



Fabrício Cabral <fabriciofx@gmail.com>

[UML-BR] polimorfismo em C

39 messages

Marcelo Elias Del Valle <mvallebr@gmail.com>

Sat, Dec 11, 2010 at 9:32 PM

Reply-To: UML-BR@yahoogrupos.com.br

To: UML-BR@yahoogrupos.com.br

Caros,

Recebi um link esses dias que achei interessante, sobre polimorfismo em C, e resolvi compartilhar com o grupo.

<http://www.thradams.com/codeblog/cpoly.htm>

Há tempos atrás eu tinha gerado uma polêmica sobre isso aqui no grupo, mas poucos devem se lembrar. Contudo, continuo acreditando em orientação a objetos mesmo em linguagens procedurais, como C.

Claro que uma linguagem de mais alto nível tornaria mais fácil a separação de preocupações, mas esse link é um exemplo de que é possível aplicar os mesmos conceitos numa linguagem como C.

Abraços,

—

Marcelo Elias Del Valle

11 6995 0065

[As partes desta mensagem que não continham texto foram removidas]

—*—*—*—

| [através de email](#) | [Responder através da web](#) | [Adicionar um novo tópico](#)
Mensagens neste tópico (1)

ATIVIDADE NOS ÚLTIMOS DIAS: [Novos usuários 2](#) |
[Visite seu Grupo](#)

Está solteiro? Hoje pode ser seu último dia, crie seu perfil!

Hotels.com - Reveillon em Buenos Aires a partir de R\$ 99

Mais de 200 mil Vagas de EMPREGO, em 1.600 cidades brasileiras.

YAHOO! GRUPOS
BRASIL

Trocar para: [SÃ³ Texto](#), [Resenha DiÃ¡ria](#) • [Sair do grupo](#) • [Termos de uso](#)

—*—*—*—

Marcio Tierno <mtierno@rocketmail.com>

Sun, Dec 12, 2010 at 10:04 AM

Reply-To: UML-BR@yahoogrupos.com.br

To: UML-BR@yahoogrupos.com.br

Marcelo

Eu lembro! Hehehehehehe!!!

Trata-se de um programa bem "safadinho" esse que vc postou o link.

Em primeiro lugar, é preciso dizer que C tem polimorfismo por sobrecarga de operador (overload), que é um polimorfismo estático.

O programinha cujo link vc postou *EMULA* polimorfismo universal por inclusão, mas só emula. Ele faz uso de estruturas de baixo nível nativas de C, especificamente a habilidade de um ponteiro apontar para qualquer endereço de memória - o que inclui o endereço de uma função. Como não há polimorfismo por inclusão em C, o artifício (bem esperto, diga-se) é o uso de struts, que são um tipo abstrato de dado. Struts não têm polimorfismo, mas ele criou uma strut com dois campos ponteiros. E quando instancia um objeto Circle ou Box, esse "construtor" faz os campos da strut apontarem para os endereços correspondentes das funções para Circle ou para Box.

Espertinho. Mas não é polimorfismo universal por inclusão, pois não é simples incluir uma nova classe ou um novo método nesse modelo.

[]s

MT.

--- Em UML-BR@yahoogrupos.com.br, Marcelo Elias Del Valle <mvallebr@...> escreveu

>

> Caros,

>

> Recebi um link esses dias que achei interessante, sobre polimorfismo em

> C, e resolvi compartilhar com o grupo.

> <http://www.thradams.com/codeblog/cpoly.htm>

>

> Há tempos atrás eu tinha gerado uma polêmica sobre isso aqui no grupo,

> mas poucos devem se lembrar. Contudo, continuo acreditando em orientação a

> objetos mesmo em linguagens procedurais, como C.

> Claro que uma linguagem de mais alto nível tornaria mais fácil a

> separação de preocupações, mas esse link é um exemplo de que é possível

> aplicar os mesmos conceitos numa linguagem como C.

>

> Abraços,

> --

> Marcelo Elias Del Valle

> 11 6995 0065

>

>

> [As partes desta mensagem que não continham texto foram removidas]

>

_____'_'____

| [através de email](#) | [Responder através da web](#) | [Adicionar um novo tópico](#)
Mensagens neste tópico (2)

ATIVIDADE NOS ÚLTIMOS DIAS: [Novos usuários 2](#) |
[Visite seu Grupo](#)

YAHOO! GRUPOS
BRASIL

Trocar para: [SÃ Text](#), [Resenha Diãria](#) • [Sair do grupo](#) • [Termos de uso](#)

_____'_'____

Fabricao Cabral <fabriciofx@gmail.com>
To: UML-BR@yahoogrupos.com.br

Mon, Dec 13, 2010 at 11:46 AM

Olá todos,

concordo com o MT. Aliás, esta era uma opinião errada que eu tinha até bem pouco tempo atrás: onde começa e termina os conceitos da programação orientada a objetos?

Se você der uma olhada nos aspectos da POO, verá que a linguagem C não aborda todos os conceitos esperados para uma linguagem voltada a OO. No entanto, é possível fazer uma programa *baseada* (similar) a POO usando C, mas deve-se ter a ressalva de que não é a mesma coisa.

E só por curiosidade, existem muitos códigos em C que usam desta habilidade. De cabeça mesmo, cito o kernel do Linux e a biblioteca gráfica GTK+ (que usa muitos mais recursos além destes).

Atenciosamente,

2010/12/12 Marcio Tierno <mtierno@rocketmail.com>

[Quoted text hidden]

--

-fx

Marcelo Elias Del Valle <mvallebr@gmail.com>
Reply-To: UML-BR@yahoogrupos.com.br
To: UML-BR@yahoogrupos.com.br

Tue, Dec 14, 2010 at 7:25 AM

Olá pessoas!

Desculpem a demora para responder. Provavelmente minhas mensagens estarão um pouco mais demoradas essa semana, mas não falha, só tarda! :D

Seguinte, vou questionar bastante essa afirmação, mas não vou ligar se vocês me massacrarem, ok?

Qual é a definição de orientação a objetos? Existe uma diferença entre dizer que:

1. C não é uma linguagem orientada a objetos
2. Não é possível programar de forma orientada a objetos em C

Se vocês estiverem falando da primeira afirmação, eu concordo, de fato C é uma linguagem estruturada, procedural, não é OO. Contudo, não concordo com a segunda afirmação.

Se formos pensar que o compilador apenas emular o polimorfismo universal, então por que não podemos dizer que o compilador de C++ só emula também? A implementação é diferente, mas o resultado final é o mesmo, não é?

Eu considero que orientação a objetos é um conceito e, portanto, ortogonal à forma de implementação. O que desejamos com OO é separation of concerns, é reutilização do código através do desacoplamento. É estruturar a maneira como fazemos código de forma a conseguirmos isolar cada vez mais partes reutilizáveis do mesmo. Nós conseguimos isso com C, concordam?

Fui procurar por uma definição sobre polimorfismo na internet. Para fins dessa discussão, vou usar a primeira que encontrei:

(Fonte: http://en.wikipedia.org/wiki/Object-oriented_programming)

"*Polymorphism is a process in which a class has all the state and behavior of another class.*"

Dada essa definição, concordam que ela é plenamente satisfeita pela programação em C? Onde está escrito que, se a linguagem não implementar polimorfismo de inclusão você não pode implementar um programa em OO com

ela? Até a versão 1.4, java não implementava polimorfismo paramétrico. Não dava para escrever em OO com ela?

Ponto 2: não estou plenamente convencido de que não é possível implementar polimorfismo de inclusão em C, só emular. Pela definição de polimorfismo de inclusão, temos:

(Fonte: http://www.di.ubi.pt/~pprata/poo/POO_08_09_T08.pdf)

"*3.1) Polimorfismo de inclusão*

* Uma função definida num determinado tipo pode também operar todos os seus
*

*subtipos. *

*(Resulta directamente do mecanismo de herança – uma operação definida na classe *

*base é também aplicável aos objectos de todas as subclasses)***

Essa definição não é satisfeita com a implementação em C, dadas as definições de tipos e subtipos?

Abraços,
Marcelo.

Em 13 de dezembro de 2010 12:46, Fabrício Cabral <fabriciofx@gmail.com> escreveu:

> Olá todos,
>
> concordo com o MT. Aliás, esta era uma opinião errada que eu tinha até bem
> pouco tempo atrás: onde começa e termina os conceitos da programação
> orientada a objetos?
>
> Se você der uma olhada nos aspectos da POO, verá que a linguagem C
> não aborda todos os conceitos esperados para uma linguagem voltada a
> OO. No entanto, é possível fazer uma programa *baseada* (similar) a
> POO usando C, mas deve-se ter a ressalva de que não é a mesma coisa.
>
> E só por curiosidade, existem muitos códigos em C que usam desta
> habilidade.
> De cabeça mesmo, cito o kernel do Linux e a biblioteca gráfica GTK+ (que
> usa muitos mais recursos além destes).
>
> Atenciosamente,
>
> 2010/12/12 Marcio Tierno <mtierno@rocketmail.com>
>
> >
> >
> >
> > Marcelo
> >
> > Eu lembro! Hehehehehehe!!!
> >
> > Trata-se de um programa bem "safadinho" esse que vc postou o link.
> >
> > Em primeiro lugar, é preciso dizer que C tem polimorfismo por sobrecarga
> de
> > operador (overload), que é um polimorfismo estático.
> >
> > O programinha cujo link vc postou *EMULA* polimorfismo universal por
> > inclusão, mas só emula. Ele faz uso de estruturas de baixo nível nativas
> de
> > C, especificamente a habilidade de um ponteiro apontar para qualquer
> > endereço de memória - o que inclui o endereço de uma função. Como não há
> > polimorfismo por inclusão em C, o artifício (bem esperto, diga-se) é o
> uso
> > de struts, que são um tipo abstrato de dado. Struts não têm polimorfismo,
> > mas ele criou uma strut com dois campos ponteiros. E quando instancia um
> > objeto Circle ou Box, esse "construtor" faz os campos da strut apontarem
> > para os endereços correspondentes das funções para Circle ou para Box.

> >
> > Espertinho. Mas não é polimorfismo universal por inclusão, pois não é
> > simples incluir uma nova classe ou um novo método nesse modelo.
> >
> > []s
> >
> > MT.
> >
> > --- Em UML-BR@yahoogrupos.com.br <UML-BR%40yahoogrupos.com.br>, Marcelo
> > Elias Del Valle <mvallebr@...> escreveu
> >
> > >
> > > Caros,
> > >
> > > Recebi um link esses dias que achei interessante, sobre polimorfismo em
> > > C, e resolvi compartilhar com o grupo.
> > > <http://www.thradams.com/codeblog/cpoly.htm>
> > >
> > > Há tempos atrás eu tinha gerado uma polêmica sobre isso aqui no grupo,
> > > mas poucos devem se lembrar. Contudo, continuo acreditando em
> > orientação
> > a
> > > objetos mesmo em linguagens procedurais, como C.
> > > Claro que uma linguagem de mais alto nível tornaria mais fácil a
> > > separação de preocupações, mas esse link é um exemplo de que é possível
> > > aplicar os mesmos conceitos numa linguagem como C.
> > >
> > > Abraços,
> > > --
> > > Marcelo Elias Del Valle
> > > 11 6995 0065
> > >
> > >
> > > [As partes desta mensagem que não continham texto foram removidas]
> > >
> >
> >
> >
>
>
>
> --
> --fx
>
>
> [As partes desta mensagem que não continham texto foram removidas]
>
>
>
> -----
>
> Links do Yahoo! Grupos
>
>
>

--
Marcelo Elias Del Valle
11 6995 0065

[As partes desta mensagem que não continham texto foram removidas]

Links do Yahoo! Grupos

<*> Para visitar o site do seu grupo na web, acesse:
<http://br.groups.yahoo.com/group/UML-BR/>

<*> Para sair deste grupo, envie um e-mail para:
UML-BR-unsubscribe@yahoogrupos.com.br

<*> O uso que você faz do Yahoo! Grupos está sujeito aos:
<http://br.yahoo.com/info/utos.html>

Fernanda Coelho <nanda.coelho@gmail.com>

Tue, Dec 14, 2010 at 10:33 AM

Reply-To: UML-BR@yahoogrupos.com.br

To: UML-BR@yahoogrupos.com.br

Eu programo em C a mais de 10 anos. E o fato dela nao ser uma linguagem OO nao a limita na implementacao. Tudo eh questao de dominio de conceitos e projetos adequados. Java nao eh uma linguagem OO ela permite q se implemente OO mas tb permite a forma estrutural. O mesmo se aplica a linguagem C. E ainda acho C uma linguagem muitissimo eficaz, dada a sua capacidade de manipular os extremos da maquina. Existem programas que se feitos em Java, jamais chegaram de uma performance conseguida pelo mesmo programa em C, mesmo q esteja fazendo OO em C. Concordo contigo Marcelo.

