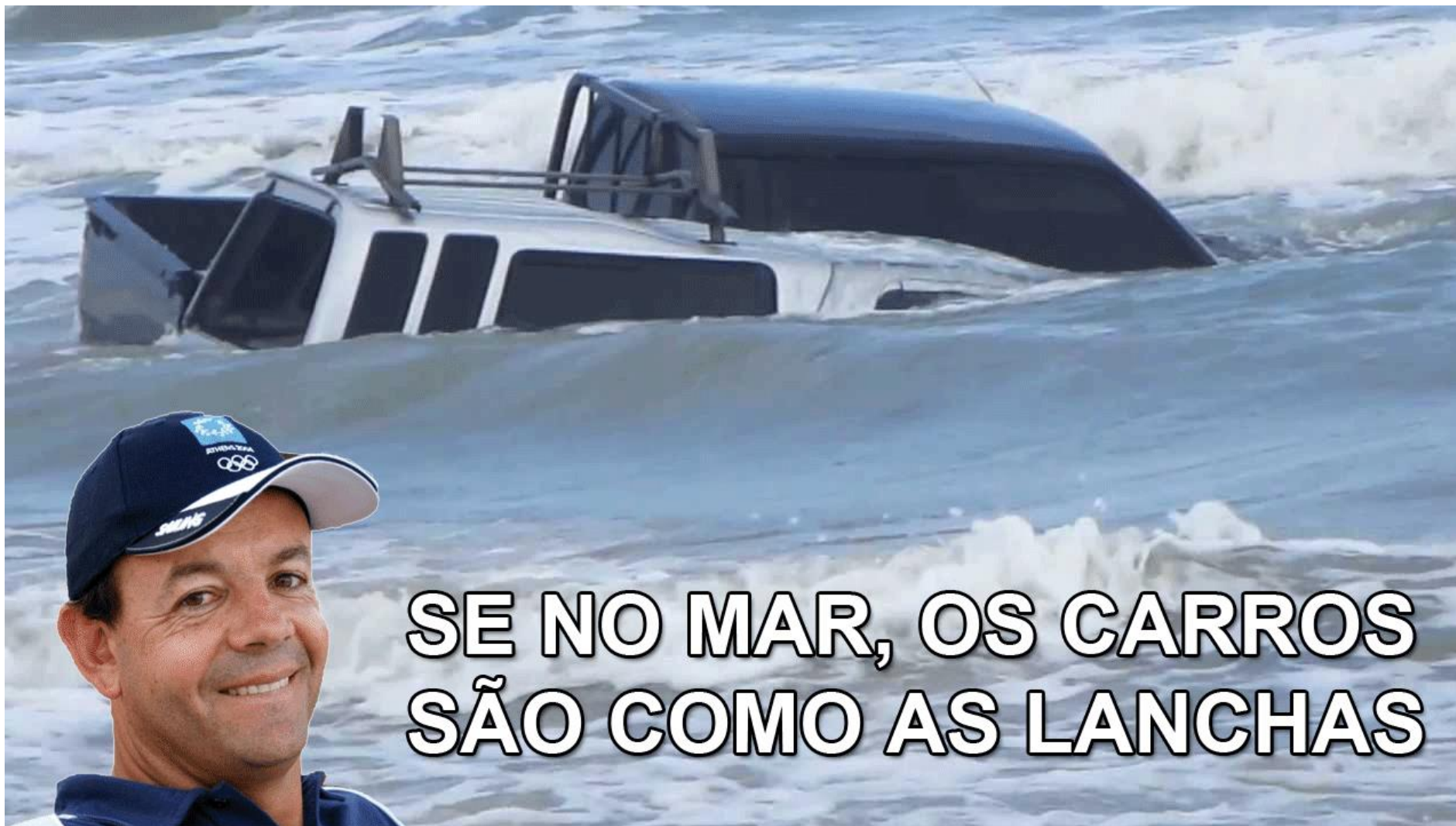
- Requisitos funcionais ou linguagem de componente (LC);
- Requisitos não funcionais ou *Crosscutting Elements*;
- *Aspect* encapsula os *CE*;
- *Join Point*: Onde *LC* e *CE* se cruzam;

Como funciona a AOP?



- *Point Cut*: Quais comportamentos o Aspect deve aderir;
- *Advice*: Código a ser executado antes, durante, depois e em caso de erro;

Como funciona a AOP?



SE NO MAR, OS CARROS
SÃO COMO AS LANCHAS

Como funciona a AOP?



