



Fabrício Cabral <fabriciofx@gmail.com>

[dotnetarchitects] Nomenclaturas em Orientação a Objetos23 messages

Acaz Souza Pereira <acazsouza@gmail.com>

Thu, Jul 19, 2012 at 12:12 AM

Reply-To: dotnetarchitects@googlegroups.com

To: ".Net Architects" <dotnetarchitects@googlegroups.com>

Essa semana achei este post "[Your coding conventions are hurting you](#)" e achei bem interessante. Gostaria de ler a opinião do pessoal aqui do grupo também.

Ele critica a nomenclatura usada hoje em dia pela maioria (leia-se gurus) e acho que tenho que concordar.

Venho achando OOP um paradigma muito "difícil" pois ele parece ser algo muito fácil de ser mal interpretado, é difícil se ver um código OOP bem feito. Quando digo bem feito, digo de fácil interpretação até para leigos. Geralmente se vê códigos com over-engineering e com nomes complicados, as vezes nos seres humanos gostamos de dificultar as coisas fáceis para que sentimos mais valor no que estamos fazendo, eu li uma vez que isso é um viés cognitivo e nessa nossa área é muito fácil deixar as coisas mais difíceis. Seria bom deixarmos o trabalho difícil para o que é realmente difícil.

É fato ver dúvidas de modelagem que fazem pouca ou nenhuma diferença na solução final. Perguntam como se existisse uma resposta exata. Isso é um trabalho subjetivo, só iremos saber se tal modelagem é boa ou não de fato quando os problemas (que não sabemos quais) aparecerem e esses problemas são únicos para cada projeto e para cada equipe. Tem uma apresentação do Nat Pryce (autor do "Growing Object-Oriented Software, Guided by Tests" e desenvolvedor da biblioteca jMock) no InfoQ sobre o assunto: [Stop Refactoring!](#)

Enfim, é o primeiro artigo o motivo do tópico.

—

[Acaz Souza Pereira]MSN/GTalk: acazsouza@gmail.com

Skype: acazsouza

Cel: (31) 8706-4103

twitter.com/acazsouzaacazsouza.com.br

—

Você recebeu esta mensagem porque faz parte do grupo .Net Architects hospedado no Google Groups.

Para postar envie uma mensagem para dotnetarchitects@googlegroups.com

Para sair do grupo envie uma mensagem para dotnetarchitects+unsubscribe@googlegroups.com

Para mais opções visite o grupo em <http://groups.google.com/group/dotnetarchitects?hl=pt-br>

Thiago Silva <thiagosb13@gmail.com>

Thu, Jul 19, 2012 at 4:43 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Essa questão de nomenclatura realmente é complicado.

Mas acho que os 'gurus' estão mudando o discurso à medida que o DDD ganha força.

No projeto que trabalho, nomenclatura é uma coisa muito importante, fugimos totalmente dos sufixos e prefixos, salvo raríssimas exceções("i" prefixo de interfaces e "Exception" sufixo de exceções). Mas não é tarefa fácil.

Abraço,

2012/7/19 Acaz Souza Pereira <acazsouza@gmail.com>

Essa semana achei este post "Your coding conventions are hurting you" e achei bem interessante. Gostaria de ler a opinião do pessoal aqui do grupo também.

Ele critica a nomenclatura usada hoje em dia pela maioria (leia-se gurus) e acho que tenho que concordar.

Venho achando OOP um paradigma muito "difícil" pois ele parece ser algo muito fácil de ser mal interpretado, é difícil se ver um código OOP bem feito. Quando digo bem feito, digo de fácil interpretação até para leigos. Geralmente se vê códigos com over-engineering e com nomes complicados, as vezes nos seres humanos gostamos de dificultar as coisas fáceis para que sentimos mais valor no que estamos fazendo, eu li uma vez que isso é um viés cognitivo e nessa nossa área é muito fácil deixar as coisas mais difíceis. Seria bom deixarmos o trabalho difícil para o que é realmente difícil.

É fato ver dúvidas de modelagem que fazem pouca ou nenhuma diferença na solução final. Perguntam como se existisse uma resposta exata. Isso é um trabalho subjetivo, só iremos saber se tal modelagem é boa ou não de fato quando os problemas (que não sabemos quais) aparecerem e esses problemas são únicos para cada projeto e para cada equipe. Tem uma apresentação do Nat Pryce (autor do "Growing Object-Oriented Software, Guided by Tests" e desenvolvedor da biblioteca jMock) no InfoQ sobre o assunto: [Stop Refactoring!](#)

Enfim, é o primeiro artigo o motivo do tópico.