Abraco

[Quoted text hidden]

Rodolpho Ugolini <rodolpho.ugolini@gmail.com>

Tue, Dec 14, 2010 at 12:11 PM

Reply-To: UML-BR@yahoogrupos.com.br

To: UML-BR@yahoogrupos.com.br

Fernanda,

Java é Orientado por Objetos. Dá pra questionar isto?

Abraço,
Rodolpho

[Quoted text hidden]

> > (Fonte: http://www.di.ubi.pt/~pprata/poo/POO_08_09_T08.pdf<http://www.di.ubi.pt/%7Epprata/poo/POO_08_09_T08.pdf>

[Quoted text hidden]

Rodolpho Ugolini <rodolpho.ugolini@gmail.com>

Tue, Dec 14, 2010 at 12:29 PM

Reply-To: UML-BR@yahoogrupos.com.br

To: UML-BR@yahoogrupos.com.br

Marcelo,

C é uma linguagem nativamente procedural, pois não tem *classes* que encapsulam comportamento e dados.

Dá sim pra simular conceitos OO em C, com mais ou menos trabalho (boa sorte ao tentar implementar herança "na unha" ;-). Ok. Os conceitos de OO são excelentes para melhorar o design (encapsulamento, acoplamento/coesão, etc.).

Mas não dá pra dizer que você está programando OO: a linguagem não te oferece os mecanismos, cabe a você implementa-los.

Um avião não é um carro, apesar de ter rodas. Você pode serrar as asas e usa-lo somente no chão, ainda assim não é um carro, apesar de funcionar.

Não existe consenso sobre uma definição "dura" do que é ou não OO. Quando "mais OO" for uma linguagem, menos trabalho dá implementar os mecanismos necessários para realizar a visão OO.

Você pode aplicar um "estilo OO" a um programa em qualquer linguagem.

Mas isto não faz "C" uma linguagem OO. Ela é procedural.

Abraço,
Rodolpho

2010/12/14 Marcelo Elias Del Valle <mvallebr@gmail.com>

[Quoted text hidden]

> (Fonte: http://www.di.ubi.pt/~pprata/poo/POO_08_09_T08.pdf<http://www.di.ubi.pt/%7Epprata/poo/POO_08_09_T08.pdf>

[Quoted text hidden]

Anderson Meira <anderson.duarte.meira@gmail.com>

Tue, Dec 14, 2010 at 12:22 PM

Reply-To: UML-BR@yahoogrupos.com.br

To: UML-BR@yahoogrupos.com.br

Rodolpho,

Parece incrível, mas sim, dá para questionar.

Java, C++ e C# dão suporte a OO. Uma linguagem realmente OO é Smalltalk.

Por exemplo, elas têm coisas que NÃO SÃO objetos, como os primitivos.

Sds,
Anderson Meira
Sun Certified Java Programmer 6.0

-----Mensagem original-----

De: UML-BR@yahoogrupos.com.br [mailto:UML-BR@yahoogrupos.com.br] Em nome de Rodolpho Ugolini

Enviada em: terça-feira, 14 de dezembro de 2010 13:11

Para: UML-BR@yahoogrupos.com.br

Assunto: Re: [UML-BR] Re: polimorfismo em C

Fernanda,

Java é Orientado por Objetos. Dá pra questionar isto?

Abraço,
Rodolpho

| [através de email](#) | [Responder através da web](#) | [Adicionar um novo tópico](#)
[Mensagens neste tópico \(8\)](#)

ATIVIDADE NOS ÚLTIMOS DIAS: [Novos usuários 3](#) |
[Visite seu Grupo](#)

Rodolpho Ugolini <rodolpho.ugolini@gmail.com>
Reply-To: UML-BR@yahoogrupos.com.br
To: UML-BR@yahoogrupos.com.br

Tue, Dec 14, 2010 at 12:45 PM

Anderson,

Como eu disse, não existe um consenso do que faz uma linguagem ser considerada OO ou não.

Java, C++, C# e Smalltalk são OO, pois permitem aplicar os conceitos OO (encapsulamento, hierarquia e polimorfismo) *nativamente*. C, não.

Agora, a implementação destes mecanismos varia de linguagem para linguagem. SmallTalk é mais elegante e implementa nativamente mais conceitos OO.

Em 97 tive oportunidade de conversar com o James Gosling em 97, quando ele esteve no Brasil, sobre os tipos primitivos. A explicação foi simples: performance. Java deveria rodar em coisas minúsculas e limitadas.

Isso não faz dela uma linguagem procedural, faz?

Se eu tiver 2 colunas: Procedural | Orientado à Objetos

Onde você colocaria Java? E "C" ?

Me aponte uma referência que sustente a sua argumentação de que Java não é Orientado por Objetos. E se não é OO, é procedural?

Abraço!
Rodolpho

2010/12/14 Anderson Meira <anderson.duarte.meira@gmail.com>

>
>
> Rodolpho,
>
> Parece incrível, mas sim, dá para questionar.
>
> Java, C++ e C# dão suporte a OO. Uma linguagem realmente OO é Smalltalk.
>
> Por exemplo, elas têm coisas que NÃO SÃO objetos, como os primitivos.
>
> Sds,
> Anderson Meira
> Sun Certified Java Programmer 6.0
>
> -----Mensagem original-----
> De: UML-BR@yahoogrupos.com.br <UML-BR%40yahoogrupos.com.br> [mailto:
> UML-BR@yahoogrupos.com.br <UML-BR%40yahoogrupos.com.br>] Em nome de
> Rodolpho Ugolini
> Enviada em: terça-feira, 14 de dezembro de 2010 13:11
> Para: UML-BR@yahoogrupos.com.br <UML-BR%40yahoogrupos.com.br>
> Assunto: Re: [UML-BR] Re: polimorfismo em C
>
>
> Fernanda,
>
> Java é Orientado por Objetos. Dá pra questionar isto?

>
> Abraço,
> Rodolpho
>
>
>

[Quoted text hidden]

Marcelo Elias Del Valle <mvallebr@gmail.com>

Tue, Dec 14, 2010 at 1:02 PM

Reply-To: UML-BR@yahoogrupos.com.br

To: UML-BR@yahoogrupos.com.br

Rodolpho,

Eu classificaria Java como parcialmente OO. Se a discussão for a linguagem ser OO ou não, eu concordo que C seja procedural. Se smalltalk ou ruby são OO? Precisaria de uma definição mais formal do que é OO. Amanhã vão inventar um novo tipo de polimorfismo ou outra maneira de implementar separação de preocupações e vão dizer que smalltalk tb não é OO. Contudo, implementar um programa de forma orientada a objetos não depende da linguagem que você usa, depende dos conceitos que você utiliza. São conceitos relacionados a OO: separação de preocupações, herança, polimorfismo, agregações, composições... Se você conseguir implementar esses conceitos em C, ótimo!

Vamos pensar, por exemplo, no UML. Há um tempo, aqui na lista, tivemos a discussão sobre a diferença entre composição e agregação. Eu lembro que aprendi naquela discussão que, se você garante, no seu programa, que o ciclo de vida de um objeto está relacionado com o ciclo de vida do outro, é uma composição.

Você não consegue, apenas com a linguagem, diferenciar uma composição de uma agregação em Java. Você depende da maneira com a qual você programa para garantir esses conceitos.

Da mesma forma, você não consegue, apenas com a linguagem, aplicar polimorfismo em C. Você depende da maneira com a qual você programa para garantir esse conceito.

Abraços,
Marcelo.

Em 14 de dezembro de 2010 13:45, Rodolpho Ugolini <rodolpho.ugolini@gmail.com> escreveu:

> Anderson,
>
> Como eu disse, não existe um consenso do que faz uma linguagem ser
> considerada OO ou não.
>
> Java, C++, C# e Smalltalk são OO, pois permitem aplicar os conceitos OO
> (encapsulamento, hierarquia e polimorfismo) *nativamente*. C, não.
>
> Agora, a implementação destes mecanismos varia de linguagem para linguagem.
> SmallTalk é mais elegante e implementa nativamente mais conceitos OO.
>
> Em 97 tive oportunidade de conversar com o James Gosling em 97, quando ele
> esteve no Brasil, sobre os tipos primitivos. A explicação foi simples:
> performance. Java deveria rodar em coisas minúsculas e limitadas.
>
> Isso não faz dela uma linguagem procedural, faz?
>
> Se eu tiver 2 colunas: Procedural | Orientado à Objetos

```
> Onde você colocaria Java? E "C" ?
>
> Me aponte uma referência que sustente a sua argumentação de que Java não é
> Orientado por Objetos. E se não é OO, é procedural?
>
> Abraço!
> Rodolpho
>
> 2010/12/14 Anderson Meira <anderson.duarte.meira@gmail.com>
>
> >
> >
> > Rodolpho,
> >
> > Parece incrível, mas sim, dá para questionar.
> >
> > Java, C++ e C# dão suporte a OO. Uma linguagem realmente OO é Smalltalk.
> >
> > Por exemplo, elas têm coisas que NÃO SÃO objetos, como os primitivos.
> >
> > Sds,
> > Anderson Meira
> > Sun Certified Java Programmer 6.0
> >
> > -----Mensagem original-----
> > De: UML-BR@yahoo grupos .com.br <UML-BR%40yahoo grupos .com.br> [mailto:UML-BR@yahoo grupos .com.br] Em nome de
> > Rodolpho Ugolini
> > Enviada em: terça-feira, 14 de dezembro de 2010 13:11
> > Para: UML-BR@yahoo grupos .com.br <UML-BR%40yahoo grupos .com.br>
> > Assunto: Re: [UML-BR] Re: polimorfismo em C
> >
> >
> > Fernanda,
> >
> > Java é Orientado por Objetos. Dá pra questionar isto?
> >
> > Abraço,
> > Rodolpho
> >
> >
> >
> >
> [As partes desta mensagem que não continham texto foram removidas]
>
>
>
> -----
>
> Links do Yahoo! Grupos
>
>
>
```

—
Marcelo Elias Del Valle
11 6995 0065

[As partes desta mensagem que não continham texto foram removidas]

_____ * _____ ; _____ * _____

| [através de email](#) | [Responder através da web](#) | [Adicionar um novo tópico](#)
 Mensagens neste tópico (10)

ATIVIDADE NOS ÚLTIMOS DIAS: [Novos usuários](#) 3 |
[Visite seu Grupo](#)

Média de 7 profissionais contratados por hora! 1 a cada 9 minutos!

Quer ser o próximo?Veja histórias de quem encontrou quem faltava!

Hotels.com - Reveillon na praia a partir de R\$ 89

YAHOO! GRUPOS
BRASIL

Trocar para: [SÃ³ Texto](#), [Resenha DiÃ¡ria](#) • [Sair do grupo](#) • [Termos de uso](#)

_____? _' _? _____

Thiago Nascimento <nascimentthiago@gmail.com>

Tue, Dec 14, 2010 at 1:07 PM

Reply-To: UML-BR@yahoogrupos.com.br

To: UML-BR@yahoogrupos.com.br

Outra perspectiva a ser analisada:

OOP has been widely available to developers for about 15 years, mainly because of the popularity of the C++ programming language. However, because of its C heritage, C++ is not completely object-oriented; you can write C++ programs that do not use object-oriented features. In contrast, Java is completely object-oriented: every Java program requires the presence of at least one class (an object-oriented feature). Furthermore, Java's official language definition includes the term object-oriented. (See Resources<<http://www.javaworld.com/javaworld/jw-04-2001/jw-0406-java101.html#resources>>.)

<http://www.javaworld.com/javaworld/jw-04-2001/jw-0406-java101.html>

<http://java.sun.com/docs/overviews/java/java-overview-1.html>

Sob essa perspectiva, Java é completamente OO.

2010/12/14 Marcelo Elias Del Valle <mvallebr@gmail.com>

>
>

[Quoted text hidden]

> rodolpho.ugolini@gmail.com <rodolpho.ugolini%40gmail.com>> escreveu:

>
>

> > Anderson,

> >

> > Como eu disse, não existe um consenso do que faz uma linguagem ser
 > > considerada OO ou não.

> >

> > Java, C++, C# e Smalltalk são OO, pois permitem aplicar os conceitos OO
 > > (encapsulamento, hierarquia e polimorfismo) *nativamente*. C, não.

> >

> > Agora, a implementação destes mecanismos varia de linguagem para
 > linguagem.

> > SmallTalk é mais elegante e implementa nativamente mais conceitos OO.