- *Aspect*: São as classes
- *Join Point*: São os comportamentos alvo da classe;
- *Point Cut*: Como o será feita a aderência (Custom Attributes, Name?);

Como funciona a AOP?



- *Advice: O método, na classe de aspecto, que executa antes, durante, depois e em caso de erro;*

Como funciona a AOP?



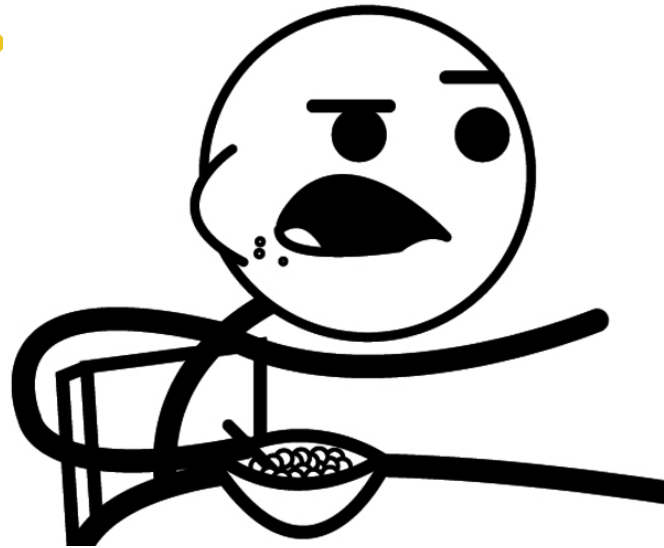
Neste ponto, surge o *Weaver*.
Ele mescla o código compilado de
ambos os componentes, gerando um
único *bytecode*.

Bytecode?!

Como funciona a AOP?



Como funciona a AOP?



Mas eu posso ter
tudo isso em
Delphi?

Como AOP seria
implementável em
Delphi?

Como AOP seria implementável em Delphi?



- TVirtualMethodInterceptor
 - Mão na massa

type

TClasseTeste = **class**

private

procedure MetodoVirtualPrivado; **virtual**;

protected

procedure MetodoVirtualProtegido; **virtual**;

published

procedure MetodoVirtualPublico; **virtual**;

procedure MetodoNaoVirtual;

end;

TClasseTesteFilha = **class**(TClasseTeste)

public

procedure MetodoVirtualProtegido; **override**;

end;

Como AOP seria implementável em Delphi?

```
var
  _classeTeste: TClasseTeste;
  _vmt: TVirtualMethodInterceptor;
begin
  try
    Writeln('Teste');

    _classeTeste := TClasseTesteFilha.Create;
    _vmt := TVirtualMethodInterceptor.Create(_classeTeste.ClassType);
```

Como AOP seria implementável em Delphi?

```
_vmt.OnBefore :=  
_  procedure(Instance: TObject; Method: TRttiMethod; const Args: TArray<TValue>;  
    out DoInvoke: Boolean; out Result: TValue)  
  begin  
    Writeln(Format('>>>>>Antes de %s.%s', [Instance.ClassName, Method.Name]));  
  end;  
  
_vmt.OnAfter :=  
_  procedure(Instance: TObject; Method: TRttiMethod; const Args: TArray<TValue>;  
    var Result: TValue)  
  begin  
    Writeln(Format('<<<<<Depois de %s.%s', [Instance.ClassName, Method.Name]));  
  end;  
  
_vmt.Proxify(_classeTeste);  
_classeTeste.MetodoNaoVirtual;
```

Como AOP seria implementável em Delphi?

```
_vmt.OnBefore :=  
_  procedure (Instance: TObject; Method: TRttiMethod; const Args: TArray<TValue>;  
    out DoInvoke: Boolean; out Result: TValue)  
  begin  
    Writeln(Format('>>>>>Antes de %s.%s', [Instance.ClassName, Method.Name]));  
  end;  
  
_vmt.OnAfter :=  
_  procedure (Instance: TObject; Method: TRttiMethod; const Args: TArray<TValue>;  
    var Result: TValue)  
  begin  
    Writeln(Format('<<<<<Depois de %s.%s', [Instance.ClassName, Method.Name]));  
  end;  
  
_vmt.Proxify(_classeTeste);  
_classeTeste.MetodoNaoVirtual.
```