--

[Acaz Souza Pereira]

MSN/GTalk: acazsouza@gmail.com

Skype: acazsouza

Cel: (31) 8706-4103

twitter.com/acazsouza

acazsouza.com.br

--

Você recebeu esta mensagem porque faz parte do grupo .Net Architects hospedado no Google Groups.

Para postar envie uma mensagem para dotnetarchitects@googlegroups.com

Para sair do grupo envie uma mensagem para dotnetarchitects+unsubscribe@googlegroups.com

Para mais opções visite o grupo em <http://groups.google.com/group/dotnetarchitects?hl=pt-br>

[Quoted text hidden]

JR <agcjuniior@gmail.com>

Sat, Jul 21, 2012 at 9:46 AM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Acaz,

Gostei muito do post. Eu também sinto que existe muita dificuldade em se "dar nomes aos bois". Não encontrar um nome adequado para uma classe, interface, propriedade, etc já é uma pista de que algo não está muito claro em nosso código.

É um assunto filosófico. Hoje em dia não temos tempo nem para "pensar".

[]s

Jr

|

[Quoted text hidden]

Acaz Souza Pereira <acazsouza@gmail.com>

Sat, Jul 21, 2012 at 12:42 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Realmente Thiago, evitar algumas convenções no estágio em que já estamos hoje iria complicar mais ainda do que ajudar. Principalmente a prefixo "I" em interfaces .Net, mesmo não sendo o ideal, já se tornou um padrão.

DDD reforça isso e existe também o [Subject-oriented programming](#) que eu acho que tenta dar a visão correta do que é OOP e diz que estado e comportamento não são intrínsecos do objeto em si, mas são fornecidos dependendo da percepção (ponto de vista) do objeto. É tudo uma questão de semântica.

[Quoted text hidden]

Cel: (31) 8706-4103

twitter.com/acazsouza

acazsouza.com.br

[Quoted text hidden]

Cezar Augusto Guimaraes Neto <cezar.alevatto@gmail.com>

Sun, Jul 22, 2012 at 11:33 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Acaz,

legal o post. A verdade no que ele fala é que na maioria das vezes existe preguiça do desenvolvedor para achar um bom nome. Outra coisa que concordo é que péssimas escolhas propagam. :) Práticas ruins propagam muito mais fácil do que boas práticas. :)

[]s

Cezar.

www.cezarguimaraes.com

[Quoted text hidden]

Para sair do grupo envie uma mensagem para dotnetarchitects+unsubscribe@googlegroups.com

Para mais opções visite o grupo em <http://groups.google.com/group/dotnetarchitects?hl=pt-br>

[Quoted text hidden]

edmilson hora <edmilson_hora@yahoo.com.br>

Mon, Jul 23, 2012 at 12:15 PM

Reply-To: dotnetarchitects@googlegroups.com

To: "dotnetarchitects@googlegroups.com" <dotnetarchitects@googlegroups.com>

Não concordo que o desenvolvedor tenha preguiça. Quem Já leu o Clean Code (Código Limpo) o próprio autor ressalta que definir bons nomes para módulos, classes e membros é uma tarefa que requer pensar bastante.

Em minha humilde opinião esse tipo de reflexão do autor do post sobre o assunto é levar o tema a um patamar desnecessário. Quem vai ler o código tem que entender com brevidade o que esta acontecendo, se tem um prefixo ou sufixo que indica melhor do que se trata o membro, o que há de tão ruim assim?

Acredito que a maioria aqui entende um valor de um código limpo, agora perturbar porque nomeia as Interfaces iniciando com "I", acho que é um pouco de exegero.

De: Cezar Augusto Guimaraes Neto <cezar.alevatto@gmail.com>

Para: dotnetarchitects@googlegroups.com

Enviadas: Domingo, 22 de Julho de 2012 23:33

Assunto: Re: [dotnetarchitects] Nomenclaturas em Orientação a Objetos

Acáz,

legal o post. A verdade no que ele fala é que na maioria das vezes existe preguiça do desenvolvedor para achar um bom nome. Outra coisa que concordo é que péssimas escolhas propagam. :) Práticas ruins propagam muito mais fácil do que boas práticas. :)

[]s

Cezar.

www.cezarguimaraes.com

On Jul 21, 2012, at 8:42 AM, Acáz Souza Pereira wrote:

Realmente Thiago, evitar algumas convenções no estágio em que já estamos hoje iria complicar mais ainda do que ajudar. Principalmente a prefixo "I" em interfaces .Net, mesmo não sendo o ideal, já se tornou um padrão.

