



Fabrício Cabral <fabriciofx@gmail.com>

[dotnetarchitects] Opinião sobre modelagem de objetos

97 messages

Bruno Costa <brunocosta.dsn@gmail.com>

Wed, Nov 9, 2011 at 2:25 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects <dotnetarchitects@googlegroups.com>

Caros, gostaria da opinião de vocês sobre a modelagem do seguinte cenário:

Os **transportes** considerados empresa são: **aéreo**, **terrestre**, **marítimo** e **fluvial**. O transporte aéreo é realizado através de **aviões** de transporte de passageiros. O transporte terrestre pode ser realizado de **trem**, **veículo alugado**, **táxi** e **ônibus**. O transporte marítimo e fluvial podem ser de **barco** ou **lancha**. As características relevantes destes transportes são a **tempo de viagem** e o **preço** (varia de acordo com o itinerário e com a empresa prestadora do serviço).

Foi feita uma classe pai, [Transporte] com os atributos <tempo> e <preco>. Três subclasses: [aereo], [terrestre], [maritmo] e [fluvial], que herdam de [Transporte] e em cada uma, enumerações com os subtipos de transporte aéreo (aviao), terrestre (trem, veículo alugado, táxi e ônibus), etc.

Segue em anexo o diagrama parcial, para uma idéia mais clara.

Bruno.

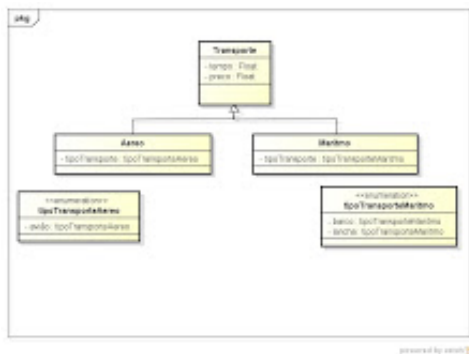
—

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>



Class Diagram0.jpg
59K

Daniel Silva <danielsilva.rj@gmail.com>

Wed, Nov 9, 2011 at 4:54 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Bruno,

Estamos modelando objetos, certo? Então, onde estão os comportamentos?

Se você nos disser que tipos de comportamentos estes objetos terão, ficará bem mais fácil te dizer se esta é uma boa saída. Da forma como está, poderemos sugerir uma classe "Coisa" acima de "Transporte"....

Qual o papel destas classes na tua modelagem? Servem apenas como referência mesmo, ou elas desempenharão algum comportamento? O negócio gira em torno delas?

Abs!

[Quoted text hidden]

Stefhan Rafael <stefhanrafael@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Wed, Nov 9, 2011 at 4:57 PM

[X] Trabalho da faculdade

[Quoted text hidden]

Rodrigo Silva de Andrade <dangling@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Wed, Nov 9, 2011 at 5:08 PM

O povo aqui cobra baratim pra fazer essa parada pra tu.

2011/11/9 Stefhan Rafael <stefhanrafael@gmail.com>

[Quoted text hidden]

[Quoted text hidden]

Vinicius Quaiato <vinicius.quaiato@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Wed, Nov 9, 2011 at 7:07 PM

vamos por partes: se você usa enumerações para representar diferentes *tipos*, considere que há algo errado.

[Quoted text hidden]

Vinicius Quaiato <vinicius.quaiato@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Wed, Nov 9, 2011 at 7:08 PM

segundo: não me parece que "tempo" e nem "preço" façam parte de transporte e sim de algo como "itinerário" e/ou "viagem".

On Wed, Nov 9, 2011 at 3:25 PM, Bruno Costa <brunocosta.dsn@gmail.com> wrote:

[Quoted text hidden]

[Quoted text hidden]

Vinicius Quaiato <vinicius.quaiato@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Wed, Nov 9, 2011 at 7:08 PM

Stefhan e Rodrigo: é interessante brincar de modelar... POUCOS de FATO sabem fazer isso... exercitem, é divertido.

[Quoted text hidden]

Stefhan Rafael <stefhanrafael@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Wed, Nov 9, 2011 at 9:32 PM

Nao tenho nada contra cenarios ficticios. Se for um trabalho da faculdade eh bem melhor ele tentar fazer sozinho e tirar alguma duvida no grupo e nao que outras pessoas façam o trabalho. Enfim, nada contra...so acho que faltou o "estou com problemas para fazer o trabalho da faculdade/curso/pos". Posso estar errado..se sim, pode ignorar oq eu falei.

[Quoted text hidden]

Rodrigo Silva de Andrade <dangling@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Thu, Nov 10, 2011 at 8:09 AM

Foi exatamente a minha impressão, parece que ele não tinha uma duvida, ele queria que fizessem o trabalho

por ele... enfim, se não foi isso, peço desculpas.

2011/11/9 Steffhan Rafael <stefhanrafael@gmail.com>

[Quoted text hidden]

[Quoted text hidden]

MayogaX <mayumisatox@gmail.com>

Thu, Nov 10, 2011 at 8:10 AM

Reply-To: dotnetarchitects@googlegroups.com

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

Eu acho que ele não está pedindo para que alguém faça o "suposto trabalho de faculdade"... ele só tá pedindo uma opinião, o que na faculdade seria como "mano, dá uma olhada pra ter certeza que eu não vou zerar".

Concordo em partes com o VQuaiato...