Como AOP seria implementável em Delphi?

```
_vmt.OnBefore :=  
_  procedure(Instance: TObject; Method: TRttiMethod; const Args: TArray<TValue>;  
    out DoInvoke: Boolean; out Result: TValue)  
  begin  
    Writeln(Format('>>>>>Antes de %s.%s', [Instance.ClassName, Method.Name]));  
  end;  
  
_vmt.OnAfter :=  
_  procedure(Instance: TObject; Method: TRttiMethod; const Args: TArray<TValue>;  
    var Result: TValue)  
  begin  
    Writeln(Format('<<<<<Depois de %s.%s', [Instance.ClassName, Method.Name]));  
  end;  
  
_vmt.Proxify(_classeTeste);  
_classeTeste.MetodoNaoVirtual.
```

Como AOP seria implementável em Delphi?

```
_vmt.OnBefore :=  
_  procedure(Instance: TObject; Method: TRttiMethod; const Args: TArray<TValue>;  
    out DoInvoke: Boolean; out Result: TValue)  
  begin  
    Writeln(Format('>>>>>Antes de %s.%s', [Instance.ClassName, Method.Name]));  
  end;  
  
_vmt.OnAfter :=  
_  procedure(Instance: TObject; Method: TRttiMethod; const Args: TArray<TValue>;  
    var Result: TValue)  
  begin  
    Writeln(Format('<<<<<Depois de %s.%s', [Instance.ClassName, Method.Name]));  
  end;  
  
_vmt.Proxify(_classeTeste);  
_classeTeste.MetodoNaoVirtual:
```

Como AOP seria implementável em Delphi?

Pai

A VMI permite que você execute um código antes, depois e em caso de exceção, de métodos virtuais

17:26

Mãe

Os Advices são métodos executados antes, durante, depois e em caso de erro em um Join Point

17:36

Caramba!

17:37 ✓

1 MENSAGEM NÃO LIDA

Mãe

Percebe, Ivair, a semelhança dos conceitos?

Como AOP seria implementável em Delphi?