DDD reforça isso e existe também o [Subject-oriented programming](#) que eu acho que tenta dar a visão correta do que é OOP e diz que estado e comportamento não são intrínsecos do objeto em si, mas são fornecidos dependendo da percepção (ponto de vista) do objeto. É tudo uma questão de semântica.

Em 19 de julho de 2012 16:43, Thiago Silva <thiagosb13@gmail.com> escreveu:

Essa questão de nomenclatura realmente é complicado.

Mas acho que os 'gurus' estão mudando o discurso à medida que o DDD ganha força.

No projeto que trabalho, nomenclatura é uma coisa muito importante, fugimos totalmente dos sufixos e prefixos, salvo raríssimas exceções("I" prefixo de interfaces e "Exception" sufixo de exceções). Mas não é tarefa fácil.

Abraço,

2012/7/19 Acáz Souza Pereira <acazsouza@gmail.com>

Essa semana achei este post "Your coding conventions are hurting you" e achei bem interessante. Gostaria de ler a opinião do pessoal aqui do grupo também.

Ele critica a nomenclatura usada hoje em dia pela maioria (leia-se gurus) e acho que tenho que concordar.

Venho achando OOP um paradigma muito "difícil" pois ele parece ser algo muito fácil de ser mal interpretado, é difícil se ver um código OOP bem feito. Quando digo bem feito, digo de fácil interpretação até para leigos. Geralmente se vê códigos com over-engineering e com nomes complicados, as vezes nos seres humanos gostamos de dificultar as coisas fáceis para que sentimos mais valor no que estamos fazendo, eu li uma vez que isso é um viés cognitivo e nessa nossa área é muito fácil deixar as coisas mais difíceis. Seria bom deixarmos o trabalho difícil para o que é realmente difícil.

É fato ver dúvidas de modelagem que fazem pouca ou nenhuma diferença na solução final. Perguntam como se existisse uma resposta exata. Isso é um trabalho subjetivo, só iremos saber se tal modelagem é boa ou não de fato quando os problemas (que não sabemos quais) aparecerem e esses problemas são únicos para cada projeto e para cada equipe. Tem uma apresentação do Nat Pryce (autor

do "Growing Object-Oriented Software, Guided by Tests" e desenvolvedor da biblioteca jMock) no InfoQ sobre o assunto: [Stop Refactoring!](#)

Enfim, é o primeiro artigo o motivo do tópico.

--

[Acáz Souza Pereira]

MSN/GTalk: acazsouza@gmail.com

Skype: acazsouza

Cel: (31) 8706-4103

[Quoted text hidden]

[Quoted text hidden]

[Quoted text hidden]

[Quoted text hidden]

[Quoted text hidden]

[Quoted text hidden]

[Quoted text hidden]

Thiago Silva <thiagosb13@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Tue, Jul 24, 2012 at 9:11 AM

Nem sempre um nome bom vem de primeira. O problema é engessá-lo. Se daqui a duas semanas, encontrarmos um nome melhor, por que não refatorar?

Cezar, concordo totalmente com você, o ser humano tende muito mais a anarquia do que a organização.

[Quoted text hidden]

Wilmer Roberto Ferreira Marques <wferreira@live.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Tue, Jul 31, 2012 at 2:34 PM

Eu acho que tudo depende de muitas coisas, mas no caso do desenvolvimento para .net eu me baseio muito na documentação da microsoft "Design Guidelines for Developing Class Libraries" (<http://msdn.microsoft.com/en-us/library/ms229042.aspx>) acho ótimo e muito limpo.

[Quoted text hidden]

Thiago Silva <thiagosb13@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Wed, Aug 1, 2012 at 11:36 AM

Isso diz respeito mais a forma dos nomes, que deveria ser premissa segui-las.

Tão importante quanto, é a semântica que se emprega nos nomes dos objetos. Por que não utilizar Carros no lugar de CarroRepository?