Tipo, acho que tem o preço do uso do transporte e o preço do itinerário, não são coisas diferentes?

Nesse caso teria uma classe Itinerário, com o transporte usado, preço, datas, carga...etc?

On 9 nov, 22:32, Steffhan Rafael <stefhanraf...@gmail.com> wrote:

> Não tenho nada contra cenários fictícios. Se for um trabalho da faculdade
> é bem melhor ele tentar fazer sozinho e tirar alguma dúvida no grupo e não
> que outras pessoas façam o trabalho. Enfim, nada contra...so acho que
> faltou o "estou com problemas para fazer o trabalho da
> faculdade/curso/pos". Posso estar errado..se sim, pode ignorar o que eu falei.

>
> Em 9 de novembro de 2011 19:08, Vinicius Quaiato <vinicius.quai...@gmail.com>

>
>
>
>

> > escreveu:

> > Steffhan e Rodrigo: é interessante brincar de modelar... POUCOS de FATO
> > sabem fazer isso... exercitem, é divertido.

>
> > On Wed, Nov 9, 2011 at 5:57 PM, Steffhan Rafael <stefhanraf...@gmail.com>wrote:

>
> >> [X] Trabalho da faculdade

>
> >> Em 9 de novembro de 2011 16:54, Daniel Silva <danielsilva...@gmail.com>escreveu:

>
> >> Bruno,

> >>> Estamos modelando objetos, certo? Então, onde estão os comportamentos?

>
> >>> Se você nos disser que tipos de comportamentos estes objetos terão,
> >>> ficará bem mais fácil te dizer se esta é uma boa saída. Da forma como está,
> >>> poderemos sugerir uma classe "Coisa" acima de "Transporte"....

>
> >>> Qual o papel destas classes na tua modelagem? Servem apenas como
> >>> referência mesmo, ou elas desempenharão algum comportamento? O negócio gira
> >>> em torno delas?

>
> >>> Abs!

>
> >>> Em 9 de novembro de 2011 15:25, Bruno Costa <brunocosta....@gmail.com>escreveu:

>
> >>> Caros, gostaria da opinião de vocês sobre a modelagem do seguinte
> >>>> cenário:

>
> >>>> Os *transportes* considerados empresa são: *aéreo*, *terrestre*, *

Vinicius, poderias explicar tua afirmação?

Atenciosamente,

2011/11/9 Vinicius Quaiato <vinicius.quaiato@gmail.com>

[Quoted text hidden]

—

Bernardo Bosak de Rezende

<http://bernardorezende.net/> - Arquitetura e Desenvolvimento

<https://github.com/bernardobrezende/> - GitHub

"The weather-cock on the church spire, though made of iron, would soon be broken by the storm-wind if it did not understand the noble art of turning to every wind."

Heinrich Heine

[Quoted text hidden]

Silvio Marques - Gmail <silviomarquesferreira@gmail.com>

Thu, Nov 10, 2011 at 9:01 AM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Pessoal, bom dia!

Pode ser que vocês me esmurem agora por falar o que vou falar e por não participar ativamente dessa lista, mas o que ví ultimamente aqui foi muita porrada pra tudo que é lado sem conhecimento de causa (não vou ficar repetindo aqui o que já foi dito ou quem disse).

Um belo exemplo foi esse do Bruno: se o cara tá fazendo um trampo de faculdade ou não não interessa, ele tá expondo um cenário e querendo tirar dúvidas!

O fato dele já ter exposto, da forma como expôs, independente da finalidade, demonstra um interesse em saber mais, querer aprender, querer estudar (OFF: ao contrário dos 73 da USP).

Não que não devamos dar porrada de vez em quando, mas desde que entrei nessa lista me senti num UFC.

Enfim, pode mandar as porradas em mim agora; sei que deve ter alguém aí que tá doido pra fazer isso.

Menos pô!

2011/11/10 Bernardo Bosak de Rezende <bernardosalada@gmail.com>

[Quoted text hidden]

—

"We're not yet where we can go!"

"I love email footer phrase!"

[Quoted text hidden]

Silvio Marques - Gmail <silviomarquesferreira@gmail.com>

Thu, Nov 10, 2011 at 9:02 AM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Aliás, aqui e no Net-Br também!

2011/11/10 Silvio Marques - Gmail <silviomarquesferreira@gmail.com>

[Quoted text hidden]

[Quoted text hidden]

Vinicius Quaiato <vinicius.quaiato@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Thu, Nov 10, 2011 at 9:33 AM

ql afirmação?

[Quoted text hidden]

Vinicius Quaiato <vinicius.quaiato@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Thu, Nov 10, 2011 at 9:35 AM

Você está certo Sílvia.

Não interessa se é um trabalho de faculdade, uma consultoria para a MS, apenas uma curiosidade, etc.

Modelagem e design são assuntos bem interessantes. Então chega de falar sobre isso e falemos sobre a modelagem do rapaz.

[Quoted text hidden]

[Quoted text hidden]

Bruno Costa <brunocosta.dsn@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Thu, Nov 10, 2011 at 9:46 AM

Obrigado Vinicius e MayogaX.