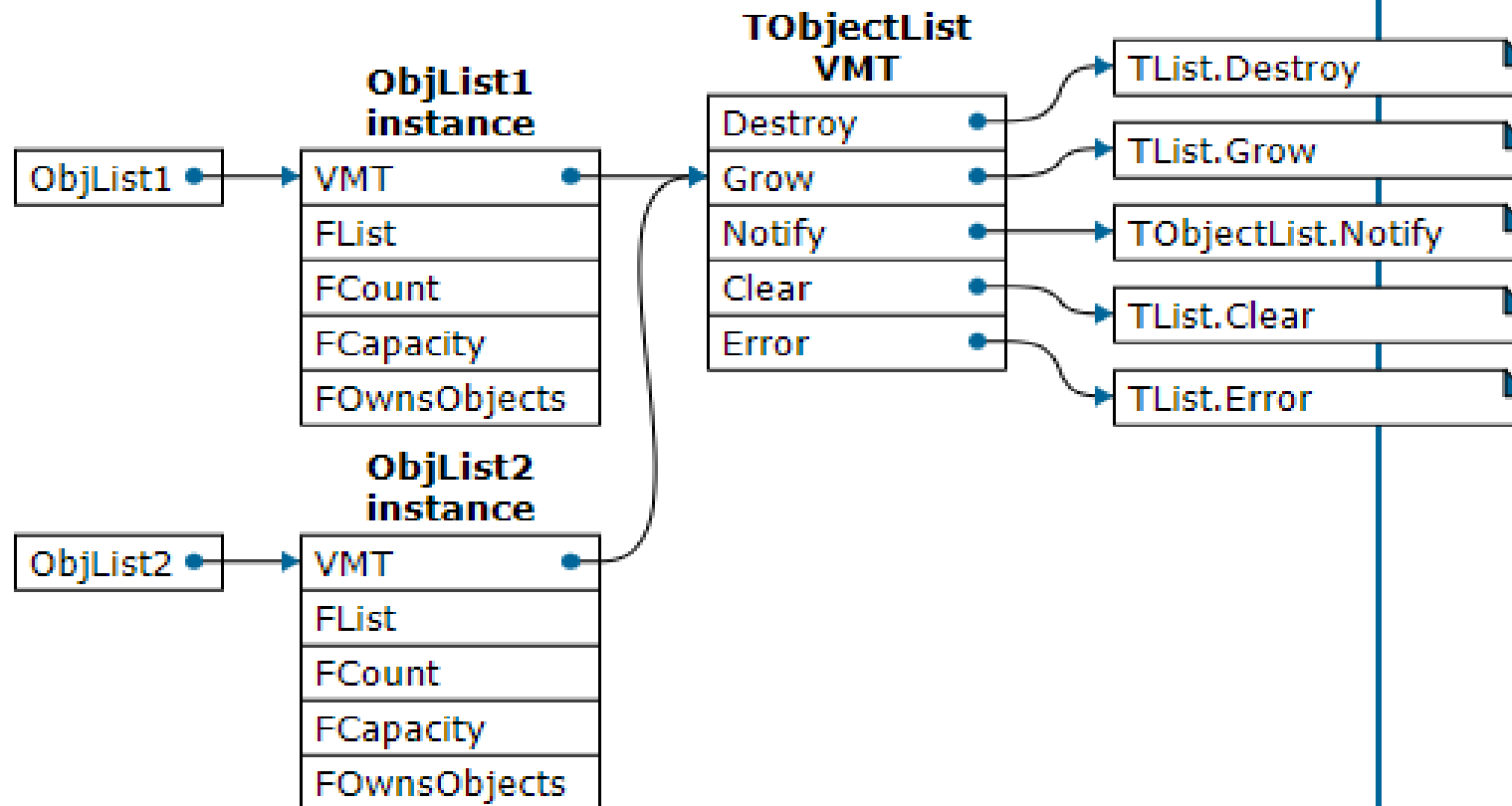


- TVirtualMethodInterceptor
 - Mão na massa
- Como funcionam os métodos virtuais?

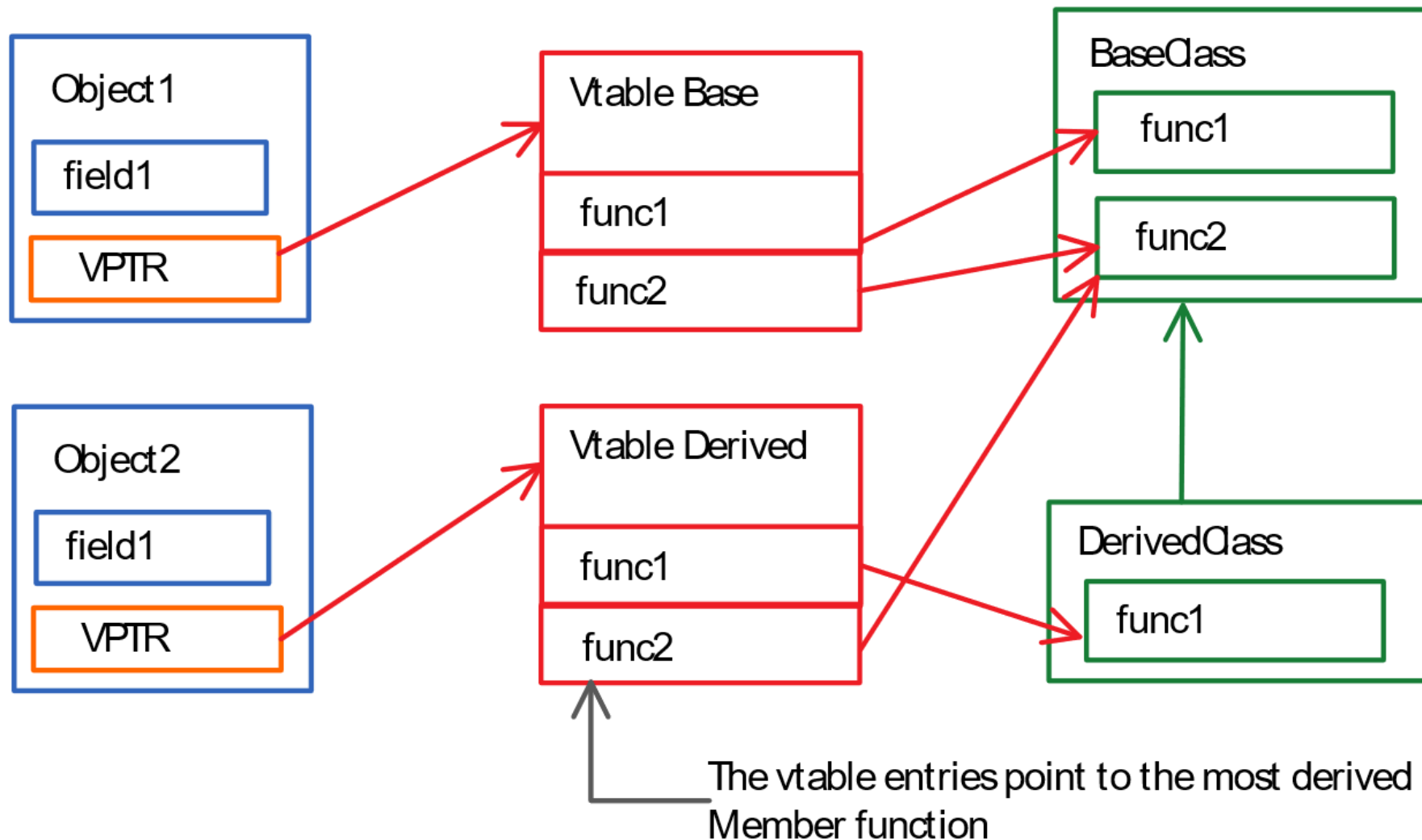
Como AOP seria implementável em Delphi?