2012/7/31 Wilmer Roberto Ferreira Marques <wferreira@live.com>

Eu acho que tudo depende de muitas coisas, mas no caso do desenvolvimento para .net eu me baseio muito na documentação da microsoft "Design Guidelines for Developing Class Libraries" (<http://msdn.microsoft.com/en-us/library/ms229042.aspx>) acho ótimo e muito limpo.

Em quinta-feira, 19 de julho de 2012 00h12min46s UTC-3, Acáz Souza escreveu:

Essa semana achei este post "Your coding conventions are hurting you" e achei bem interessante.

Gostaria de ler a opinião do pessoal aqui do grupo também.

Ele critica a nomenclatura usada hoje em dia pela maioria (leia-se gurus) e acho que tenho que concordar.

Venho achando OOP um paradigma muito "difícil" pois ele parece ser algo muito fácil de ser mal interpretado, é difícil se ver um código OOP bem feito. Quando digo bem feito, digo de fácil interpretação até para leigos. Geralmente se vê códigos com over-engineering e com nomes complicados, as vezes nos seres humanos gostamos de dificultar as coisas fáceis para que sentamos mais valor no que estamos fazendo, eu li uma vez que isso é um viés cognitivo e nessa nossa área é muito fácil deixar as coisas mais difíceis. Seria bom deixarmos o trabalho difícil para o que é realmente difícil.

É fato ver dúvidas de modelagem que fazem pouca ou nenhuma diferença na solução final. Perguntam como se existisse uma resposta exata. Isso é um trabalho subjetivo, só iremos saber se tal modelagem é boa ou não de fato quando os problemas (que não sabemos quais) aparecerem e esses problemas são únicos para cada projeto e para cada equipe. Tem uma apresentação do Nat Pryce (autor do "Growing Object-Oriented Software, Guided by Tests" e desenvolvedor da biblioteca jMock) no InfoQ sobre o assunto: [Stop Refactoring!](#)

Enfim, é o primeiro artigo o motivo do tópico.

--

[Acáz Souza Pereira]

MSN/GTalk: acazsouza@gmail.com

Skype: [acazsouza](#)

Cel: (31) 8706-4103

twitter.com/acazsouza

acazsouza.com.br

--

Você recebeu esta mensagem porque faz parte do grupo .Net Architects hospedado no Google Groups.

Para postar envie uma mensagem para dotnetarchitects@googlegroups.com

Para sair do grupo envie uma mensagem para dotnetarchitects+unsubscribe@googlegroups.com

Para mais opções visite o grupo em <http://groups.google.com/group/dotnetarchitects?hl=pt-br>

[Quoted text hidden]

Acáz Souza Pereira <acazsouza@gmail.com>

Wed, Aug 1, 2012 at 11:47 AM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Me parece um assunto complexo Thiago.

Por mais paradoxal que isso seja nos precisamos de "nomes padrão" para nos comunicarmos entre nossos pares com mais clareza.

Acho que não precisa que seja "CarroRepository", pode ser um outro nome que se adequa melhor no contexto, mas no fim o conceito é o mesmo, é o conceito de repositório documentado com o nome Repository Pattern. O benefício que se tem seguindo isso é que um programador vendo Repository no fim do nome ele já vai sacar na hora que se trata do conceito de RepositoryPattern, esse ao meu ver é o benefício que se tem. E a parte ruim é que se o programador não conhecer o conceito documentado as coisas vão ficar sem nexos, sem se harmonizar com o domínio trabalhado.

[Quoted text hidden]

Cel: (31) 8706-4103

twitter.com/acazsouza

acazsouza.com.br

[Quoted text hidden]

Thiago Silva <thiagosb13@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Wed, Aug 1, 2012 at 12:04 PM

Pois é, mas se o cara (programador/cliente/PO...) não conhece o conceito de repository, você consegue explicar o que significa Carros num nível menos técnico, sem ter que explicar o padrão Repository.

2012/8/1 Acaz Souza Pereira <acazsouza@gmail.com>

[Quoted text hidden]

[Quoted text hidden]

Leandro Araujo Teixeira <leandro.recebe@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Wed, Aug 1, 2012 at 12:14 PM

Tivemos problemas aqui com coisas assim, um gerente que gosta de otimizar tudo. No inicio era (usando o exemplo citado) CarRepository tem que ser tudo em ingles para não ficar getNomeUsuario, vamos mudar para CarRepository depois use abreviações botão para btn lista para lst e CarRepository vira CarRep e por ultimo vamos reaproveitar código e quando alguém herda isso já vira CR e o próximo que chegar não tem a menor idéia do que seja sem ler todo código desde o inicio.