[Quoted text hidden]

Gustavo Rocha <gustavohrocha@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Thu, Nov 10, 2011 at 9:53 AM

Também queria saber o contexto da questão do enum pra tipo. fiquei curioso

[Quoted text hidden]

Stefhan Rafael <stefhanrafael@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Thu, Nov 10, 2011 at 9:54 AM

Eu gosto de transparência, ou seja, se for para a faculdade fala que é pra faculdade, se for pra consultoria da MS fale que é pra consultoria da MS etc.. que é o que tudo mundo geralmente faz "estou com um problema aqui na empresa, em projeto X". Mas deixa isso pra lá, se tem gente interessada em responder em qualquer condição não existe um problema. O intuito não era criar polemica. Inclusive posso até estar enganado na questão de mastigar trabalho da faculdade, mas ficou esta impressão ao ler a thread. Enfim, boa modelagem.

[Quoted text hidden]

Marco Aurelio <spmarco@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Thu, Nov 10, 2011 at 10:59 AM

Então... transparência é relativa... ele pode falar o que quiser, não vamos fazer a modelagem com base de para quem ou o que tá fazendo... simplesmente ajudar já tá bom....

Obs: fiquei com dúvida também quando o Quaiato falou de Enum também...

[Quoted text hidden]

Vinicius Quaiato <vinicius.quaiato@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Thu, Nov 10, 2011 at 11:24 AM

Sobre o enum:

Se estamos falando de classes e objetos, herança, polimorfismo, etc, por que eu preciso de um enum de tipos? Isso vai levar a coisas como

```
if(obj.Tipo == Tipos.X){...}  
else if (obj.Tipo == Tipos.Y) {...}
```

ou um switch...

Se estamos falando de OO, devemos desenhar as coisas de forma que objetos de uma mesma família possam responder de forma transparente independente de suas classes:

```
var objA = new Caminhao();  
var objB = new Lancha();
```

se é dado que eles são da mesma família, e usados no mesmo contexto, eles deveria ser capazes de responder igualmente a coisas como:

```
objA.FazerViagem();  
objB.FazerViagem();
```

Isto quer dizer que QUEM UTILIZA esses objetos deve fazer uso de uma superclasse, ou interface, que seja comum:

```
public void Reservar(Transporte transporte){  
    //qualquer código de negócios  
    transporte.FazerViagem();  
}
```

algo assim, sem ifs, sem switch.

[Quoted text hidden]

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

Thu, Nov 10, 2011 at 11:24 AM

Olá todos,

eu assino várias listas de discussão, e na grande maioria, eu sempre vejo alguns membros reclamando quando aparecem dúvidas simples. Muitos destes acham que as dúvidas "simples" são de estudantes querendo que os outros façam os seus trabalhos da faculdade.

Então só gostaria, se possível, de pedir uma coisa: quando aparecer uma dúvida "simples" ou "boba", vamos dar uma chance a esta dúvida ou pergunta? Quem sabe esta pergunta não acaba rendendo uma boa discussão? Até rever certos "conceitos" que achamos que sabemos, mas que talvez não seja bem assim?

Eu digo isso por experiência própria: uma outra lista que eu assino, fiz certas perguntas "bobas" mas que acabaram gerando uma *ótima* discussão, inclusive com os mais experientes da lista. E no final, *todos* da lista saíram ganhando.

Faço essa observação, porque outro dia, aqui mesmo nesta lista, fiz uma pergunta "boba" sobre um domínio e DDD e mandaram eu ir ler o livro do Evans. Nem se importaram em começar uma troca de ideias ou debate.

Na minha opinião, é com este tipo de comportamento ou atitude que quem perde somos todos nós da lista.

Desculpem pelo "desabafo", mas é isso que às vezes eu sinto por aqui.

Atenciosamente,

2011/11/9 Vinicius Quaiato <vinicius.quaiato@gmail.com>

Stefhan e Rodrigo: é interessante brincar de modelar... POUCOS de FATO sabem fazer isso... exercitem, é divertido.

—

—fx

Vinicius Quaiato <vinicius.quaiato@gmail.com>

Thu, Nov 10, 2011 at 11:26 AM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

complementando:

Acho que enum visa delimitar tipos que não sejam objetos.

Exemplo?

Cores

BandeirasCartaoCredito

Sexo

etc

Isso quer dizer que esses enums não delimitam objetos e comportamentos, apenas tipos em escolhas.

[Quoted text hidden]

Fabício Cabral <fabriciofx@gmail.com>

Thu, Nov 10, 2011 at 11:30 AM

To: dotnetarchitects@googlegroups.com

Silvio,

além de concordar com o que você disse, também já manifestei a minha opinião em outro e-mail.

Atenciosamente,

2011/11/10 Silvio Marques - GMail <silviomarquesferreira@gmail.com>

Pessoal, bom dia!

Pode ser que vocês me esmurrem agora por falar o que vou falar e por não participar ativamente dessa lista, mas o que ví ultimamente aqui foi muita porrada pra tudo que é lado sem conhecimento de causa (não vou ficar repetindo aqui o que já foi dito ou quem disse).

Um belo exemplo foi esse do Bruno: se o cara tá fazendo um trampo de faculdade ou não não interessa, ele tá expondo um cenário e querendo tirar dúvidas!

O fato dele já ter exposto, da forma como expôs, independente da finalidade, demonstra um interesse em saber mais, querer aprender, querer estudar (OFF: ao contrário dos 73 da USP).

Não que não devamos dar porrada de vez em quando, mas desde que entrei nessa lista me senti num UFC.