> >

> > Em 97 tive oportunidade de conversar com o James Gosling em 97, quando
> ele
> > esteve no Brasil, sobre os tipos primitivos. A explicação foi simples:
> > performance. Java deveria rodar em coisas minúsculas e limitadas.
> >
> > Isso não faz dela uma linguagem procedural, faz?
> >
> > Se eu tiver 2 colunas: Procedural | Orientado à Objetos
> >
> > Onde você colocaria Java? E "C" ?
> >
> > Me aponte uma referência que sustente a sua argumentação de que Java não
> é
> > Orientado por Objetos. E se não é OO, é procedural?
> >
> > Abraço!
> > Rodolpho
> >
> > 2010/12/14 Anderson Meira <anderson.duarte.meira@gmail.com><anderson.duarte.meira%40gmail.com>
> >
> >
> > >
> > >
> > > Rodolpho,
> > >
> > > Parece incrível, mas sim, dá para questionar.
> > >
> > > Java, C++ e C# dão suporte a OO. Uma linguagem realmente OO é
> Smalltalk.
> > >
> > > Por exemplo, elas têm coisas que NÃO SÃO objetos, como os primitivos.
> > >
> > > Sds,
> > > Anderson Meira
> > > Sun Certified Java Programmer 6.0
> > >
> > > -----Mensagem original-----
> > > De: UML-BR@yahoogrupos.com.br <UML-BR%40yahoogrupos.com.br> <UML-BR%40yahoogrupos.com.br> [mailto:
> > > UML-BR@yahoogrupos.com.br <UML-BR%40yahoogrupos.com.br> <UML-BR%40yahoogrupos.com.br>] Em nome de
> > > Rodolpho Ugolini
> > > Enviada em: terça-feira, 14 de dezembro de 2010 13:11
> > > Para: UML-BR@yahoogrupos.com.br <UML-BR%40yahoogrupos.com.br> <UML-BR%40yahoogrupos.com.br>
> > > Assunto: Re: [UML-BR] Re: polimorfismo em C
> > >
> > >
> > > Fernanda,
> > >
> > > Java é Orientado por Objetos. Dá pra questionar isto?
> > >
> > > Abraço,
> > > Rodolpho
> > >
> > >
> > >
> > >
> > [As partes desta mensagem que não continham texto foram removidas]
> >
> >
> >
> > -----
> >

> > Links do Yahoo! Grupos

> >

> >

> >

>

> --

> Marcelo Elias Del Valle

> 11 6995 0065

>

> [As partes desta mensagem que não continham texto foram removidas]

>

>

>

--

Thiago Nascimento

perl -e '\$_="tMM naaCt Feocmama_itpUilucoGa";\$_=\$1,print \$2 while

s/(.)(.)/;print substr\$_,1,1;'

"...just because I don't know the meaning of my art, does not mean it has no meaning..." S.D.

[As partes desta mensagem que não continham texto foram removidas]

Links do Yahoo! Grupos

[Quoted text hidden]

Marcio Tierno <mtierno@rocketmail.com>

Reply-To: UML-BR@yahoogrupos.com.br

To: UML-BR@yahoogrupos.com.br

Tue, Dec 14, 2010 at 1:08 PM

Fernanda

Acho que ninguém está questionando C, nem atacando a linguagem. Eu mesmo adoro C. Quando era adolescente, eu cantava de galo junto a meus amigos nerds pq eu sabia programar em C. Eles me olhavam com um misto de admiração e inveja, além de uma ponta de ceticismo ("será que esse cara sabe mesmo programar em C?").

Como C é quase um assembler, dá pra fazer qualquer coisa com ela. Mas não dá pra dizer que C seja OO ou mesmo que seja possível programar OO em C, pela ausência de herança (emulável, como mostrou nosso amigo Marcelo) e pela total ausência de encapsulamento - isso não dá pra emular.

Então, só pra reforçar: C é legal. Não, C é muuuuuito legal. É linguagem de gente grande. Mas não é OO.

Inclusive, um cara que foi ídolo da minha geração, o Stroustrup, criou um C orientado a objetos: chama C++.

[]s!!

MT.

-- Em UML-BR@yahoogrupos.com.br, Fernanda Coelho <nanda.coelho@...> escreveu

>

> Eu programo em C a mais de 10 anos. E o fato dela nao ser uma linguagem OO

> nao a limita na implementacao. Tudo eh questao de dominio de conceitos e

> projetos adequados. Java nao eh uma linguagem OO ela permite q se implemente
 > OO mas tb permite a forma estrutural. O mesmo se aplica a linguagem C. E
 > ainda acho C uma linguagem muitissimo eficaz, dada a sua capacidade de
 > manipular os extremos da maquina. Existem programas que se feitos em Java,
 > jamais chegaram de uma performance conseguida pelo mesmo programa em C,
 > mesmo q esteja fazendo OO em C. Concordo contigo Marcelo.
 >
 > Abraco
 >
 > 2010/12/14 Marcelo Elias Del Valle <mvallebr@...>

[Quoted text hidden]

> > (Fonte: http://en.wikipedia.org/wiki/Object-oriented_programming)
 > > "**Polymorphism is a process in which a class has all the state and
 > > behavior of another class.**"
 > > Dada essa definição, concordam que ela é plenamente satisfeita pela
 > > programação em C? Onde está escrito que, se a linguagem não implementar
 > > polimorfismo de inclusão você não pode implementar um programa em OO com
 > > ela? Até a versão 1.4, java não implementava polimorfismo paramétrico. Não
 > > dava para escrever em OO com ela?
 > > Ponto 2: não estou plenamente convencido de que não é possível
 > > implementar polimorfismo de inclusão em C, só emular. Pela definição de
 > > polimorfismo de inclusão, temos:
 > > (Fonte: http://www.di.ubi.pt/~pprata/poo/POO_08_09_T08.pdf)
 > > "**3.1) Polimorfismo de inclusão*
 > > * Uma função definida num determinado tipo pode também operar todos os seus
 > > *
 > > *subtipos. *
 > > *(Resulta directamente do mecanismo de herança – uma operação definida na
 > > classe *
 > > *base é também aplicável aos objectos de todas as subclasses)**"
 > > Essa definição não é satisfeita com a implementação em C, dadas as
 > > definições de tipos e subtipos?
 > >
 > > Abraços,
 > > Marcelo.
 > >
 > > Em 13 de dezembro de 2010 12:46, Fabrício Cabral <fabriciofx@...
 > > > escreveu:
 > > >
 > > > Olá todos,
 > > >
 > > > concordo com o MT. Aliás, esta era uma opinião errada que eu tinha até
 > > > bem
 > > > pouco tempo atrás: onde começa e termina os conceitos da programação
 > > > orientada a objetos?
 > > >
 > > > Se você der uma olhada nos aspectos da POO, verá que a linguagem C
 > > > não aborda todos os conceitos esperados para uma linguagem voltada a
 > > > OO. No entanto, é possível fazer uma programa *baseada* (similar) a
 > > > POO usando C, mas deve-se ter a ressalva de que não é a mesma coisa.
 > > >
 > > > E só por curiosidade, existem muitos códigos em C que usam desta
 > > > habilidade.
 > > > De cabeça mesmo, cito o kernel do Linux e a biblioteca gráfica GTK+ (que
 > > > usa muitos mais recursos além destes).
 > > >
 > > > Atenciosamente,
 > > >
 > > > 2010/12/12 Marcio Tierno <mtierno@...>

[Quoted text hidden]

> > > > --- Em UML-BR@yahoogrupos.com.br <UML-BR%40yahoogrupos.com.br>,
 > > > Marcelo
 > > > > Elias Del Valle <mvallebr@...> escreveu
 > > > >

> > > >
> > > > Caros,
> > > >
> > > > Recebi um link esses dias que achei interessante, sobre polimorfismo
> > em
> > > > C, e resolvi compartilhar com o grupo.
> > > > <http://www.thradams.com/codeblog/cpoly.htm>

[Quoted text hidden]

| [através de email](#) | [Responder através da web](#) | [Adicionar um novo tópico](#)
[Mensagens neste tópico \(12\)](#)

ATIVIDADE NOS ÚLTIMOS DIAS: [Novos usuários 3](#) |
[Visite seu Grupo](#)

[VideoChat Online. Converse com pretendentes em tempo real!](#)

[Hotels.com - Reveillon na praia a partir de R\\$ 89](#)

[Mais de 200 mil Vagas de EMPREGO, em 1.600 cidades brasileiras.](#)

YAHOO! GRUPOS
BRASIL

Trocar para: [SÃ Textos](#), [Resenha Diária](#) • [Sair do grupo](#) • [Termos de uso](#)

Rodolpho Ugolini <rodolpho.ugolini@gmail.com>
Reply-To: UML-BR@yahoogrupos.com.br
To: UML-BR@yahoogrupos.com.br

Tue, Dec 14, 2010 at 1:32 PM

Marcelo,

Fui buscar o começo da discussão pra lembrar qual era mesmo o objetivo da discussão: "continuo acreditando em orientação a objetos mesmo em linguagens procedurais, como C. Claro que uma linguagem de mais alto nível tornaria mais fácil a separação de preocupações, mas esse link é um exemplo de que é possível aplicar os mesmos conceitos numa linguagem como C."

Vamos definir como base da nossa discussão que o conceito mínimo uma linguagem precisa implementar para ser considerada OO é encapsulamento: dados e comportamento juntos, numa mesma "entidade", sendo que o acesso direto aos dados é vedado externamente. Também deve ser possível criar hierarquias entre estas entidades.

Concorda que é o mínimo?

Em Java, C++, SmallTalk, Eiffel, Ruby, etc. é nativo, utilizando classes e herança.

Para implementar em C, usando força bruta, você pode colocar ponteiros de função em structs. Mas não fica perfeito. Se for implementar herança então... vai ter de ser "na mão".

O problema é que o ganho, conforme a complexidade do código aumenta, vai caindo. Até ficar "negativo", ficando mais difícil entender e dar manutenção no código com esses artifícios do que usando um código procedural elegante.

A minha conclusão e observação final desta thread é: apesar de ser possível, adotar um "estilo OO" em programas de linguagens procedurais, simulando

mecanismos OO, questiono os benefícios destes artifícios (a não ser a satisfação pessoal do sujeito que quer testar os limites da linguagem). Além do que, isto não torna a linguagem menos procedural (ou mais OO), nem o programa mais legível, e portanto, manutenível.

Abraço!
Rodolpho

2010/12/14 Marcelo Elias Del Valle <mvallebr@gmail.com>

>
>
> Rodolpho,
>
> Eu classificaria Java como parcialmente OO. Se a discussão for a
> linguagem ser OO ou não, eu concordo que C seja procedural. Se smalltalk ou
> ruby são OO? Precisaria de uma definição mais formal do que é OO. Amanhã vão
> inventar um novo tipo de polimorfismo ou outra maneira de implementar
> separação de preocupações e vão dizer que smalltalk tb não é OO.
> Contudo, implementar um programa de forma orientada a objetos não
> depende da linguagem que você usa, depende dos conceitos que você utiliza.
> São conceitos relacionados a OO: separação de preocupações, herança,
> polimorfismo, agregações, composições... Se você conseguir implementar
> esses
> conceitos em C, ótimo!
> Vamos pensar, por exemplo, no UML. Há um tempo, aqui na lista, tivemos a
> discussão sobre a diferença entre composição e agregação. Eu lembro que
> aprendi naquela discussão que, se você garante, no seu programa, que o
> ciclo
> de vida de um objeto está relacionado com o ciclo de vida do outro, é uma
> composição.
> Você não consegue, apenas com a linguagem, diferenciar uma composição de
> uma agregação em Java. Você depende da maneira com a qual você programa
> para
> garantir esses conceitos.
> Da mesma forma, você não consegue, apenas com a linguagem, aplicar
> polimorfismo em C. Você depende da maneira com a qual você programa para
> garantir esse conceito.
>
> Abraços,
> Marcelo.
>
> Em 14 de dezembro de 2010 13:45, Rodolpho Ugolini <
> rodolpho.ugolini@gmail.com <rodolpho.ugolini%40gmail.com>> escreveu:
>
>
> > Anderson,
> >
> > Como eu disse, não existe um consenso do que faz uma linguagem ser
> > considerada OO ou não.
> >
> > Java, C++, C# e Smalltalk são OO, pois permitem aplicar os conceitos OO
> > (encapsulamento, hierarquia e polimorfismo) *nativamente*. C, não.
> >
> > Agora, a implementação destes mecanismos varia de linguagem para
> > linguagem.
> > SmallTalk é mais elegante e implementa nativamente mais conceitos OO.
> >
> > Em 97 tive oportunidade de conversar com o James Gosling em 97, quando
> > ele
> > esteve no Brasil, sobre os tipos primitivos. A explicação foi simples:
> > performance. Java deveria rodar em coisas minúsculas e limitadas.
> >
> > Isso não faz dela uma linguagem procedural, faz?
> >