[Quoted text hidden]

—

Leandro Araújo Teixeira



www.fundacao.cc

[Quoted text hidden]

Cezar Augusto Guimaraes Neto <cezar.alevatto@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com
Cc: Cezar Augusto Guimaraes Neto <cezar.alevatto@gmail.com>

Sat, Aug 4, 2012 at 5:38 AM

Concordo plenamente que encontrar um bom nome não é fácil. Por isto a maioria simplesmente desisti. Por isto mencionei que é muitas vezes pura preguiça. O cara prefere não gastar tempo para achar um bom nome e simplesmente coloca a primeira coisa que vem na cabeça. Sobre o artigo, acho que ele coloca alguns outros pontos que vão além de simplesmente colocar um "I " nas interfaces. Então acho que tem o seu valor chamar atenção esta questão. Se lermos o nosso código mesmo, sempre vamos achar espaço para a melhorar algum nome que não ficou legal.

[Quoted text hidden]

[Quoted text hidden]

Fabício Cabral <fabriciofx@gmail.com>
To: dotnetarchitects@googlegroups.com

Sat, Aug 4, 2012 at 9:38 PM

Acho que vale a pena dar uma olhada neste artigo do Sergio Taborda:

<http://www.javabuilding.com/principles/nomenclatura.html>

[]'s

2012/8/4 Cezar Augusto Guimaraes Neto <cezar.alevatto@gmail.com>

[Quoted text hidden]

[Quoted text hidden]

[Quoted text hidden]

[Quoted text hidden]

[Quoted text hidden]

[Quoted text hidden]

[Quoted text hidden]

Cel: (31) 8706-4103

twitter.com/acazsouza

acazsouza.com.br

--

Você recebeu esta mensagem porque faz parte do grupo .Net Architects hospedado no Google Groups.

Para postar envie uma mensagem para dotnetarchitects@googlegroups.com

Para sair do grupo envie uma mensagem para dotnetarchitects+unsubscribe@googlegroups.com

Para mais opções visite o grupo em <http://groups.google.com/group/dotnetarchitects?hl=pt-br>

--

Você recebeu esta mensagem porque faz parte do grupo .Net Architects hospedado no Google Groups.

Para postar envie uma mensagem para dotnetarchitects@googlegroups.com

Para sair do grupo envie uma mensagem para dotnetarchitects+unsubscribe@googlegroups.com

Para mais opções visite o grupo em <http://groups.google.com/group/dotnetarchitects?hl=pt-br>

--

Você recebeu esta mensagem porque faz parte do grupo .Net Architects hospedado no Google Groups.

Para postar envie uma mensagem para dotnetarchitects@googlegroups.com

Para sair do grupo envie uma mensagem para dotnetarchitects+unsubscribe@googlegroups.com

Para mais opções visite o grupo em <http://groups.google.com/group/dotnetarchitects?hl=pt-br>

--

Você recebeu esta mensagem porque faz parte do grupo .Net Architects hospedado no Google Groups.

Para postar envie uma mensagem para dotnetarchitects@googlegroups.com

Para sair do grupo envie uma mensagem para dotnetarchitects+unsubscribe@googlegroups.com

Para mais opções visite o grupo em <http://groups.google.com/group/dotnetarchitects?hl=pt-br>

--

-fx

edmilson hora <edmilson_hora@yahoo.com.br>

Sun, Aug 5, 2012 at 8:06 AM

Reply-To: dotnetarchitects@googlegroups.com

To: "dotnetarchitects@googlegroups.com" <dotnetarchitects@googlegroups.com>

Esse artigo que o Fabricio indicou, me chamou a atenção o trecho abaixo, vocês concordam com o autor do post?

"Nomeando Interfaces

Muitos gostam de prefixar suas interfaces com um I maiusculo. Esta prática não é recomendada [1] por violar explicitamente o encapsulamento fornecendo informação demais. Além disso, se nos arrependemos depois e quisermos mudar para uma classe essa mudança irá provocar uma refactoramento desnecessário do código inteiro e até de código cliente já escrito por outras pessoas."