Enfim, pode mandar as porradas em mim agora; sei que deve ter alguém aí que tá doido pra fazer isso.

Menos pô!

—

—fx

Fabício Cabral <fabriciofx@gmail.com>

Thu, Nov 10, 2011 at 11:37 AM

To: dotnetarchitects@googlegroups.com

Olá Vinicius,

parando para analisar um pouco os seus exemplos, me pareceu que a utilização de enums é como se fossem de valores para atributos de objetos. Veja se é isso mesmo:

Cores:

VERMELHO, AZUL, ...

BandeirasCartaoCredito:

VISA, MASTERCARD, ...

Sexo:

MASCULINO, FEMININO.

Mas quando eu desenvolvo um software, eu procuro sempre deixar o sistema o mais flexível possível. A ideia de utilizar enums, pelo menos ao meu ver, deixa o sistema um tanto quanto "amarrado". Exemplo: se eu quiser adicionar uma nova bandeira de cartão de crédito, eu vou ter que modificar o código do meu sistema para suportar isso.

No caso de se querer permitir ao próprio utilizador do sistema de adicionar ou remover bandeiras de cartão de crédito do meu sistema, vou acabar transformando o que era uma propriedade de um objeto, em um objeto?

Atenciosamente,

2011/11/10 Vinicius Quaiato <vinicius.quaiato@gmail.com>

[Quoted text hidden]

--

-fx

Vinicius Quaiato <vinicius.quaiato@gmail.com>

Thu, Nov 10, 2011 at 11:43 AM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Pois é... por isso é que eu falei sobre valores. Enums são apenas valores. Na verdade enum é um "syntax sugar" para um inteiro, nada além disso.

Provavelmente adicionar uma nova bandeira de cartão não é apenas "fazer o cadastro do nome da bandeira" ou adicionar uma linha na "tabela de bandeiras no banco". Então não vejo problemas ter isso em enums, são apenas valores.

O meu ponto é: não utilizar enums para diferenciar tipos de objetos.

[Quoted text hidden]

[Quoted text hidden]

Carlos Abdalla <caabdalla@gmail.com>

Thu, Nov 10, 2011 at 12:02 PM

Reply-To: dotnetarchitects@googlegroups.com

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

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

Concordo em partes! Se sua escolha for simples escolha e nao modificar comportamento do sistema ok, caso contrario vc ira ter a codificar do mesmo jeito a nao ser a vc faca sistemas de simples acoplamento de componentes e afins mas para fazer o componente precisaria codificar mas seria apenas ele, pensando assim para que criar um objeto para esse trípode de escolha?

Um exemplo do que eu disse, em nossos ecommerces temos essa opção do cliente simplesmente colocar um novo meio de pagamento mas para isso ele precisa codificar esse componente o que quem acaba

fazendo somos nos, btw ...

Com os exemplos citados eu continuo com a opção do Quaiato.

Att.

Carlos Abdalla

Enviado via iPhone

[Quoted text hidden]

[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]

Carlos Abdalla <caabdalla@gmail.com>

Thu, Nov 10, 2011 at 12:03 PM

Reply-To: dotnetarchitects@googlegroups.com

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

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

Concordo plenamente inclusive o Quaiato levantou algumas coisas q valeu a pena recordar conceitualmente!

Att.

Carlos Abdalla

Enviado via iPhone

[Quoted text hidden]

[Quoted text hidden]

--

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]

Juan Lopes <me@juanlopes.net>

Thu, Nov 10, 2011 at 12:12 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Enums só levam a isso porque em .NET enums não tem comportamento.

Java é melhor que .NET. Nesse sentido. :P

2011/11/10 Vinicius Quaiato <vinicius.quaiato@gmail.com>

[Quoted text hidden]

[Quoted text hidden]

Rafael Ponte <rponete@gmail.com>

Thu, Nov 10, 2011 at 12:20 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Isso é verdade, logo em java podemos ter enums funcionando como value objects com comportamentos simples.

2011/11/10 Juan Lopes <me@juanlopes.net>

[Quoted text hidden]

—
Rafael Ponte
<http://www.rponte.com.br>
[Quoted text hidden]

Vinicius Quaiato <vinicius.quaiato@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Thu, Nov 10, 2011 at 12:23 PM

Para isso existem as classes Juan LOL
[Quoted text hidden]

Vinicius Quaiato <vinicius.quaiato@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Thu, Nov 10, 2011 at 12:30 PM

Sim isso é bacana, mas dá pra resolver isso com um pattern <ouvem-se trombetas e uma luz forte como que vindo do paraíso>:

<http://viniciusquaiato.com/blog/evitando-ifs-e-switches-com-state-pattern-isso-e-oo/>

[Quoted text hidden]

[Quoted text hidden]

Juan Lopes <me@juanlopes.net>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Thu, Nov 10, 2011 at 12:37 PM

Classes Juan?

Sent from mobile device.

[Quoted text hidden]

Vinicius Quaiato <vinicius.quaiato@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Thu, Nov 10, 2011 at 12:43 PM

classes Juan are just like "joker classes" :D
[Quoted text hidden]

MayogaX <mayumisatox@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: ".Net Architects" <dotnetarchitects@googlegroups.com>

Thu, Nov 10, 2011 at 12:44 PM

Eu gostei da discussão que tá rendendo essa thread.