Figure 1

Illustration of the object and vmt layouts resulting from the code of Listing 1. Note that both `TObjectList` instances point to the *same* VMT.



Como AOP seria implementável em Delphi?

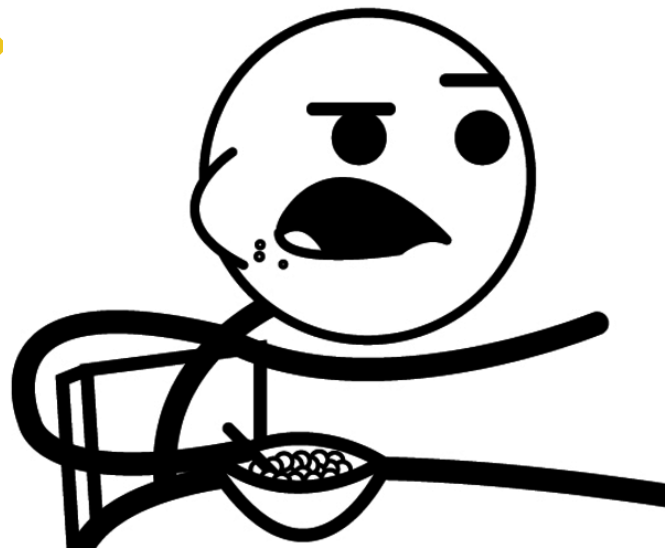


Como AOP seria implementável em Delphi?



- TVirtualMethodInterceptor
- Como funcionam os métodos virtuais?
- Custo de processamento?

Como AOP seria implementável em Delphi?



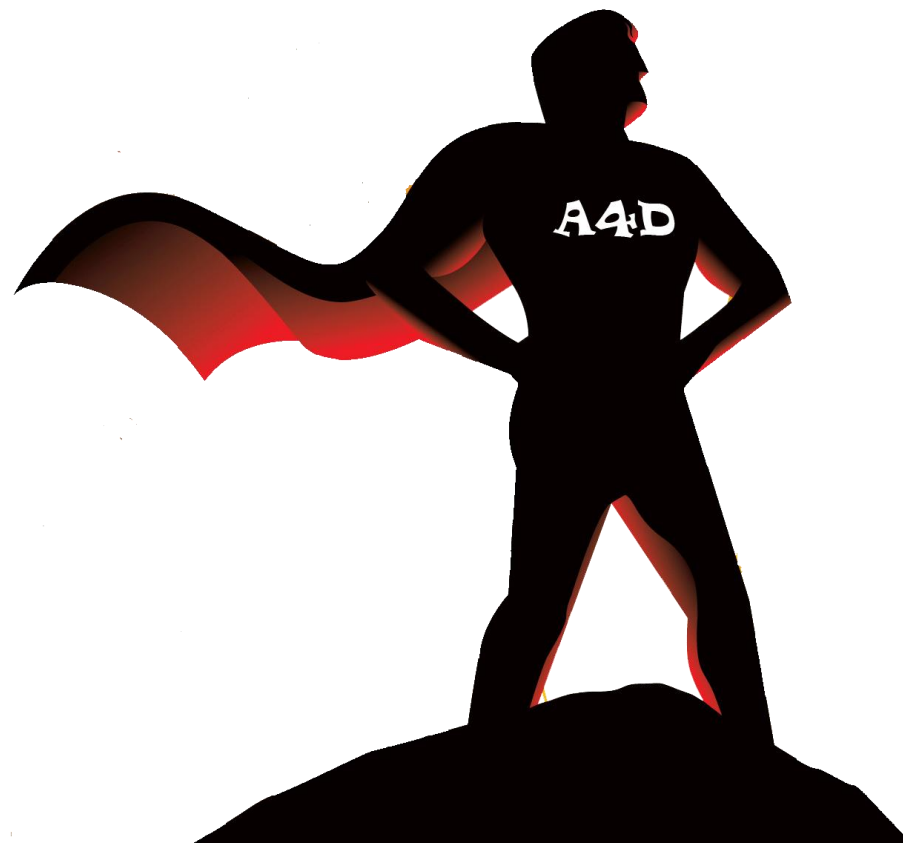
Mas eu vou ter
que implementar
tudo **do zero?!**

Como AOP seria implementável em Delphi?