> > Se eu tiver 2 colunas: Procedural | Orientado à Objetos
> >
> > Onde você colocaria Java? E "C" ?
> >
> > Me aponte uma referência que sustente a sua argumentação de que Java não
> > é
> > Orientado por Objetos. E se não é OO, é procedural?
> >
> > Abraço!
> > Rodolpho
> >
> > 2010/12/14 Anderson Meira <anderson.duarte.meira@gmail.com<anderson.duarte.meira%40gmail.com>
> >
> >
> > >
> > >
> > > Rodolpho,
> > >
> > > Parece incrível, mas sim, dá para questionar.
> > >
> > > Java, C++ e C# dão suporte a OO. Uma linguagem realmente OO é
> > Smalltalk.
> > >
> > > Por exemplo, elas têm coisas que NÃO SÃO objetos, como os primitivos.
> > >
> > > Sds,
> > > Anderson Meira
> > > Sun Certified Java Programmer 6.0
> > >
> > > -----Mensagem original-----
> > > De: UML-BR@yahoogrupos.com.br <UML-BR%40yahoogrupos.com.br> <UML-BR%40yahoogrupos.com.br> [mailto:
> > > UML-BR@yahoogrupos.com.br <UML-BR%40yahoogrupos.com.br> <UML-BR%40yahoogrupos.com.br>] Em nome de
> > >
> > > Rodolpho Ugolini
> > > Enviada em: terça-feira, 14 de dezembro de 2010 13:11
> > > Para: UML-BR@yahoogrupos.com.br <UML-BR%40yahoogrupos.com.br> <UML-BR%40yahoogrupos.com.br>
> > >
> > > Assunto: Re: [UML-BR] Re: polimorfismo em C
> > >
> > >
> > > Fernanda,
> > >
> > > Java é Orientado por Objetos. Dá pra questionar isto?
> > >
> > > Abraço,
> > > Rodolpho
> > >
> > >
> > >
> > >
> > [As partes desta mensagem que não continham texto foram removidas]
> >
> >
> >
> > -----
> >
> > Links do Yahoo! Grupos
> >
> >
> >
>

> --
> Marcelo Elias Del Valle
> 11 6995 0065
>
> [As partes desta mensagem que não continham texto foram removidas]
>
>
>

[As partes desta mensagem que não continham texto foram removidas]

Links do Yahoo! Grupos

[Quoted text hidden]

Marcio Tierno <mtierno@rocketmail.com>
Reply-To: UML-BR@yahoogrupos.com.br
To: UML-BR@yahoogrupos.com.br

Tue, Dec 14, 2010 at 1:33 PM

Marcelo

Sobre o ponto 2 (já que concordamos com o 1):

É possível programar seguindo um "estilo OO" em qualquer linguagem. Até assembler. Uma vez em trabalhei numa firma em que o chefe decidiu que iria fazer do Oracle 7 um banco OO. Bom, o Oracle 7 ainda nem era um banco relacional decente (vivia dando tela azul no windows NT), que dirá transformá-lo em OO? O fato é que o cara fez. Deu um trabalho dos infernos. Pra falar a verdade, daria para ter implementado 2 sistemas inteiros só com o esforço que o cara dispendeu no framework. Ficou bem "OO". Pena que na hora de rodar a performance ficou um lixo e tivemos que voltar para os selects e inserts da vida.

Mas não é possível implementar herança, por exemplo. O que vc mostrou no início deste post foi uma emulação de polimorfismo universal por inclusão, mas que, na verdade, é polimorfismo ad hoc. Para ter herança, tem que ser possível criar uma classe filha que simplesmente herde tudo da classe mãe, sem necessidade de qualquer outra implementação que não a declaração da classe. Não dá pra fazer isso em C. Não dá para ter construtores e destrutores, não dá para ter encapsulamento de verdade.

Mas, concordo que dá para usar as diretrizes de OO em C.

Mais do que isso, fica contraproducente. Veja o exemplinho que vc deu. Dá mais trabalho programar a emulação do polimorfismo do que a funcionalidade do programa.

Sobre as definições que vc postou, a primeira sobre polimorfismo está errada. Ela define herança, já que polimorfismo não é inerente a classes mas, sim, a operações (definição de métodos).

Então, para usarmos uma fonte mais acadêmica e confiável, postei na área de arquivos do grupo o artigo seminal do Caredelli e do Wegner sobre polimorfismo, publicado originalmente no ACM Computer Surveys, de 1985(página do grupo, link arquivos, título "ACM Computer Surveys Polymorphism.pdf"). É neste artigo que pela primeira vez se amplia os tipos originais de polimorfismo, resultando nos 4 tipos principais aceitos hoje: polimorfismo universal por inclusão, polimorfismo universal paramétrico, polimorfismo ad hoc por sobrecarga e polimorfismo ad hoc por coerção.

Para finalizar: se para "implementar" construções OO numa linguagem que não as possui nativamente você vai gastar mais tempo e esforço do que para fazer a aplicação, essa linguagem não é OO.

[]s

MT.

--- Em UML-BR@yahoogrupos.com.br, Marcelo Elias Del Valle <mvallebr@...> escreveu

>

> Olá pessoas!

>

> Desculpem a demora para responder. Provavelmente minhas mensagens estarão

> um pouco mais demoradas essa semana, mas não falha, só tarda! :D

> Seguinte, vou questionar bastante essa afirmação, mas não vou ligar se

> vocês me massacrarem, ok?

> Qual é a definição de orientação a objetos? Existe uma diferença entre

> dizer que:

>

> 1. C não é uma linguagem orientada a objetos

> 2. Não é possível programar de forma orientada a objetos em C

>

>

> Se vocês estiverem falando da primeira afirmação, eu concordo, de fato C

> é uma linguagem estruturada, procedural, não é OO. Contudo, não concordo com

> a segunda afirmação.

> Se formos pensar que o compilador apenas emular o polimorfismo

> universal, então por que não podemos dizer que o compilador de C++ só emula

> também? A implementação é diferente, mas o resultado final é o mesmo, não é?

> Eu considero que orientação a objetos é um conceito e, portanto,

> ortogonal à forma de implementação. O que desejamos com OO é separation of

> concerns, é reutilização do código através do desacoplamento. É estruturar a

> maneira como fazemos código de forma a conseguirmos isolar cada vez mais

> partes reutilizáveis do mesmo. Nós conseguimos isso com C, concordam?

> Fui procurar por uma definição sobre polimorfismo na internet. Para fins

> dessa discussão, vou usar a primeira que encontrei:

> (Fonte: http://en.wikipedia.org/wiki/Object-oriented_programming)

> "Polymorphism is a process in which a class has all the state and

> behavior of another class."

> Dada essa definição, concordam que ela é plenamente satisfeita pela

> programação em C? Onde está escrito que, se a linguagem não implementar

> polimorfismo de inclusão você não pode implementar um programa em OO com

> ela? Até a versão 1.4, java não implementava polimorfismo paramétrico. Não

> dava para escrever em OO com ela?

> Ponto 2: não estou plenamente convencido de que não é possível

> implementar polimorfismo de inclusão em C, só emular. Pela definição de

> polimorfismo de inclusão, temos:

> (Fonte: http://www.di.ubi.pt/~pprata/poo/POO_08_09_T08.pdf)

> "3.1) Polimorfismo de inclusão"

> * Uma função definida num determinado tipo pode também operar todos os seus

> *

> *subtipos. *

> *(Resulta directamente do mecanismo de herança – uma operação definida na

> classe *

> *base é também aplicável aos objectos de todas as subclasses)"

> Essa definição não é satisfeita com a implementação em C, dadas as

> definições de tipos e subtipos?

>

> Abraços,

> Marcelo.

>

> Em 13 de dezembro de 2010 12:46, Fabrício Cabral <fabriciofx@...> escreveu:

>

> > Olá todos,

> >

> > concordo com o MT. Aliás, esta era uma opinião errada que eu tinha até bem

> > pouco tempo atrás: onde começa e termina os conceitos da programação

> > orientada a objetos?

> >

> > Se você der uma olhada nos aspectos da POO, verá que a linguagem C

> > não aborda todos os conceitos esperados para uma linguagem voltada a

> > OO. No entanto, é possível fazer uma programa *baseada* (similar) a
> > POO usando C, mas deve-se ter a ressalva de que não é a mesma coisa.
> >
> > E só por curiosidade, existem muitos códigos em C que usam desta
> > habilidade.
> > De cabeça mesmo, cito o kernel do Linux e a biblioteca gráfica GTK+ (que
> > usa muitos mais recursos além destes).
> >
> > Atenciosamente,
> >
> > 2010/12/12 Marcio Tierno <mtierno@...>
> >
> > >
> > >
> > >
> > > Marcelo
> > >
> > > Eu lembro! Hehehehehehe!!!
> > >
> > > Trata-se de um programa bem "safadinho" esse que vc postou o link.
> > >
> > > Em primeiro lugar, é preciso dizer que C tem polimorfismo por sobrecarga
> > de
> > > operador (overload), que é um polimorfismo estático.
> > >
> > > O programinha cujo link vc postou *EMULA* polimorfismo universal por
> > > inclusão, mas só emula. Ele faz uso de estruturas de baixo nível nativas
> > de
> > > C, especificamente a habilidade de um ponteiro apontar para qualquer
> > > endereço de memória - o que inclui o endereço de uma função. Como não há
> > > polimorfismo por inclusão em C, o artifício (bem esperto, diga-se) é o
> > uso
> > > de struts, que são um tipo abstrato de dado. Struts não têm polimorfismo,
> > > mas ele criou uma strut com dois campos ponteiros. E quando instancia um
> > > objeto Circle ou Box, esse "construtor" faz os campos da strut apontarem
> > > para os endereços correspondentes das funções para Circle ou para Box.
> > >
> > > Espertinho. Mas não é polimorfismo universal por inclusão, pois não é
> > > simples incluir uma nova classe ou um novo método nesse modelo.
> > >
> > > []s
> > >
> > > MT.
> > >
> > > --- Em UML-BR@yahoogrupos.com.br <UML-BR%40yahoogrupos.com.br>, Marcelo
> > > Elias Del Valle <mvallebr@> escreveu
> > >
> > > >
> > > > Caros,
> > > >
> > > > Recebi um link esses dias que achei interessante, sobre polimorfismo em
> > > > C, e resolvi compartilhar com o grupo.
> > > > <http://www.thradams.com/codeblog/cpoly.htm>
> > > >
> > > > Há tempos atrás eu tinha gerado uma polêmica sobre isso aqui no grupo,
> > > > mas poucos devem se lembrar. Contudo, continuo acreditando em
> > orientação
> > > a
> > > > objetos mesmo em linguagens procedurais, como C.
> > > > Claro que uma linguagem de mais alto nível tornaria mais fácil a
> > > > separação de preocupações, mas esse link é um exemplo de que é possível
> > > > aplicar os mesmos conceitos numa linguagem como C.
> > > >
> > > > Abraços,
> > > > --

_____ * _____ ; _____ *

ATIVIDADE NOS ÚLTIMOS DIAS: [Novos usuários 3](#) | [Novos arquivos 1](#) | [Visite seu Grupo](#)

YAHOO! GRUPOS
BRASIL

_____? _____? _____?

Tue, Dec 14, 2010 at 2:24 PM

<https://mail.google.com/mail/u/0/?ui=2&ik=a514c0c01e&view=pt&q=uml-br%20fabriciofx%40gmail.com&gs=true&search=query&th=12cd7fddda7e133...> 21/42

editora, para programar OO em Assembler.

Bem, essa discussão é antigo. e o C é procedural. Na implementação do C++, decidiram implementar as condições para se programar OO. Portanto, o C++ não é uma linguagem OO pura, como o Smalltalk e o Java.

Assim é possível programar OO em C++, só que tem que saber bem os conceitos de OO para não fazer uma "misturada" de conceitos OO com procedural. Aí vira uma bagunça só.

Já vi programas que o autor dizia que era OO por ter sido programado em C++ e era totalmente procedural.

Abraços

[Quoted text hidden]

> --- Em UML-BR@yahoogrupos.com.br <UML-BR%40yahoogrupos.com.br>, Marcelo

[Quoted text hidden]

> > (Fonte: http://en.wikipedia.org/wiki/Object-oriented_programming)

> > "Polymorphism is a process in which a class has all the state and

> > behavior of another class."

> > Dada essa definição, concordam que ela é plenamente satisfeita pela

> > programação em C? Onde está escrito que, se a linguagem não implementar

> > polimorfismo de inclusão você não pode implementar um programa em OO com

> > ela? Até a versão 1.4, java não implementava polimorfismo paramétrico.

> Não

> > dava para escrever em OO com ela?

> > Ponto 2: não estou plenamente convencido de que não é possível

> > implementar polimorfismo de inclusão em C, só emular. Pela definição de

> > polimorfismo de inclusão, temos:

> > (Fonte: http://www.di.ubi.pt/~pprata/poo/POO_08_09_T08.pdf<http://www.di.ubi.pt/%7Epprata/poo/POO_08_09_T08.pdf>

[Quoted text hidden]

> > > --- Em UML-BR@yahoogrupos.com.br <UML-BR%40yahoogrupos.com.br><UML-BR%

> > > 40yahoogrupos.com.br>, Marcelo

> > > > Elias Del Valle <mvallebr@> escreveu

> > > >

> > > >

> > > > Caros,

> > > >

> > > > Recebi um link esses dias que achei interessante, sobre

> > > > polimorfismo em

> > > > C, e resolvi compartilhar com o grupo.