Mas eu me confundi...acho que não li algum email...por que estão falando de Enum mesmo?

On 10 nov, 12:43, Vinicius Quaiato <vinicius.quaiato@gmail.com> wrote:

> Pois é... por isso é que eu falei sobre valores. Enums são apenas valores.

> Na verdade enum é um "syntax sugar" para um inteiro, nada além disso.

>

> Provavelmente adicionar uma nova bandeira de cartão não é apenas "fazer o

> cadastro do nome da bandeira" ou adicionar uma linha na "tabela de

> bandeiras no banco". Então não vejo problemas ter isso em enums, são apenas

> valores.

>

> O meu ponto é: não utilizar enums para diferenciar tipos de objetos.

>

> On Thu, Nov 10, 2011 at 12:37 PM, Fabrício Cabral <fabrici...@gmail.com> wrote:
>
>
>
> > Olá Vinicius,
>
> > parando para analisar um pouco os seus exemplos, me pareceu
> > que a utilização de enums é como se fossem de valores para
> > atributos de objetos. Veja se é isso mesmo:
>
> > Cores:
> > VERMELHO, AZUL, ...
>
> > BandeirasCartaoCredito:
> > VISA, MASTERCARD, ...
>
> > Sexo:
> > MASCULINO, FEMININO.
>
> > Mas quando eu desenvolvo um software, eu procuro sempre
> > deixar o sistema o mais flexível possível. A ideia de utilizar
> > enums, pelo menos ao meu ver, deixa o sistema um tanto
> > quanto "amarrado". Exemplo: se eu quiser adicionar uma
> > nova bandeira de cartão de crédito, eu vou ter que modificar
> > o código do meu sistema para suportar isso.
>
> > No caso de se querer permitir ao próprio utilizador do sistema
> > de adicionar ou remover bandeiras de cartão de crédito do
> > meu sistema, vou acabar transformando o que era uma
> > propriedade de um objeto, em um objeto?
>
> > Atenciosamente,
>
> > 2011/11/10 Vinicius Quaiato <vinicius.quai...@gmail.com>
>
> >> complementando:
>
> >> Acho que enum visa delimitar tipos que não sejam objetos.
>
> >> Exemplo?
> >> Cores
> >> BandeirasCartaoCredito
> >> Sexo
> >> etc
>
> >> Isso quer dizer que esses enums não delimitam objetos e comportamentos,
> >> apenas tipos em escolhas.
>
> >> On Thu, Nov 10, 2011 at 12:24 PM, Vinicius Quaiato <
> >> vinicius.quai...@gmail.com> wrote:
>
> >>> Sobre o enum:
>
> >>> Se estamos falando de classes e objetos, herança, polimorfismo, etc, por
> >>> que eu preciso de um enum de tipos? Isso vai levar a coisas como
>
> >>> if(obj.Tipo == Tipos.X){.}
> >>> else if (obj.Tipo == Tipos.Y) {...}
>
> >>> ou um switch...
>
> >>> Se estamos falando de OO, devemos desenhar as coisas de forma que
> >>> objetos de uma mesma família possam responder de forma transparente
> >>> independente de suas classes:
>

```
> >>> var objA = new Caminhao();
> >>> var objB = new Lancha();
>
> >>> se é dado que eles são da mesma família, e usados no mesmo contexto,
> >>> eles deveria ser capazes de responder igualmente a coisas como:
>
> >>> objA.FazerViagem();
> >>> objB.FazerViagem();
>
> >>> Isto quer dizer que QUEM UTILIZA esses objetos deve fazer uso de uma
> >>> superclasse, ou interface, que seja comum:
>
> >>> public void Reservar(Transporte transporte){
> >>>     //qualquer código de negócios
> >>>     transporte.FazerViagem();
> >>> }
>
> >>> algo assim, sem ifs, sem switch.
>
> >>> On Thu, Nov 10, 2011 at 11:59 AM, Marco Aurelio <spma...@gmail.com>wrote:
>
> >>>> Então... transparência e relativa... ele pode falar o que quiser, não
> >>>> vamos fazer a modelagem com base de para quem ou o que ta fazendo...
> >>>> simplesmente ajudar ja ta bom....
> >>>> Obs: fiquei com dúvida tambem quando o Quaiato falou de Enum tambem...
> >>>> Em 10/11/2011 10:54, "Stefhan Rafael" <stefhanraf...@gmail.com>
> >>>> escreveu:
>
> >>>>> Eu gosto de transparência, ou seja, se for para a faculdade fala que eh
> >>>>> pra faculdade, se for pra consultoria da MS fale que eh para consultoria da
> >>>>> MS etc.. que eh o que tudo mundo geralmente faz "estou com um problema aqui
> >>>>> na empresa, em projeto X". Mas deixa isso pra la, se tem gente interessada
> >>>>> em responder em qualquer condição não existe um problema.O intuito não era
> >>>>> criar polemica. Inclusive posso até estar enganado na questao de mastigar
> >>>>> trabalho da faculdade,mas ficou esta impressão ao ler a thread. Enfim, boa
> >>>>> modelagem.
>
> >>>>> Em 10 de novembro de 2011 09:35, Vinicius Quaiato <
> >>>>> vinicius.quai...@gmail.com> escreveu:
>
> >>>>>> Você está certo Sílvio.
>
> >>>>>> Não interessa se é um trabalho de faculdade, uma consultoria para a
> >>>>>> MS, apenas uma curiosidade, etc.
>
> >>>>>> Modelagem e design são assuntos bem interessantes. Então chega de
> >>>>>> falar sobre isso e falemos sobre a modelagem do rapaz.
>
> >>>>>> On Thu, Nov 10, 2011 at 10:01 AM, Silvio Marques - GMail <
> >>>>>> silviomarquesferre...@gmail.com> wrote:
>
> >>>>>>> Pessoal, bom dia!
>
> >>>>>>> Pode ser que vocês me esmurem agora por falar o que vou falar e por
> >>>>>>> não participar ativamente dessa lista, mas o que ví ultimamente aqui foi
> >>>>>>> muita porrada pra tudo que é lado sem conhecimento de causa (não vou ficar
> >>>>>>> repetindo aqui o que já foi dito ou quem disse).
>
> >>>>>>> Um belo exemplo foi esse do Bruno: se o cara tá fazendo um trampo de
> >>>>>>> faculdade ou não não interessa, ele tá expondo um cenário e querendo tirar
> >>>>>>> dúvidas!
>
> >>>>>>> O fato dele já ter exposto, da forma como expôs, independente da
> >>>>>>> finalidade, demonstra um interesse em saber mais, querer aprender, querer
> >>>>>>> estudar (OFF: ao contrário dos 73 da USP).
```