- 1) Viola o encapsulamente informar que uma interface é uma interface??
- 2) Classes e Interfaces são coisas bem distintas, então quando temos que trocar uma interface por uma classe temos que refatorar coisas sim, senão não faria sentido trocar uma interface por uma classe abstrata que simplesmente força as "classes filhas" a implementar metodos abstratos.

bom! é o que penso no momento.

De: Fabrício Cabral <fabriciofx@gmail.com>

Para: dotnetarchitects@googlegroups.com

Enviadas: Sábado, 4 de Agosto de 2012 21:38

[Quoted text hidden]

[Quoted text hidden]

[]'s

[Quoted text hidden]

[Quoted text hidden]

[Quoted text hidden]

[Quoted text hidden]

[Quoted text hidden]

[Quoted text hidden]

[Quoted text hidden]

[Quoted text hidden]

Cel: (31) 8706-4103

twitter.com/acazsouza

acazsouza.com.br

--

Você recebeu esta mensagem porque faz parte do grupo .Net Architects hospedado no Google Groups.

Para postar envie uma mensagem para dotnetarchitects@googlegroups.com

Para sair do grupo envie uma mensagem para dotnetarchitects+unsubscribe@googlegroups.com

Para mais opções visite o grupo em <http://groups.google.com/group/dotnetarchitects?hl=pt-br>

--

Você recebeu esta mensagem porque faz parte do grupo .Net Architects hospedado no Google Groups.
Para postar envie uma mensagem para dotnetarchitects@googlegroups.com
Para sair do grupo envie uma mensagem para dotnetarchitects+unsubscribe@googlegroups.com
Para mais opções visite o grupo em <http://groups.google.com/group/dotnetarchitects?hl=pt-br>

--

Você recebeu esta mensagem porque faz parte do grupo .Net Architects hospedado no Google Groups.
Para postar envie uma mensagem para dotnetarchitects@googlegroups.com
Para sair do grupo envie uma mensagem para dotnetarchitects+unsubscribe@googlegroups.com
Para mais opções visite o grupo em <http://groups.google.com/group/dotnetarchitects?hl=pt-br>

--

Você recebeu esta mensagem porque faz parte do grupo .Net Architects hospedado no Google Groups.
Para postar envie uma mensagem para dotnetarchitects@googlegroups.com
Para sair do grupo envie uma mensagem para dotnetarchitects+unsubscribe@googlegroups.com
Para mais opções visite o grupo em <http://groups.google.com/group/dotnetarchitects?hl=pt-br>

--

--fx

--

Você recebeu esta mensagem porque faz parte do grupo .Net Architects hospedado no Google Groups.
Para postar envie uma mensagem para dotnetarchitects@googlegroups.com
Para sair do grupo envie uma mensagem para dotnetarchitects+unsubscribe@googlegroups.com
Para mais opções visite o grupo em <http://groups.google.com/group/dotnetarchitects?hl=pt-br>

[Quoted text hidden]

Daniel Moreira Yokoyama <moreira.yokoyama@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Sun, Aug 5, 2012 at 12:10 PM

Pra variar, falta compreensão do papel da interface como recurso d modelagem na oop.
Existe uma mania insuportável de diferenciar interfaces e classes abstratas somente pela característica de ter ou não implementação.

[Quoted text hidden]

Bruno Busquete <busquete.bruno@gmail.com>

Sat, Aug 4, 2012 at 12:03 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Concordo em utilizar "I" para interfaces e "Repository" para classes que usem um padrão de repositório, pois tudo isso já está na cabeça do desenvolvedor.

Talvez um leigo realmente não entenda o que se está se passando, mas não acredito que um desenvolvedor tenha problemas nesses casos, pelo contrário, acho que facilita!

Agora, traduzir tudo para o inglês, não concordo!

Se for para padronizar, deixe tudo em português. "ObterCarro" ou "CriarCarro", pois já é complicado nomear métodos, e, quando passamos para o inglês corremos o risco de nos distanciar ainda mais do negócio.

Em quinta-feira, 19 de julho de 2012 00h12min46s UTC-3, Acaz Souza escreveu:

[Quoted text hidden]

[Quoted text hidden]

Juan Lopes <me@juanlopes.net>

Sun, Aug 5, 2012 at 5:32 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Se classes e interfaces são coisas distintas na sua modelagem, talvez você precise criar mais interfaces. Por que eu digo isso?