> > > > <http://www.thradams.com/codeblog/cpoly.htm>

> > > >

> > > > Há tempos atrás eu tinha gerado uma polêmica sobre isso aqui no

> > > > grupo,

> > > > mas poucos devem se lembrar. Contudo, continuo acreditando em

> > > > orientação

> > > > a

> > > > objetos mesmo em linguagens procedurais, como C.

> > > > Claro que uma linguagem de mais alto nível tornaria mais fácil a

> > > > separação de preocupações, mas esse link é um exemplo de que é

> > > > possível

> > > > aplicar os mesmos conceitos numa linguagem como C.

> > > >

> > > > Abraços,

> > > > --

> > > > Marcelo Elias Del Valle

> > > > 11 6995 0065

> > > >

> > > >

> > > > [As partes desta mensagem que não continham texto foram removidas]

> > > >

> > > >

> > > >

> > > >

> > >

> > >

> > >
> > > --
> > > -fx
> > >
> > >
> > > [As partes desta mensagem que não continham texto foram removidas]
> > >
> > >
> > >
> > > -----
> > >
> > > Links do Yahoo! Grupos
> > >
> > >
> > >
> > >
> > > --
> > > Marcelo Elias Del Valle
> > > 11 6995 0065
> > >
> > >
> > > [As partes desta mensagem que não continham texto foram removidas]
> > >
>
>
>

--
José Carlos Alão de Oliveira

[As partes desta mensagem que não continham texto foram removidas]

Links do Yahoo! Grupos

[Quoted text hidden]

Rodolpho Ugolini <rodolpho.ugolini@gmail.com>

Tue, Dec 14, 2010 at 2:28 PM

Reply-To: UML-BR@yahoogrupos.com.br

To: UML-BR@yahoogrupos.com.br

José Carlos,

Você disse: "concordo com o Marcelo". Só não entendi em qual ponto...

Rodolpho

[Quoted text hidden]

José Carlos Alão de Oliveira <jcalao44@gmail.com>

Tue, Dec 14, 2010 at 2:53 PM

Reply-To: UML-BR@yahoogrupos.com.br

To: UML-BR@yahoogrupos.com.br

No ponto em que você pode programar OO até em assembler, parcialmente, pois para ter herança é bem complicado para implementar. O esforço não compensa.

Em 14 de dezembro de 2010 15:28, Rodolpho Ugolini <rodolpho.ugolini@gmail.com> escreveu:

[Quoted text hidden]

pauloc <paulopcsys@yahoo.com.br>
Reply-To: UML-BR@yahoogrupos.com.br
To: UML-BR@yahoogrupos.com.br

Tue, Dec 14, 2010 at 3:40 PM

Bom dia,

Leio com atenção as mensagens neste forum e, sobre o assunto em discussão sugiro aos colegas que recorram as transcrições de Bjarne Stroustrup (<http://www2.research.att.com/~bs/homepage.html>), inventor e implementador do C++.

Seu livro The C++ Programming Language (<http://www2.research.att.com/~bs/C++.html>) declara formalmente e implementa a linguagem C++ , como evolução ao C.

Existem diversos apontamentos em sua FAQ (http://www2.research.att.com/~bs/bs_faq.html) que podem esclarecer os fundamentos que levaram o pesquisador a criação de uma linguagem específica para OOP, em substituição ao C.

Também pode-se encontrar transcrições de Dennis M. Ritchie (<http://cm.bell-labs.com/cm/cs/who/dmr/index.html>) e Brian Kernighan (<http://www.cs.princeton.edu/~bwk/> ou <http://plan9.bell-labs.com/who/bwk/>) sobre o livro The C Programming Language (<http://plan9.bell-labs.com/cm/cs/cbook/index.html>) .

Estas linguagem foram implementadas para utilização em UNIX, o SO usado no meio academico ou de pesquisa naquela época.

Penso que uma leitura atenta dos escritos destes grandes mestres pode esclarecer muitas das afirmações feitas nesta discussão.

Obrigado,

De: Marcelo Elias Del Valle <mvallebr@gmail.com>
Para: UML-BR@yahoogrupos.com.br
Enviadas: Sábado, 11 de Dezembro de 2010 22:32:03
Assunto: [UML-BR] polimorfismo em C

Caros,

Recebi um link esses dias que achei interessante, sobre polimorfismo em C, e resolvi compartilhar com o grupo.
<http://www.thradams.com/codeblog/cpoly.htm>

Há tempos atrás eu tinha gerado uma polêmica sobre isso aqui no grupo, mas poucos devem se lembrar. Contudo, continuo acreditando em orientação a objetos mesmo em linguagens procedurais, como C.
Claro que uma linguagem de mais alto nível tornaria mais fácil a separação de preocupações, mas esse link é um exemplo de que é possível aplicar os mesmos conceitos numa linguagem como C.

Abraços,

—

Marcelo Elias Del Valle
11 6995 0065

[As partes desta mensagem que não continham texto foram removidas]

[As partes desta mensagem que não continham texto foram removidas]

— ? — ? — ? —

| [através de email](#) | [Responder através da web](#) | [Adicionar um novo tópico](#)
[Mensagens neste tópico](#) (1)

ATIVIDADE NOS ÚLTIMOS DIAS: [Novos usuários](#) 4 | [Novos arquivos](#) 1 |
[Visite seu Grupo](#)

Mais de 200.000 Vagas de Emprego! Cadastre seu Cv por 7 dias GRÁTIS!

Quer ser o próximo?Veja histórias de quem encontrou quem faltava!

Hotels.com - Las Vegas por menos de R\$ 29

YAHOO! GRUPOS
BRASIL

Trocar para: [SÃ Text](#)o, [Resenha DiÃria](#) • [Sair do grupo](#) • [Termos de uso](#)

— ? — ? — ? —

carcarah@gmail.com <carcarah@gmail.com>

Reply-To: UML-BR@yahoogrupos.com.br

To: UML-BR@yahoogrupos.com.br

Tue, Dec 14, 2010 at 4:16 PM

Uma vez vi um especialista na área falando que não existem linguagens OO e não OO, mas linguagens mais, ou menos, OO nativamente.

O fato é que o núcleo do Java que é nativo não é lá muito grande, muito do que existe nele é fruto de implementações feitas nele.

Na verdade uma pessoa pode não saber nada de OO e programar em Java totalmente procedural, sem OO. Você dirá que ela estaria programando errado, mas, será você dizendo, não o Java.

O fato é que eu entendo que programar OO é, antes de ser um recurso de uma linguagem, uma forma de desenvolver.

Até digo para você que em certas situações a solução é mesmo procedural, OO não se justifica, é matar mosquito com tiro de canhão.

Nesse ponto eu critico o OO, e até Java, pois é como o cara que tem um relógio de pulso que resiste até 200 metros de profundidade na água.

Em 14 de dezembro de 2010 12:45, Rodolpho Ugolini <rodolpho.ugolini@gmail.com> escreveu:

> Anderson,

>

> Como eu disse, não existe um consenso do que faz uma linguagem ser
> considerada OO ou não.

>

> Java, C++, C# e Smalltalk são OO, pois permitem aplicar os conceitos OO
> (encapsulamento, hierarquia e polimorfismo) *nativamente*. C, não.

>

> Agora, a implementação destes mecanismos varia de linguagem para linguagem.
> SmallTalk é mais elegante e implementa nativamente mais conceitos OO.

>

> Em 97 tive oportunidade de conversar com o James Gosling em 97, quando ele
> esteve no Brasil, sobre os tipos primitivos. A explicação foi simples:
> performance. Java deveria rodar em coisas minúsculas e limitadas.

>

> Isso não faz dela uma linguagem procedural, faz?

>

```
> Se eu tiver 2 colunas: Procedural | Orientado à Objetos
>
> Onde você colocaria Java? E "C" ?
>
> Me aponte uma referência que sustente a sua argumentação de que Java não é
> Orientado por Objetos. E se não é OO, é procedural?
>
> Abraço!
> Rodolpho
>
> 2010/12/14 Anderson Meira <anderson.duarte.meira@gmail.com>
>
> >
> >
> > Rodolpho,
> >
> > Parece incrível, mas sim, dá para questionar.
> >
> > Java, C++ e C# dão suporte a OO. Uma linguagem realmente OO é Smalltalk.
> >
> > Por exemplo, elas têm coisas que NÃO SÃO objetos, como os primitivos.
> >
> > Sds,
> > Anderson Meira
> > Sun Certified Java Programmer 6.0
> >
> > -----Mensagem original-----
> > De: UML-BR@yahoo.grupos.com.br <UML-BR%40yahoo.grupos.com.br> [mailto:
> > UML-BR@yahoo.grupos.com.br <UML-BR%40yahoo.grupos.com.br>] Em nome de
> > Rodolpho Ugolini
> > Enviada em: terça-feira, 14 de dezembro de 2010 13:11
> > Para: UML-BR@yahoo.grupos.com.br <UML-BR%40yahoo.grupos.com.br>
> > Assunto: Re: [UML-BR] Re: polimorfismo em C
> >
> >
> > Fernanda,
> >
> > Java é Orientado por Objetos. Dá pra questionar isto?
> >
> > Abraço,
> > Rodolpho
> >
> >
> >
> >
> [As partes desta mensagem que não continham texto foram removidas]
>
>
>
> -----
>
> Links do Yahoo! Grupos
>
>
>
>
--
//-----//
Posso não concordar com nenhuma das
palavras que você disser, mas defenderei
até a morte o direito de dizê-las.
```

[As partes desta mensagem que não continham texto foram removidas]

| [através de email](#) | [Responder através da web](#) | [Adicionar um novo tópico](#)
Mensagens neste tópico (18)

ATIVIDADE NOS ÚLTIMOS DIAS: [Novos usuários 4](#) | [Novos arquivos 1](#) |
[Visite seu Grupo](#)

YAHOO! GRUPOS
BRASIL

Trocar para: [SÃ Text](#)o, [Resenha DiÃria](#) • [Sair do grupo](#) • [Termos de uso](#)

Fernanda Coelho <nanda.coelho@gmail.com>
Reply-To: UML-BR@yahoogrupos.com.br
To: UML-BR@yahoogrupos.com.br

Tue, Dec 14, 2010 at 5:03 PM

Oi Rodolfo,

Sim, posso questionar. Eh possivel eu fazer um main com um if else
, com um print entre eles fechar esse main e pronto: Tenho um programinha
estruturado. Nao eh uma classe, nem um objeto, apenas faz umas coisas. Posso
"complicar" colocando um for ou um while... Java nao eh necessariamente
orientado a objeto, isso que eu quero dizer. Isso foi pergunta de uma prova
da especializacao q eu fiz na UFMG rrsrs, e fui uma das poucas pessoas q
disseram q ele nao era necessariamente OO e acertar.

Mas eh claro, ele nao foi dotado do que foi, pra desenvolver aplicacoes
estruturadas. Mas questionar eu posso sim.

Abraco

[Quoted text hidden]

> http://www.di.ubi.pt/%7Epprata/poo/POO_08_09_T08.pdf<http://www.di.ubi.pt/~pprata/poo/POO_08_09_T08.pdf>

[Quoted text hidden]

Rodolpho Ugolini <rodolpho.ugolini@gmail.com>
Reply-To: UML-BR@yahoogrupos.com.br
To: UML-BR@yahoogrupos.com.br

Tue, Dec 14, 2010 at 5:09 PM

Fernanda,

Então quer dizer que linguagens que tem "If".."elsa", "While", "For", não
são "necessariamente" Orientadas à Objetos?

[Quoted text hidden]

> > http://www.di.ubi.pt/%7Epprata/poo/POO_08_09_T08.pdf<
> http://www.di.ubi.pt/~pprata/poo/POO_08_09_T08.pdf<http://www.di.ubi.pt/%7Epprata/poo/POO_08_09_T08.pdf>

[Quoted text hidden]

Rodolpho Ugolini <rodolpho.ugolini@gmail.com>
Reply-To: UML-BR@yahoogrupos.com.br
To: UML-BR@yahoogrupos.com.br

Tue, Dec 14, 2010 at 5:54 PM

Fernanda, só pra aprofundar:

Em SmallTalk, uma linguagem reconhecidamente pura OO, o exemplo que
você descreveu ficaria:

```
=====
1 to:10 do[:iterador|

(iterador>3) ifTrue:[ conta:=conta+1.

Transcript show: (conta asString).

] ifFalse:[ x:=0 ]
].
=====
```

O código implementa exatamente o exemplo que você citou, e isto faz de SmallTalk menos orientado por objetos? É claro que não.

É possível "nivelar por baixo" e usar qualquer linguagem OO de maneira Procedural (ou funcional) num único linguagem que só "faz umas coisas". Isso não torna a linguagem menos orientada por objetos, porque toda linguagem é uma mistura de elementos OO e Procedurais (as construções básicas são sempre if, else, while/do, etc.).