>
 > >>>>>> Não que não devamos dar porrada de vez em quando, mas desde que
 > >>>>>> entrei nessa lista me senti num UFC.
 >
 > >>>>>> Enfim, pode mandar as porradas em mim agora; sei que deve ter alguém
 > >>>>>> aí que tá doido pra fazer isso.
 >
 > >>>>>> Menos pô!
 >
 > >>>>>> 2011/11/10 Bernardo Bosak de Rezende <bernardosal...@gmail.com>
 >
 > >>>>>> Bah que tristeza, quase ninguém interessado em discutir a
 > >>>>>> modelagem...
 >
 > >>>>>> Vinicius, poderias explicar tua afirmação?
 >
 > >>>>>> Atenciosamente,
 >
 > >>>>>> 2011/11/9 Vinicius Quaiato <vinicius.quai...@gmail.com>
 >
 > >>>>>>> vamos por partes: se você usa enumerações para representar
 > >>>>>>> diferentes *tipos, *considere que há algo errado.
 >
 > >>>>>>> On Wed, Nov 9, 2011 at 3:25 PM, Bruno Costa <
 > >>>>>>> brunocosta....@gmail.com> wrote:
 >
 > >>>>>>> Caros, gostaria da opinião de vocês sobre a modelagem do seguinte
 > >>>>>>> cenário:
 >
 > >>>>>>>> Os *transportes *considerados empresa são: *aéreo*, *terrestre*,
 > >>>>>>>> *marítimo* e *fluvial*. O transporte aéreo é realizado através
 > >>>>>>>> de *aviões *de transporte de passageiros. O transporte terrestre
 > >>>>>>>> pode ser realizado de* trem, veículo alugado, táxi e ônibus*. O
 > >>>>>>>> transporte marítimo e fluvial podem ser de *barco ou lancha*. As
 > >>>>>>>> características relevantes destes transportes são a* tempo de
 > >>>>>>>> viagem e o preço* (varia de acordo com o itinerário e com a
 > >>>>>>>> empresa prestadora do serviço).
 >
 > >>>>>>>> Foi feita uma classe pai, [Transporte] com os atributos <tempo> e
 > >>>>>>>> <preco>. Três subclasses: [aereo], [terrestre], [maritmo] e [fluvial], que
 > >>>>>>>> herdam de [Trasporte] e em cada uma, enumerações com os subtipos de
 > >>>>>>>> transporte aéreo (aviao), terrestre (trem, veículo alugado, táxi e ônibus),
 > >>>>>>>> etc.
 >
 > >>>>>>>> Segue em anexo o diagrama parcial, para uma idéia mais clara.
 >
 > >>>>>>>> Bruno.
 >
 > >>>>>>>> --
 > >>>>>>>> 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>
 >
 > >>>>>>> --
 > >>>>>>> Bernardo Bosak de Rezende
 > >>>>>>><http://bernardorezende.net/>- Arquitetura e Desenvolvimento
 > >>>>>>><https://github.com/bernardobrezende/>- GitHub
 >
 > >>>>>>> "The weather-cock on the church spire, though made of iron, would
 > >>>>>>> soon be broken by the storm-wind if it did not understand the noble art of
 > >>>>>>> turning to every wind."
 > >>>>>>> Heinrich Heine
 >
 > >>>>>>> --
 > >>>>>>> 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>
 >
 > >>>>>>> --
 > >>>>>>> "We're not yet where we can go!"
 > >>>>>>> "I love email footer phrase!"
 >
 > >>>>>>> --
 > >>>>>>> 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
 >
 > ...
 >
 > mais »

--
 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
 [Quoted text hidden]

Juan Lopes <me@juanlopes.net>
 Reply-To: dotnetarchitects@googlegroups.com
 To: dotnetarchitects@googlegroups.com

Thu, Nov 10, 2011 at 12:54 PM

Por que TipoTransporteAereo e TipoTransporteMaritmo são enums.