A única forma teoricamente correta de objetos se comunicarem é através de troca de mensagens. Em C#, troca de mensagem significa invocação de método. Quando o SOLID diz "programe para interfaces", isso não significa que você só possa depender de interfaces (no sentido estrito). Mas sim que você só deva depender da troca de mensagens entre objetos.

A interface não é um elemento concreto de uma modelagem. Ela só determina uma abstração para um conjunto de mensagens que faça sentido num contexto único. Todo seu conhecimento é sobre objetos e como enviar mensagens para eles. Elas são tão etéreas que em linguagens dinâmicas elas nem existem. A mensagem em si é o elemento de modelagem.

Se na sua linguagem isso é uma classe ou uma interface, o seu código não deveria saber. No final, tudo são interfaces.

2012/8/5 edmilson hora <edmilson_hora@yahoo.com.br>

[Quoted text hidden]

—

<https://github.com/juanlpopes>

[Quoted text hidden]

Cezar Augusto Guimaraes Neto <cezar.alevatto@gmail.com>

Sun, Aug 5, 2012 at 7:35 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Bruno,

o fato de desenvolvedores estarem acostumados em colocar "I" nos nomes das interfaces não faz disto uma boa prática.

Apenas não entendi o que você quis dizer "Talvez um leigo realmente não entenda o que se está se passando,...pelo contrário, acho que facilita!"

[]s

Cezar.

cezarguimaraes.com

On Aug 4, 2012, at 8:03 AM, Bruno Busquete <busquete.bruno@gmail.com> wrote:

Concordo em utilizar "I" para interfaces e "Repository" para classes que usem um padrão de repositório, pois tudo isso já está na cabeça do desenvolvedor.

Talvez um leigo realmente não entenda o que se está se passando, mas não acredito que um desenvolvedor tenha problemas nesses casos, pelo contrário, acho que facilita!

Agora, traduzir tudo para o inglês, não concordo!

Se for para padronizar, deixe tudo em português. "ObterCarro" ou "CriarCarro", pois já é complicado nomear métodos, e, quando passamos para o inglês corremos o risco de nos distanciar ainda mais do negócio.

Em quinta-feira, 19 de julho de 2012 00h12min46s UTC-3, Acáz Souza escreveu:

Essa semana achei este post "Your coding conventions are hurting you" e achei bem interessante. Gostaria de ler a opinião do pessoal aqui do grupo também.

Ele critica a nomenclatura usada hoje em dia pela maioria (leia-se gurus) e acho que tenho que concordar.

Venho achando OOP um paradigma muito "difícil" pois ele parece ser algo muito fácil de ser mal interpretado, é difícil se ver um código OOP bem feito. Quando digo bem feito, digo de fácil interpretação até para leigos. Geralmente se vê códigos com over-engineering e com nomes complicados, as vezes nos seres humanos gostamos de dificultar as coisas fáceis para que sintamos mais valor no que estamos fazendo, eu li uma vez que isso é um viés cognitivo e nessa nossa área é muito fácil deixar as coisas mais difíceis. Seria bom deixarmos o trabalho difícil para o que é realmente difícil.

É fato ver dúvidas de modelagem que fazem pouca ou nenhuma diferença na solução final. Perguntam como se existisse uma resposta exata. Isso é um trabalho subjetivo, só iremos saber se tal modelagem é boa ou não de fato quando os problemas (que não sabemos quais) aparecerem e esses problemas são únicos para cada projeto e para cada equipe. Tem uma apresentação do Nat Pryce (autor do "Growing Object-Oriented Software, Guided by Tests" e desenvolvedor da biblioteca jMock) no InfoQ sobre o assunto: [Stop Refactoring!](#)

Enfim, é o primeiro artigo o motivo do tópico.

--
[Acáz Souza Pereira]

MSN/GTalk: acazsouza@gmail.com

Skype: acazsouza

Cel: (31) 8706-4103

twitter.com/acazsouza

acazsouza.com.br

--
Você recebeu esta mensagem porque faz parte do grupo .Net Architects hospedado no Google Groups.