Existe sim (concordo com você) código mais orientado por objeto e menos orientado por objeto. E Java é uma linguagem que permite mais "bobagens procedurais" do que SmallTalk, por exemplo (estou concordando com você).

No caso da sua prova, se a pergunta fosse "Java é orientado por objetos" eu iria errar e contestar o professor. Se a pergunta fosse "Java é uma linguagem OO pura" ou "É possível implementar um programa procedural em Java", daí concordamos e eu iria acertar.

Mas ela é sim, orientada por objetos. Isso não dá pra questionar.

Abraço!!!

[Quoted text hidden]

>> > http://www.di.ubi.pt/%7Epprata/poo/POO_08_09_T08.pdf<http://www.di.ubi.pt/~pprata/poo/POO_08_09_T08.pdf>

[Quoted text hidden]

[Quoted text hidden]

Fernanda Coelho <nanda.coelho@gmail.com>

Tue, Dec 14, 2010 at 6:23 PM

Reply-To: UML-BR@yahoogrupos.com.br

To: UML-BR@yahoogrupos.com.br

eu disse q o java nao eh puramente orientado a objeto e que posso sim implementar programas estruturados nele. digo isso pq a grande maioria diz q ele eh puramente OO. Entao, se ele aceita uma confeccao estruturada, ele nao eh puramente OO. Nao conheco o C++ mas deve ser do mesmo jeito.

[Quoted text hidden]

> http://www.di.ubi.pt/%7Epprata/poo/POO_08_09_T08.pdf<http://www.di.ubi.pt/~pprata/poo/POO_08_09_T08.pdf>

> >

> > <

> > > http://www.di.ubi.pt/%7Epprata/poo/POO_08_09_T08.pdf<http://www.di.ubi.pt/~pprata/poo/POO_08_09_T08.pdf>

> <

> > http://www.di.ubi.pt/~pprata/poo/POO_08_09_T08.pdf<

> http://www.di.ubi.pt/%7Epprata/poo/POO_08_09_T08.pdf<http://www.di.ubi.pt/~pprata/poo/POO_08_09_T08.pdf>

> >

> > >

> > > >

[Quoted text hidden]

Rodolpho Ugolini <rodolpho.ugolini@gmail.com>

Reply-To: UML-BR@yahoogrupos.com.br

To: UML-BR@yahoogrupos.com.br

Tue, Dec 14, 2010 at 7:01 PM

Fernanda,

Acho que está acontecendo confusão: Java não é considerada uma linguagem OO pura porque implementa, como já foi falado aqui, tipos primitivos que não são Objetos e também permite operações entre objetos, ao invés de mensagens.

C++ é diferente de Java (a sintaxe é parecida) porque, ao contrário de Java, permite uma programação 100% procedural.

twitter.com/rugolini

[Quoted text hidden]

Rafael Chaves <rchaves.jug@gmail.com>

Reply-To: UML-BR@yahoogrupos.com.br

To: UML-BR@yahoogrupos.com.br

Tue, Dec 14, 2010 at 7:07 PM

Vou me abster de participar da polemica (ate' porque eu estou sempre certo e voces todos sempre errados), mas, Rodolpho, quer for loop mais orientado a objetos do que esse?

Para os que nao conhecem Smalltalk, a linguagem nao tem for loops nem ifs. Tudo e' na base de mandar um mensagem a um objeto.

No exemplo do Rodolpho, ele invocou o metodo "to: do:" no objeto 1, que e' um inteiro. Esse metodo recebe dois parametros: o limite, e o bloco a ser executado (que tambem e' um objeto). Coisa similar ocorre em seguida: "(iterador > 3)" e' uma expressao booleana, que resulta num num objeto Boolean, que implementa o metodo "ifTrue:". Esse metodo so' executa o bloco passado se o boolean for True. Caso contrario, o bloco e' ignorado.

Depois de 14 anos programando em Java, rola uma lagrima so' de ver um exemplo bobo de codigo Smalltalk. Bons tempos... A melhor linguagem que ja' usei, disparado. Ruby ou Groovy nao se aproximam...

P.S. Rodolpho, acho que entendi a tua intencao - mesmo usando apenas mecanismos O.O., o codigo nao tem nada de O.O. - mas para quem nao conhece nada de Smalltalk, a ideia vai ficar perdida.

Rafael

[Quoted text hidden]

Marcio Tierno <mtierno@rocketmail.com>

Reply-To: UML-BR@yahoogrupos.com.br

To: UML-BR@yahoogrupos.com.br

Tue, Dec 14, 2010 at 7:25 PM

--- Em UML-BR@yahoogrupos.com.br, Rafael Chaves <rchaves.jug@...> escreveu
>

> Vou me abster de participar da polemica (ate' porque eu estou sempre
> certo e voces todos sempre errados)

KKKKKKKKKKKKKKKKKK!!!! Vou adotar essa!!!

MT.

| [através de email](#) | [Responder através da web](#) | [Adicionar um novo tópico](#)
Mensagens neste tópico (25)

ATIVIDADE NOS ÚLTIMOS DIAS: [Novos usuários 4](#) | [Novos arquivos 1](#) |
[Visite seu Grupo](#)

[Yahoo Encontros! Converse com milhares de pretendentes!](#)

[Mais de 200 mil Vagas de EMPREGO, em 1.600 cidades brasileiras.](#)

[Hotels.com - Las Vegas por menos de R\\$ 29](#)

YAHOO! GRUPOS
BRASIL

Trocar para: [SÃ Textos](#), [Resenha DiÃria](#) • [Sair do grupo](#) • [Termos de uso](#)

Rodolpho Ugolini <rodolpho.ugolini@gmail.com>

Tue, Dec 14, 2010 at 9:52 PM

Reply-To: UML-BR@yahoogrupos.com.br

To: UML-BR@yahoogrupos.com.br

Ótima Rafael!

É. Eu quis mostrar que dá pra criar uma "rotina procedural" em qualquer linguagem. Mesmo numa linguagem "perfeita" como SmallTalk (você deve gostar de Eiffel também né Rafael? :-))

Meu ponto foi justamente argumentar que o fato de Java não ser "OO puro" (concordando com a Fernanda) não é pelo fato de colocar ifs ou fazer loops (fiz isso em SmallTalk, usando os recursos próprios da linguagem). Mas sim de Java ter coisas não OO (os tipos primitivos) e ter operadores que não são mensagens (para as operações básicas).

Obrigado. No meio da discussão até me perdi.

Gosto demais desta lista porque sempre chegamos nos nossos limites de conhecimento.

Um abraço a todos,
Rodolpho - twitter.com/rugolini

[Quoted text hidden]

peringfc@gmail.com <peringfc@gmail.com>

Tue, Dec 14, 2010 at 8:04 PM

Reply-To: UML-BR@yahoogrupos.com.br

To: UML-BR@yahoogrupos.com.br

Eu nao acredito que estou lendo isto que java c++ nao sao linguagem oo o que entao e linguagem oo ???
Pessoal voces estao de brincadeira ??

Enviado pelo meu aparelho BlackBerry® da Vivo

[Quoted text hidden]

Fabrcio Cabral <fabriciofx@gmail.com>

Wed, Dec 15, 2010 at 9:45 AM

To: UML-BR@yahoogrupos.com.br

Olá todos,

Márcio, concordo com você em quase tudo. Só discordo desta afirmação:

"total ausência de encapsulamento". Até onde eu saiba (e por favor, me corrija se estiver errado), C possui um tipo de encapsulamento por arquivo, que é a inclusão da palavra reservada static antes das funções e variáveis globais, o que deixa estes elementos private para *o arquivo* (eles não podem ser acessados (ou executados) a partir de outros arquivos).

Atenciosamente,

2010/12/14 Marcio Tierno <mtierno@rocketmail.com>

[Quoted text hidden]

--
-fx

Rodrigo Wolschick <rodrigogui@gmail.com>

Wed, Dec 15, 2010 at 12:29 PM

Reply-To: UML-BR@yahoogrupos.com.br

To: UML-BR@yahoogrupos.com.br

Nao adianta discutir se eh OO ou NAO OO, enquanto vcs continuarem a criar objetos chamados CarroList, VendasController, MalasHandler, NotaFiscalInterface, package utils, com a classe TudoUtil.class, isso eh o que faz algo ser procedural.

No momento em que eu ver um sistema em que objetos sejam objetos, como ContaCorrente, NotaFiscal, Cliente, Banco, eu vou chamar de OO.

Caso contrario, eh orientado a procedimentos.

Obrigado,

2010/12/15 Fabrício Cabral <fabriciofx@gmail.com>

> Olá todos,

>

> Márcio, concordo com você em quase tudo. Só discordo desta afirmação:

> "total ausência de encapsulamento". Até onde eu saiba (e por favor, me
> corrija se estiver errado), C possui um tipo de encapsulamento por arquivo,
> que é a inclusão da palavra reservada static antes das funções e variáveis
> globais, o que deixa estes elementos private para *o arquivo* (eles não
> podem
> ser acessados (ou executados) a partir de outros arquivos).

>

> Atenciosamente,

>

> 2010/12/14 Marcio Tierno <mtierno@rocketmail.com>

>

> >

> >

> >

> > Fernanda

> >

> > Acho que ninguém está questionando C, nem atacando a linguagem. Eu mesmo

> > adoro C. Quando era adolescente, eu cantava de galo junto a meus amigos

> > nerds pq eu sabia programar em C. Eles me olhavam com um misto de

> > admiração

> > e inveja, além de uma ponta de ceticismo ("será que esse cara sabe mesmo

> > programar em C?").

> >

> > Como C é quase um assembler, dá pra fazer qualquer coisa com ela. Mas não

> > dá pra dizer que C seja OO ou mesmo que seja possível programar OO em C,

> > pela ausência de herança (emulável, como mostrou nosso amigo Marcelo) e

> > pela total ausência de encapsulamento - isso não dá pra emular.

> >
 > > Então, só pra reforçar: C é legal. Não, C é muuuuuito legal. É linguagem
 > de
 > > gente grande. Mas não é OO.
 > >
 > > Inclusive, um cara que foi ídolo da minha geração, o Stroustrup, criou um
 > C
 > > orientado a objetos: chama C++.
 > >
 > > []s!!
 > >
 > > MT.
 > >
 > > --- Em UML-BR@yahoogrupos.com.br <UML-BR%40yahoogrupos.com.br>, Fernanda

[Quoted text hidden]

> > > (Fonte: http://en.wikipedia.org/wiki/Object-oriented_programming)
 > > > "Polymorphism is a process in which a class has all the state and
 > > > behavior of another class."
 > > > Dada essa definição, concordam que ela é plenamente satisfeita pela
 > > > programação em C? Onde está escrito que, se a linguagem não
 > implementar
 > > > polimorfismo de inclusão você não pode implementar um programa em OO
 > com
 > > > ela? Até a versão 1.4, java não implementava polimorfismo
 > paramétrico.
 > > Não
 > > > dava para escrever em OO com ela?
 > > > Ponto 2: não estou plenamente convencido de que não é possível
 > > > implementar polimorfismo de inclusão em C, só emular. Pela definição
 > de
 > > > polimorfismo de inclusão, temos:
 > > > (Fonte: http://www.di.ubi.pt/~pprata/poo/POO_08_09_T08.pdf)

[Quoted text hidden]

> > > > --- Em UML-BR@yahoogrupos.com.br <UML-BR%40yahoogrupos.com.br>
 > > <UML-BR%
 > > 40yahoogrupos.com.br>,
 > > > Marcelo
 > > > > Elias Del Valle <mvallebr@> escreveu
 > > > > >
 > > > > >
 > > > > > Caros,
 > > > > >
 > > > > > Recebi um link esses dias que achei interessante, sobre
 > > polimorfismo
 > > > em
 > > > > C, e resolvi compartilhar com o grupo.
 > > > > > <http://www.thradams.com/codeblog/cpoly.htm>
 > > > > >

[Quoted text hidden]

Rodrigo Guilherme Wolschick
 Porto Alegre/Venâncio Aires - RS - Brazil

—

<http://blog.rodrigow.com/> <<http://blog.rodrigow.com/>>*
 [http://twitter.com/rodrigo**gw](http://twitter.com/rodrigo****gw)* <http://twitter.com/rodrigo****gw>*
 [http://flickr.com/photos/r**odrigow](http://flickr.com/photos/r****odrigow)* <<http://flickr.com/photos/rodrigow>>
 *
 *

[As partes desta mensagem que não continham texto foram removidas]

[Links do Yahoo! Grupos](#)

[Quoted text hidden]

SIZA, Rafael <rafaelsiza@gmail.com>
Reply-To: UML-BR@yahoogrupos.com.br
To: UML-BR@yahoogrupos.com.br

Wed, Dec 15, 2010 at 12:51 PM

Prezados,

Meus 0,10 cents:

Não sei se todos criam (ou já criaram) objetos como o Rodrigo mencionou, mas concordo com ele. Há programadores que insistem apenas no "mão na massa" sem a preocupação da aplicação do conceito.