2011/11/10 MayogaX <mayumisatox@gmail.com>

Eu gostei da discussão que tá rendendo essa thread.

Mas eu me confundi...acho que não li algum email...por que estão falando de Enum mesmo?

On 10 nov, 12:43, Vinicius Quaiato <vinicius.quai...@gmail.com> wrote:
> Pois é... por isso é que eu falei sobre valores. Enums são apenas valores.
> Na verdade enum é um "syntax sugar" para um inteiro, nada além disso.
>
> Provavelmente adicionar uma nova bandeira de cartão não é apenas "fazer o
> cadastro do nome da bandeira" ou adicionar uma linha na "tabela de
> bandeiras no banco". Então não vejo problemas ter isso em enums, são apenas
> valores.
>
> O meu ponto é: não utilizar enums para diferenciar tipos de objetos.
>
> On Thu, Nov 10, 2011 at 12:37 PM, Fabrício Cabral <fabrici...@gmail.com> wrote:
>
>
>
> > Olá Vinicius,
>
> > parando para analisar um pouco os seus exemplos, me pareceu
> > que a utilização de enums é como se fossem de valores para
> > atributos de objetos. Veja se é isso mesmo:
>
> > Cores:
> > VERMELHO, AZUL, ...
>
> > BandeirasCartaoCredito:
> > VISA, MASTERCARD, ...
>
> > Sexo:
> > MASCULINO, FEMININO.
>
> > Mas quando eu desenvolvo um software, eu procuro sempre
> > deixar o sistema o mais flexível possível. A ideia de utilizar
> > enums, pelo menos ao meu ver, deixa o sistema um tanto
> > quanto "amarrado". Exemplo: se eu quiser adicionar uma
> > nova bandeira de cartão de crédito, eu vou ter que modificar
> > o código do meu sistema para suportar isso.
>
> > No caso de se querer permitir ao próprio utilizador do sistema
> > de adicionar ou remover bandeiras de cartão de crédito do
> > meu sistema, vou acabar transformando o que era uma
> > propriedade de um objeto, em um objeto?
>
> > Atenciosamente,
>
> > 2011/11/10 Vinicius Quaiato <vinicius.quai...@gmail.com>
>
> >> complementando:
>
> >> Acho que enum visa delimitar tipos que não sejam objetos.
>
> >> Exemplo?
> >> Cores
> >> BandeirasCartaoCredito
> >> Sexo
> >> etc
>
> >> Isso quer dizer que esses enums não delimitam objetos e comportamentos,
> >> apenas tipos em escolhas.
>
> >> On Thu, Nov 10, 2011 at 12:24 PM, Vinicius Quaiato <
> >> vinicius.quai...@gmail.com> wrote:
>
> >>> Sobre o enum:
>


```
> >>> Se estamos falando de classes e objetos, herança, polimorfismo, etc, por
> >>> que eu preciso de um enum de tipos? Isso vai levar a coisas como
>
> >>> if(obj.Tipo == Tipos.X){..}
> >>> else if (obj.Tipo == Tipos.Y) {...}
>
> >>> ou um switch...
>
> >>> Se estamos falando de OO, devemos desenhar as coisas de forma que
> >>> objetos de uma mesma família possam responder de forma transparente
> >>> independente de suas classes:
>
> >>> var objA = new Caminhao();
> >>> var objB = new Lancha();
>
> >>> se é dado que eles são da mesma família, e usados no mesmo contexto,
> >>> eles deveria ser capazes de responder igualmente a coisas como:
>
> >>> objA.FazerViagem();
> >>> objB.FazerViagem();
>
> >>> Isto quer dizer que QUEM UTILIZA esses objetos deve fazer uso de uma
> >>> superclasse, ou interface, que seja comum:
>
> >>> public void Reservar(Transporte transporte){
> >>>     //qualquer código de negócios
> >>>     transporte.FazerViagem();
> >>> }
>
> >>> algo assim, sem ifs, sem switch.
>
> >>> On Thu, Nov 10, 2011 at 11:59 AM, Marco Aurelio <spma...@gmail.com>wrote:
>
> >>>> Então... transparência e relativa... ele pode falar o que quiser, não
> >>>> vamos fazer a modelagem com base de para quem ou o que ta fazendo...
> >>>> simplesmente ajudar ja ta bom....
> >>>> Obs: fiquei com dúvida tambem quando o Quaiato falou de Enum tambem...
> >>>> Em 10/11/2011 10:54, "Stefhan Rafael" <stefhanraf...@gmail.com>
> >>>> escreveu:
>
> >>>> Eu gosto de transparência, ou seja, se for para a faculdade fala que eh
> >>>>> pra faculdade, se for pra consultoria da MS fale que eh para consultoria da
> >>>>> MS etc.. que eh o que tudo mundo geralmente faz "estou com um problema aqui
> >>>>> na empresa, em projeto X". Mas deixa isso pra la, se tem gente interessada
> >>>>> em responder em qualquer condição não existe um problema.O intuito não era
> >>>>> criar polemica. Inclusive posso até estar enganado na questao de mastigar
> >>>>> trabalho da faculdade,mas ficou esta impressão ao ler a thread. Enfim, boa
> >>>>> modelagem.
>
> >>>>> Em 10 de novembro de 2011 09:35, Vinicius Quaiato <
> >>>>> vinicius.quai...@gmail.com> escreveu:
>
> >>>>>> Você está certo Sílvio.
>
> >>>>>> Não interessa se é um trabalho de faculdade, uma consultoria para a
> >>>>>> MS, apenas uma curiosidade, etc.
>
> >>>>>> Moelagem e design são assuntos bem interessantes. Então chega de
> >>>>>> falar sobre isso e falemos sobre a modelagem do rapaz.
>
> >>>>>> On Thu, Nov 10, 2011 at 10:01 AM, Silvio Marques - GMail <
> >>>>>> silviomarquesferre...@gmail.com> wrote:
>
> >>>>>>> Pessoal, bom dia!
```