- Tentar fazer o seu pode ser legal
- Já existem opções prontas
 - Spring4D
 - Dsharp

Como AOP seria implementável em Delphi?

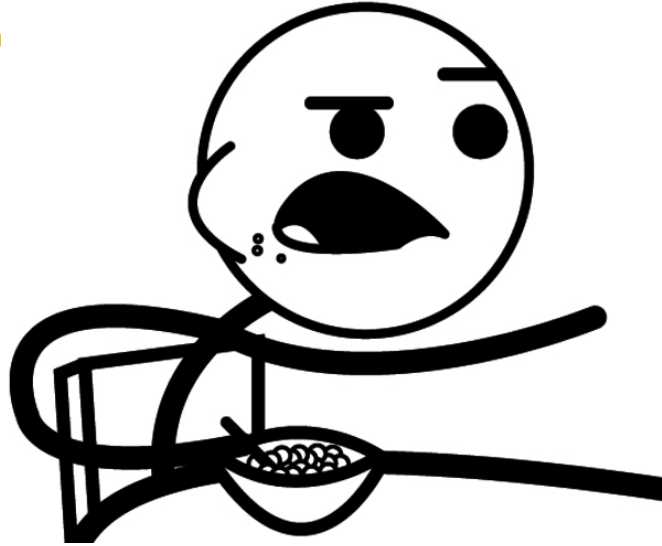


Aspect4Delphi – Ezequiel Juliano
(tem no GitHub)

Criando o primeiro aspecto

Controle de transação
(Mão na massa)

Criando o primeiro aspecto



- Muito código a mais;
- Está poluindo o form;
- Tem como esconder isso aí?

Aspectos nos próprios objetos

Controle de acesso
(Mão na massa)

Embarcadero

Conference

Aspectos nos próprios objetos



- Você pode alterar o código dos objetos, incluindo o aspecto na sua criação;
- Quando não tiver o código, utilize herança;

Aspectos nos próprios objetos



Você vai precisar de:

- Um *Singleton* para acessar o contexto dos aspectos;
- Escrever a classe do aspecto;
- Sobrescrever o construtor e destrutor da classe;
- Definir os *Join Points*

Limitações

Embarcadero

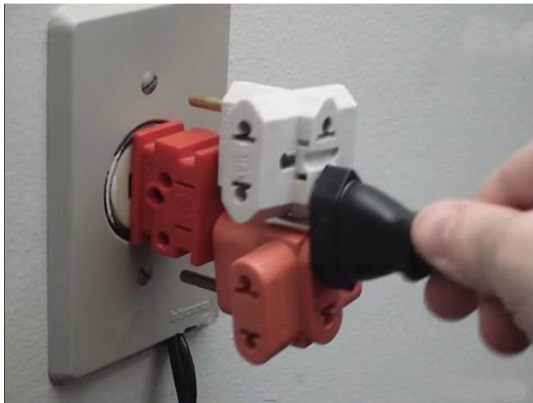
Conference

Limitações



- Se você deseja utilizar CustomAttributes, há uma limitação derivada do próprio princípio da VMI.

Limitações - Workaround



- Criar um método virtual na classe nova e garantir sua chamada no *Join Point* desejado;
- Criar um aspecto filho, cuja especialidade seja para aquela classe/método;
- Alterar o framework (meu preferido)

Boas práticas

Para sistemas novos ou legados

Embarcadero

Conference

Boas práticas

- Utilize Fábricas
- Utilize frameworks de inversão de controle;
- Use testes automatizados;



Boas práticas

- Deixe os aspectos simples;
- Utilize exceções específicas



Boas práticas

- Estude os frameworks
- Baixe o Aspect4Delphi e estude!
- <https://github.com/ezequieljuliano/Aspect4Delphi>



OBRIGADO



francisco.almeida@db1.com.br

github.com/ftathiago

blogdoft.blogspot.com

blog.db1.com.br

www.linkedin.com/in/ftathiago

Embarcadero

Conference





Embarcadero Conference