[]'s

--
Rafael Siza
rafaelsiza@gmail.com

[Quoted text hidden]

> > > > (Fonte: http://www.di.ubi.pt/~pprata/poo/POO_08_09_T08.pdf<http://www.di.ubi.pt/%7Epprata/poo/POO_08_09_T08.pdf>

[Quoted text hidden]

Rodrigo Wolschick <rodrigogui@gmail.com>
Reply-To: UML-BR@yahoogrupos.com.br
To: UML-BR@yahoogrupos.com.br

Wed, Dec 15, 2010 at 1:23 PM

Rafael,

diria mais: sem o dominio da aplicacao que estao construindo. Isso volta ao tempo em que precisaríamos escrever Casos de Uso, entao um documento tecnico, entao o programador vai lah e cria um procedimento para que o caso de uso passe em alguns cenarios de teste.

:)

2010/12/15 SIZA, Rafael <rafaelsiza@gmail.com>

[Quoted text hidden]

Marcio Tierno <mtierno@rocketmail.com>
Reply-To: UML-BR@yahoogrupos.com.br
To: UML-BR@yahoogrupos.com.br

Wed, Dec 15, 2010 at 1:29 PM

Fabrcio

Static é um identificador que declara uma variável como pertencente ao meta-nível.

Em outras palavras, static define uma variável cujo tempo de vida coincide com o do programa.

Assim, se você declara 2 variáveis dentro de uma função, uma estática e outra não estática, e atribuir às duas o valor do primeiro argumento da função, na próxima vez que você chamar a função a variável não estática estará "zerada", enquanto a variável estática estará *exatamente* com o mesmo valor que você atribuiu a ela na última vez.

Em sistemas multi-thread, variáveis estáticas são áreas compartilhadas de memória e devem ser protegidas contra acesso simultâneo.

[]s

MT.

— Em UML-BR@yahoogrupos.com.br, Fabrício Cabral <fabriciofx@...> escreveu

>

> Olá todos,

>

> Márcio, concordo com você em quase tudo. Só discordo desta afirmação:

> "total ausência de encapsulamento". Até onde eu saiba (e por favor, me

> corrija se estiver errado), C possui um tipo de encapsulamento por arquivo,

> que é a inclusão da palavra reservada static antes das funções e variáveis

> globais, o que deixa estes elementos private para *o arquivo* (eles não

> podem

> ser acessados (ou executados) a partir de outros arquivos).

>

> Atenciosamente,

>

> 2010/12/14 Marcio Tierno <mtierno@...>

>

>>

>>

>>

>> Fernanda

>>

>> Acho que ninguém está questionando C, nem atacando a linguagem. Eu mesmo

>> adoro C. Quando era adolescente, eu cantava de galo junto a meus amigos

>> nerds pq eu sabia programar em C. Eles me olhavam com um misto de admiração

>> e inveja, além de uma ponta de ceticismo ("será que esse cara sabe mesmo

>> programar em C?").

>>

>> Como C é quase um assembler, dá pra fazer qualquer coisa com ela. Mas não

>> dá pra dizer que C seja OO ou mesmo que seja possível programar OO em C,

>> pela ausência de herança (emulável, como mostrou nosso amigo Marcelo) e

>> pela total ausência de encapsulamento - isso não dá pra emular.

>>

>> Então, só pra reforçar: C é legal. Não, C é muuuuuuito legal. É linguagem de

>> gente grande. Mas não é OO.

>>

>> Inclusive, um cara que foi ídolo da minha geração, o Stroustrup, criou um C

>> orientado a objetos: chama C++.

>>

>> []s!!

>>

>> MT.

>>

>> — Em UML-BR@yahoogrupos.com.br <UML-BR%40yahoogrupos.com.br>, Fernanda

>> Coelho <nanda.coelho@...> escreveu

>>

>>>

>>> Eu programo em C a mais de 10 anos. E o fato dela nao ser uma linguagem

>>> OO

>>> nao a limita na implementacao. Tudo eh questao de dominio de conceitos e

>>> projetos adequados. Java nao eh uma linguagem OO ela permite q se

>>> implemente

>>> OO mas tb permite a forma estrutural. O mesmo se aplica a linguagem C. E

>>> ainda acho C uma linguagem muitissimo eficaz, dada a sua capacidade de

>>> manipular os extremos da maquina. Existem programas que se feitos em

>>> Java,

>>> jamais chegaram de uma performance conseguida pelo mesmo programa em C,

>>> mesmo q esteja fazendo OO em C. Concordo contigo Marcelo.

>>>

>>> Abraco

>>>

> > > 2010/12/14 Marcelo Elias Del Valle <mvallebr@>
> >
> > >

[Quoted text hidden]

> > > > > escreveu:
> > > >
> > > > Olá todos,
> > > >
> > > > concordo com o MT. Aliás, esta era uma opinião errada que eu tinha
> > até
> > > bem
> > > > pouco tempo atrás: onde começa e termina os conceitos da programação
> > > > orientada a objetos?
> > > >
> > > > Se você der uma olhada nos aspectos da POO, verá que a linguagem C
> > > > não aborda todos os conceitos esperados para uma linguagem voltada a
> > > > OO. No entanto, é possível fazer uma programa *baseada* (similar) a
> > > > POO usando C, mas deve-se ter a ressalva de que não é a mesma coisa.
> > > >
> > > > E só por curiosidade, existem muitos códigos em C que usam desta
> > > > habilidade.
> > > > De cabeça mesmo, cito o kernel do Linux e a biblioteca gráfica GTK+
> > (que
> > > > usa muitos mais recursos além destes).
> > > >
> > > > Atenciosamente,
> > > >
> > > > 2010/12/12 Marcio Tierno <mtierno@>
> >
> > > >
> > > > >
> > > > >
> > > > >
> > > > >

[Quoted text hidden]

> > > > > > --- Em UML-BR@yahoogrupos.com.br <UML-BR%40yahoogrupos.com.br><UML-BR%

[Quoted text hidden]

_____'_'____

| [através de email](#) | [Responder através da web](#) | [Adicionar um novo tópico](#)
[Mensagens neste tópico \(32\)](#)

ATIVIDADE NOS ÚLTIMOS DIAS: [Novos usuários 5](#) | [Novos arquivos 1](#) |
[Visite seu Grupo](#)

YAHOO! GRUPOS
BRASIL

Trocar para: [SÃº Texto](#), [Resenha DiÃ¡ria](#) • [Sair do grupo](#) • [Termos de uso](#)

_____'_'____

Fernanda Coelho <nanda.coelho@gmail.com>
Reply-To: UML-BR@yahoogrupos.com.br
To: UML-BR@yahoogrupos.com.br

Wed, Dec 15, 2010 at 10:54 PM

MT,
hj vamos discutir como gente grande! Enfim algo q eu possa discutir contigo!!!! C nao eh OO, eu nao disse isso. Eh possivel aplicar conceitos de OO em programacao C, talvez nao todos, mas muita coisa se aproveita e pq nao posso aproveitar bons conceitos em C? Eu posso encapsular em C sim, os compiladores mais atuais nao deixam passar duas variaveis com o mesmo nome com definicoes de escopo diferente. A heranca, nunca tentei, mas deve ser dureza como tudo em C. O q eu disse eh isso: Java nao eh puramente OO, se tiver q classificar, ela eh as duas coisas OO e estrutural. C eh estrutural,

mas da pra aproveitar conceitos OO aliados a um bom projeto de programacao. C++ nao posso opniar... nao conheco. E a unica linguagem q sei ser OO eh o smaltalk, mas isso me disseram la na pos graduacao. Nao conheco, entao nao vou entrar no merito.

Acho q agora me fiz entender, ou nao?

[]'s
Fernanda

2010/12/14 Marcio Tierno <mtierno@rocketmail.com>

>
>
>
> Fernanda
>
> Acho que ninguém está questionando C, nem atacando a linguagem. Eu mesmo
> adoro C. Quando era adolescente, eu cantava de galo junto a meus amigos
> nerds pq eu sabia programar em C. Eles me olhavam com um misto de admiração
> e inveja, além de uma ponta de ceticismo ("será que esse cara sabe mesmo
> programar em C?").
>
> Como C é quase um assembler, dá pra fazer qualquer coisa com ela. Mas não
> dá pra dizer que C seja OO ou mesmo que seja possível programar OO em C,
> pela ausência de herança (emulável, como mostrou nosso amigo Marcelo) e
> pela total ausência de encapsulamento - isso não dá pra emular.
>
> Então, só pra reforçar: C é legal. Não, C é muuuuuuito legal. É linguagem de
> gente grande. Mas não é OO.
>
> Inclusive, um cara que foi ídolo da minha geração, o Stroustrup, criou um C
> orientado a objetos: chama C++.
>
> []s!!
>
> MT.
>
> --- Em UML-BR@yahoogrupos.com.br <UML-BR%40yahoogrupos.com.br>, Fernanda
[Quoted text hidden]
> > > (Fonte: http://en.wikipedia.org/wiki/Object-oriented_programming)
> > > ""Polymorphism is a process in which a class has all the state and
> > > behavior of another class.""
> > > Dada essa definição, concordam que ela é plenamente satisfeita pela
> > > programação em C? Onde está escrito que, se a linguagem não implementar
> > > polimorfismo de inclusão você não pode implementar um programa em OO
> > com
> > > ela? Até a versão 1.4, java não implementava polimorfismo paramétrico.
> > Não
> > > dava para escrever em OO com ela?
> > > Ponto 2: não estou plenamente convencido de que não é possível
> > > implementar polimorfismo de inclusão em C, só emular. Pela definição de
> > > polimorfismo de inclusão, temos:
> > > (Fonte: http://www.di.ubi.pt/~pprata/poo/POO_08_09_T08.pdf)
[Quoted text hidden]
> > > > --- Em UML-BR@yahoogrupos.com.br <UML-BR%40yahoogrupos.com.br><UML-BR%40yahoogrupos.com.br>,
> > > Marcelo
> > > > Elias Del Valle <mvallebr@> escreveu
> > > >
> > > >
> > > > > Caros,
> > > > >
> > > > > Recebi um link esses dias que achei interessante, sobre
> > polimorfismo

> > > em
> > > > C, e resolvi compartilhar com o grupo.
> > > > > <http://www.thradams.com/codeblog/cpoly.htm>
> > > > >

[Quoted text hidden]

[Quoted text hidden]

Fernanda Coelho <nanda.coelho@gmail.com>

Wed, Dec 15, 2010 at 11:22 PM

Reply-To: UML-BR@yahoogrupos.com.br

To: UML-BR@yahoogrupos.com.br

Rodolfo, eu te entendo qdo falas dos tipos primitivos. Nao estou dizendo q Java nao eh OO, ela eh. Mas permite estrutural, alias eu ja fiz uma manutencao em programa java q tive q fazer por demanda do urgente em cima do mais importante e o q foi mais facil pra mim? Pensar estrutural. Fiz um programa estrutural, numa linguagem OO sem aproveitar os recursos dela. C eh estrutural. O negocio de C eh aproveitar o melhor de outros.

Se eu partir de um projeto zero, posso dizer pra se usar tecnologia Java ou tecnologia C, concorda? Vai depender do projeto. E de repente, as preocupacoes podem ser limitadas a tal ponto q uma parte seja C e outra Java, concorda?

Entao, C pode aproveitar coisas boas dos conceitos durante a implementacao do projeto, mas nao deve assumir todo o projeto se uma parte dele fica melhor se executado em Java, e neste caso, o programador vai usar tipos primitivos, mas fazendo classes, objetos, herancas e q tais. E vice-versa: Onde C eh usado melhor, Java nao deve dar pitaco, pq ele eh uma linguagem pesada, ocupa memoria d+ e o gerenciamento automatico dele, pelo q ja vi, nao eh 100% onde eu preciso de precisao.

Acho q me fiz entender, ou nao?

[]'s

[Quoted text hidden]

[Quoted text hidden]

Marcio Tierno <mtierno@rocketmail.com>

Thu, Dec 16, 2010 at 9:04 AM

Reply-To: UML-BR@yahoogrupos.com.br

To: UML-BR@yahoogrupos.com.br

Fernanda

Falamos de coisas diferentes. Pode-se aplicar conceitos de construção civil, da Arte da Guerra do Sun Tzu e até do Feng Shui em programação, seja em C ou em Basic. Também pode-se aplicar (ou tentar bastante) conceitos de OO em qualquer linguagem. Daí a dizer que se está programando OO nessas linguagens vai uma grande distância.

C não é OO. Ponto. Não dá para encapsular como pede OO (information hiding). E não dá para aplicar herança. Ao menos de uma forma produtiva. Como eu disse em meu post anterior, para implementar essas coisas em C, vc vai gastar mais tempo e esforço do que para fazer as funcionalidades do sistema em si - e aí não faz sentido.