Para postar envie uma mensagem para dotnetarchitects@googlegroups.com

Para sair do grupo envie uma mensagem para dotnetarchitects+unsubscribe@googlegroups.com

Para mais opções visite o grupo em <http://groups.google.com/group/dotnetarchitects?hl=pt-br>

[Quoted text hidden]

Acáz Souza Pereira <acazsouza@gmail.com>

Mon, Aug 6, 2012 at 9:19 AM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Certo. É essa a crítica que eu via de quem criticava o "I" nas Interfaces, que no final não devia fazer diferença se é Interface ou não. Tudo devia ser abstração no final.

[Quoted text hidden]

Eric Lemes <ericlemes@gmail.com>

Sun, Aug 12, 2012 at 11:51 AM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Galera,

Acho que tem um pouco de "história" pra entender como as coisas são do jeito que são.

C# é uma linguagem que derivou do C e do C++. Em C e C++ tem alguns problemas clássicos com tipagem que não ocorrem em C#.

Alguns exemplos:

- 1) Um `*char` em C, pode ser qualquer coisa. Pode ser uma string ansi terminada em nulo, pode ser uma string wide terminada em nulo (apesar de usar mais `*wchar`), pode ser um BSTR (uma string que à esquerda do ponteiro tem 32 ou 64 bits que diz o tamanho dela), pode ser uma string utf8 (usa 1 byte para caracteres simples, de 1 a 4 bytes para caracteres especiais). Se vc não usar notação húngara fica louco. Alguém já viu algo pior que notação húngara? (Ex.: `char *szMinhaStringTerminadaEmZero`).
- 2) Em C++ existe o conceito de herança múltipla, mas não existe o conceito de interface. Uma interface em C++ é uma "pure virtual class" ou seja, uma classe que não implementa nada, porém, não é declarada como public interface. Acredito que seja daí que surgiu o "I" como prefixo para interfaces.

Quando derivaram tudo isso, misturou uma série de conceitos no C# que tem uma tipagem bem mais simples.

Mas o não colocar alguns prefixos e mudar algumas convenções, podem gerar alguns problemas. Por exemplo, se vc importa componentes COM no teu código, vai ter alguns tipos prefixados com "I" e outros não. Outra coisa legal é qdo vc usa `DllImport` no código e tenta dar nomes para as coisas do Windows da mesma forma como faz em C#.

Em resumo, o que eu sempre defendo nos times que trabalho é invés de tentar chegar num único consenso com toda a comunidade pra todo mundo trabalhar do mesmo jeito (o que eu acho utópico), eu crio a regra do condomínio dentro do time, no começo do projeto, ou seja, documentamos nossas code conventions para que o código seja o mais uniforme possível. Documentamos em que lingua o código será escrito, como dar nome aos membros da classe, sufixos, prefixos e tudo mais.

A primeira regra criada é que se não existe uma convenção para determinada situação, deve-se criá-la imediatamente antes de sair escrevendo o código.

Abraço,

Eric

2012/8/6 Acáz Souza Pereira <acazsouza@gmail.com>

[Quoted text hidden]

[Quoted text hidden]

Cel: (31) 8706-4103

twitter.com/acazsouza

acazsouza.com.br

--

Você recebeu esta mensagem porque faz parte do grupo .Net Architects hospedado no Google Groups.
Para postar envie uma mensagem para dotnetarchitects@googlegroups.com
Para sair do grupo envie uma mensagem para dotnetarchitects+unsubscribe@googlegroups.com
Para mais opções visite o grupo em <http://groups.google.com/group/dotnetarchitects?hl=pt-br>

[Quoted text hidden]

edmilson hora <edmilson_hora@yahoo.com.br>

Mon, Aug 13, 2012 at 7:55 AM

Reply-To: dotnetarchitects@googlegroups.com

To: "dotnetarchitects@googlegroups.com" <dotnetarchitects@googlegroups.com>

@Eric, eu acredito que a regra do condominio é bem sensata.

De: Eric Lemes <ericlemes@gmail.com>

Para: dotnetarchitects@googlegroups.com

Enviadas: Domingo, 12 de Agosto de 2012 11:51

[Quoted text hidden]

[Quoted text hidden]

[Quoted text hidden]

Thiago Silva <thiagosb13@gmail.com>

Mon, Aug 13, 2012 at 11:57 AM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Concordo com a regra de condomínio, desde que ela possa ser atualizada mediante novas ideias.

2012/8/13 edmilson hora <edmilson_hora@yahoo.com.br>

[Quoted text hidden]

[Quoted text hidden]