Sobre Java não ser OO, veja a questão da ortogonalidade de paradigmas. Linguagens podem ser OO e produtivas, OO e funcionais, OO e reflexivas (caso de SmallTalk), etc.

C é procedural (não estrutural). Java é OO *E* procedural. Smalltalk é OO *E* reflexiva. E assim por diante.

BTW, Simula 67, a PRIMEIRA linguagem OO, também era procedural.

E, querem saber? Esse assunto foi resolvido lá por 1983, mais ou menos. As definições estão lá.

[]s

MT.

— Em UML-BR@yahoogrupos.com.br, Fernanda Coelho <nanda.coelho@...> escreveu
>

> MT,
> hj vamos discutir como gente grande! Enfim algo q eu possa discutir
> contigo!!!! C nao eh OO, eu nao disse isso. Eh possivel aplicar conceitos de
> OO em programacao C, talvez nao todos, mas muita coisa se aproveita e pq nao
> posso aproveitar bons conceitos em C? Eu posso encapsular em C sim, os
> compiladores mais atuais nao deixam passar duas variaveis com o mesmo nome
> com definicoes de escopo diferente. A heranca, nunca tentei, mas deve ser
> dureza como tudo em C. O q eu disse eh isso: Java nao eh puramente OO, se
> tiver q classificar, ela eh as duas coisas OO e estrutural. C eh estrutural,
> mas da pra aproveitar conceitos OO aliados a um bom projeto de programacao.
> C++ nao posso opniar... nao conheco. E a unica linguagem q sei ser OO eh o
> smaltalk, mas isso me disseram la na pos graduacao. Nao conheco, entao nao
> vou entrar no merito.

>
> Acho q agora me fiz entender, ou nao?
>

> []'s
> Fernanda
>

> 2010/12/14 Marcio Tierno <mtierno@...>
>

>
>
>
>

> > Fernanda
> >

> > Acho que ninguém está questionando C, nem atacando a linguagem. Eu mesmo
> > adoro C. Quando era adolescente, eu cantava de galo junto a meus amigos
> > nerds pq eu sabia programar em C. Eles me olhavam com um misto de admiração
> > e inveja, além de uma ponta de ceticismo ("será que esse cara sabe mesmo
> > programar em C?").
> >

> > Como C é quase um assembler, dá pra fazer qualquer coisa com ela. Mas não
> > dá pra dizer que C seja OO ou mesmo que seja possível programar OO em C,
> > pela ausência de herança (emulável, como mostrou nosso amigo Marcelo) e
> > pela total ausência de encapsulamento - isso não dá pra emular.
> >

> > Então, só pra reforçar: C é legal. Não, C é muuuuuuito legal. É linguagem de
> > gente grande. Mas não é OO.
> >

> > Inclusive, um cara que foi ídolo da minha geração, o Stroustrup, criou um C
> > orientado a objetos: chama C++.

> >
> > []s!!
> >

> > MT.
> >

> > — Em UML-BR@yahoogrupos.com.br <UML-BR%40yahoogrupos.com.br>, Fernanda
> > Coelho <nanda.coelho@...> escreveu
> >

> > >

> > > Eu programo em C a mais de 10 anos. E o fato dela nao ser uma linguagem
> > > OO

> > > nao a limita na implementacao. Tudo eh questao de dominio de conceitos e
> > > projetos adequados. Java nao eh uma linguagem OO ela permite q se
> > > implemente

> > > OO mas tb permite a forma estrutural. O mesmo se aplica a linguagem C. E
> > > ainda acho C uma linguagem muitissimo eficaz, dada a sua capacidade de

> > > manipular os extremos da maquina. Existem programas que se feitos em
 > > Java,
 > > > jamais chegaram de uma performance conseguida pelo mesmo programa em C,
 > > > mesmo q esteja fazendo OO em C. Concordo contigo Marcelo.
 > > >
 > > > Abraco
 > > >
 > > > 2010/12/14 Marcelo Elias Del Valle <mvallebr@>
 > >
 > > >

[Quoted text hidden]

> > > > (Fonte: http://en.wikipedia.org/wiki/Object-oriented_programming)
 > > > > "**Polymorphism is a process in which a class has all the state and
 > > > > behavior of another class.**"
 > > > > Dada essa definição, concordam que ela é plenamente satisfeita pela
 > > > > programação em C? Onde está escrito que, se a linguagem não implementar
 > > > > polimorfismo de inclusão você não pode implementar um programa em OO
 > > > com
 > > > > ela? Até a versão 1.4, java não implementava polimorfismo paramétrico.
 > > Não
 > > > > dava para escrever em OO com ela?
 > > > > Ponto 2: não estou plenamente convencido de que não é possível
 > > > > implementar polimorfismo de inclusão em C, só emular. Pela definição de
 > > > > polimorfismo de inclusão, temos:
 > > > > (Fonte: http://www.di.ubi.pt/~pprata/poo/POO_08_09_T08.pdf)
 > > > > "**3.1) Polimorfismo de inclusão*
 > > > > * Uma função definida num determinado tipo pode também operar todos os
 > > seus
 > > > > *
 > > > > *subtipos. *
 > > > > *(Resulta directamente do mecanismo de herança – uma operação definida
 > > na
 > > > > classe *
 > > > > *base é também aplicável aos objectos de todas as subclasses)**"
 > > > > Essa definição não é satisfeita com a implementação em C, dadas as
 > > > > definições de tipos e subtipos?
 > > > >
 > > > > Abraços,
 > > > > Marcelo.
 > > > >
 > > > > Em 13 de dezembro de 2010 12:46, Fabrício Cabral <fabriciofx@>
 > >
 > > > > > escreveu:
 > > > >
 > > > > > Olá todos,
 > > > > >
 > > > > > concordo com o MT. Aliás, esta era uma opinião errada que eu tinha
 > > até
 > > > > > bem
 > > > > > pouco tempo atrás: onde começa e termina os conceitos da programação
 > > > > > orientada a objetos?
 > > > > >
 > > > > > Se você der uma olhada nos aspectos da POO, verá que a linguagem C
 > > > > > não aborda todos os conceitos esperados para uma linguagem voltada a
 > > > > > OO. No entanto, é possível fazer uma programa *baseada* (similar) a
 > > > > > POO usando C, mas deve-se ter a ressalva de que não é a mesma coisa.
 > > > > >
 > > > > > E só por curiosidade, existem muitos códigos em C que usam desta
 > > > > > habilidade.
 > > > > > De cabeça mesmo, cito o kernel do Linux e a biblioteca gráfica GTK+
 > > (que
 > > > > > usa muitos mais recursos além destes).
 > > > > >
 > > > > > Atenciosamente,

> > > > >
> > > > > 2010/12/12 Marcio Tierno <mtiemo@>
> >
> > > > >
> > > > > >
> > > > > >
> > > > > >
> > > > > >

[Quoted text hidden]

> > > > > --- Em UML-BR@yahoogrupos.com.br <UML-BR%40yahoogrupos.com.br><UML-BR%40yahoogrupos.com.br>,
> > > > > Marcelo
> > > > > Elias Del Valle <mvallebr@> escreveu
> > > > > >
> > > > > > >
> > > > > > Caros,
> > > > > > >
> > > > > > Recebi um link esses dias que achei interessante, sobre
> > polimorfismo
> > > > em
> > > > > C, e resolvi compartilhar com o grupo.
> > > > > > <http://www.thradams.com/codeblog/cpoly.htm>

[Quoted text hidden]

| [através de email](#) | [Responder através da web](#) | [Adicionar um novo tópico](#)
[Mensagens neste tópico \(38\)](#)

ATIVIDADE NOS ÚLTIMOS DIAS: [Novos usuários 5](#) | [Novos arquivos 1](#) |
[Visite seu Grupo](#)

[Está sozinho? Aqui você vai encontrar quem está faltando!](#)

[Procurando Um Novo EMPREGO Em 2010? Faça Já Uma Busca de Vagas Aqui!](#)

[Hotels.com - Reveillon em Buenos Aires a partir de R\\$ 99](#)



YAHOO! GRUPOS
BRASIL

Trocar para: [SÃº Texto](#), [Resenha DiÃ¡ria](#) • [Sair do grupo](#) • [Termos de uso](#)

Fabrcio Cabral <fabriciofx@gmail.com>
To: UML-BR@yahoogrupos.com.br

Thu, Dec 16, 2010 at 11:18 AM

Márcio,

eu conheço a palavra reservada static e seu uso para variáveis na linguagem C.

Acontece que eu posso usar esta palavra *também* na declaração de funções *antes* do tipo de retorno, a fim de ter um encapsulamento em nível de arquivo. Por exemplo:

```
static int funcao_privada(int a, int b)
{
    ...
}
```

Declara uma função chamada funcao_privada() que possui como argumentos dois inteiros (a e b), retorna um inteiro e ela é "privada" em nível de arquivo

(pela presença da palavra reservada static antes do int do retorno).

Att.

2010/12/15 Marcio Tierno <mtierno@rocketmail.com>

[Quoted text hidden]

--
-fx

Rodrigo Wolschick <rodrigogui@gmail.com>

Wed, Dec 15, 2010 at 11:19 PM

Reply-To: UML-BR@yahoogrupos.com.br

To: UML-BR@yahoogrupos.com.br

Ola Fernanda.

Nao gosto de discutir esse negocio de conceito, mas como tenho um tempo sobrando, vamos lah:

""Eh possivel aplicar conceitos de OO em programacao C, talvez nao todos, mas muita coisa se aproveita e pq nao posso aproveitar bons conceitos em C?

Eu posso encapsular em C sim""

Conceito de Encapsulamento: (

http://en.wikipedia.org/wiki/Object-oriented_programming#Encapsulation):

""Encapsulation conceals the functional details of a class from objects that send messages to it.""

Como se faz uma classe em C? Se tu nao tens classe, nao tem encapsulamento, ou seja, encapsulamento soh existe quando tu tiver uma classe. Se tu consegues fazer algo parecido com encapsulamento em C, nao eh encapsulamento de OO, pode ser qualquer outra coisa, mas nao encapsulamento de OO pq para existir encapsulamento de OO precisa ter uma classe assim como diz o wikipedia.

Da mesma pagina do wikipedia:

Languages designed mainly for OO programming, but with some procedural elements. Examples: C++ <<http://en.wikipedia.org/wiki/C%2B%2B>>, C#<[http://en.wikipedia.org/wiki/C_Sharp_\(programming_language\)](http://en.wikipedia.org/wiki/C_Sharp_(programming_language))>, Java <[http://en.wikipedia.org/wiki/Java_\(programming_language\)](http://en.wikipedia.org/wiki/Java_(programming_language))>, Python<[http://en.wikipedia.org/wiki/Python_\(programming_language\)](http://en.wikipedia.org/wiki/Python_(programming_language))>

Abracos,

[Quoted text hidden]

--
Rodrigo Guilherme Wolschick
Porto Alegre/Venâncio Aires - RS - Brazil

--
<http://blog.rodrigow.com/> <<http://blog.rodrigow.com/>> *
[http://twitter.com/rodrigo**gw](http://twitter.com/rodrigo****gw)* <<http://twitter.com/rodrigogw>> *
[http://flickr.com/photos/r**odrigow](http://flickr.com/photos/r****odrigow)* <<http://flickr.com/photos/rodrigow>> *
*
*

[Quoted text hidden]

Marcelo Andrade <mfandrade@gmail.com>

Thu, Dec 16, 2010 at 11:14 PM

Reply-To: UML-BR@yahoogrupos.com.br

To: UML-BR@yahoogrupos.com.br

2010/12/14 Marcio Tierno <mtierno@rocketmail.com>:
> (..)

Gostaria de contribuir mais com este tópico, mas só para enfatizar o que já vi que alguns quiseram dizer:

Programação Orientada a Objetos (POO) <>
Análise Orientada a Objetos (AOO) <>
Linguagem Orientada a Objetos

Atts.

—
MARCELO F ANDRADE
Belem, Amazonia, Brazil

"I took the red pill"

_____'_'_____

| [através de email](#) | [Responder através da web](#) | [Adicionar um novo tópico](#)
[Mensagens neste tópico \(55\)](#)

ATIVIDADE NOS ÚLTIMOS DIAS: [Novos usuários 5](#) | [Novos arquivos 1](#) |
[Visite seu Grupo](#)

Está sozinho? Aqui você vai encontrar quem está faltando!

Faça uma Busca por um Novo EMPREGO Aqui! São mais de 200.000 vagas!

Hotels.com - Reveillon em Buenos Aires a partir de R\$ 99

☒

YAHOO! GRUPOS
BRASIL

Trocar para: [SÃ³ Texto](#), [Resenha DiÃ¡ria](#) • [Sair do grupo](#) • [Termos de uso](#)

_____'_'_____