> >>>>>> Pode ser que vocês me esmurem agora por falar o que vou falar e por
> >>>>>> não participar ativamente dessa lista, mas o que ví ultimamente aqui foi
> >>>>>> muita porrada pra tudo que é lado sem conhecimento de causa (não vou ficar
> >>>>>> repetindo aqui o que já foi dito ou quem disse).
>
> >>>>>> Um belo exemplo foi esse do Bruno: se o cara tá fazendo um trampo de
> >>>>>> faculdade ou não não interessa, ele tá expondo um cenário e querendo tirar
> >>>>>> dúvidas!
>
> >>>>>> O fato dele já ter exposto, da forma como expôs, independente da
> >>>>>> finalidade, demonstra um interesse em saber mais, querer aprender, querer
> >>>>>> estudar (OFF: ao contrário dos 73 da USP).
>
> >>>>>> Não que não devamos dar porrada de vez em quando, mas desde que
> >>>>>> entrei nessa lista me senti num UFC.
>
> >>>>>> Enfim, pode mandar as porradas em mim agora; sei que deve ter alguém
> >>>>>> aí que tá doido pra fazer isso.
>
> >>>>>> Menos pô!
>
> >>>>>> 2011/11/10 Bernardo Bosak de Rezende <bernardosal...@gmail.com>
>
> >>>>>> Bah que tristeza, quase ninguém interessado em discutir a
> >>>>>> modelagem...
>
> >>>>>> Vinicius, poderias explicar tua afirmação?
>
> >>>>>> Atenciosamente,
>
> >>>>>> 2011/11/9 Vinicius Quaiato <vinicius.quai...@gmail.com>
>
> >>>>>>> vamos por partes: se você usa enumerações para representar
> >>>>>>> diferentes *tipos, *considere que há algo errado.
>
> >>>>>>> On Wed, Nov 9, 2011 at 3:25 PM, Bruno Costa <
> >>>>>>> brunocosta....@gmail.com> wrote:
>
> >>>>>>>> Caros, gostaria da opinião de vocês sobre a modelagem do seguinte
> >>>>>>>> cenário:
>
> >>>>>>>>> Os *transportes *considerados empresa são: *aéreo*, *terrestre*,
> >>>>>>>>> *marítimo* e *fluvial*. O transporte aéreo é realizado através
> >>>>>>>>> de *aviões *de transporte de passageiros. O transporte terrestre
> >>>>>>>>> pode ser realizado de* trem, veículo alugado, táxi e ônibus*. O
> >>>>>>>>> transporte marítimo e fluvial podem ser de *barco ou lancha*. As
> >>>>>>>>> características relevantes destes transportes são a* tempo de
> >>>>>>>>> viagem e o preço* (varia de acordo com o itinerário e com a
> >>>>>>>>> empresa prestadora do serviço).
>
> >>>>>>>>> Foi feita uma classe pai, [Transporte] com os atributos <tempo> e
> >>>>>>>>> <preco>. Três subclasses: [aereo], [terrestre], [maritmo] e [fluvial], que
> >>>>>>>>> herdam de [Trasporte] e em cada uma, enumerações com os subtipos de
> >>>>>>>>> transporte aéreo (aviao), terrestre (trem, veículo alugado, táxi e ônibus),
> >>>>>>>>> etc.
>
> >>>>>>>>> Segue em anexo o diagrama parcial, para uma idéia mais clara.
>
> >>>>>>>>> Bruno.
>
> >>>>>>>>> --
> >>>>>>>>> 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>
>
> >>>>>>>>> --
> >>>>>>>>> Bernardo Bosak de Rezende
> >>>>>>>>> <http://bernardorezende.net/>- Arquitetura e Desenvolvimento
> >>>>>>>>> <https://github.com/bernardobrezende/>- GitHub
>
> >>>>>>>>> "The weather-cock on the church spire, though made of iron, would
> >>>>>>>>> soon be broken by the storm-wind if it did not understand the noble art of
> >>>>>>>>> turning to every wind."
> >>>>>>>>> Heinrich Heine
>
> >>>>>>>>> --
> >>>>>>>>> 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>
>
> >>>>>>>>> --
> >>>>>>>>> "We're not yet where we can go!"
> >>>>>>>>> "I love email footer phrase!"
>
> >>>>>>>>> --
> >>>>>>>>> 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
>
> ...
>
> mais »

--
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>

Breno Ferreira <bferreira@windowslive.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Thu, Nov 10, 2011 at 1:12 PM

Enums conseguem ter um certo comportamento via Extension Methods.
Não que ache isso uma solução tão elegante quanto a do Java, mas dá p/ fazer.

Breno Ferreira
Twitter: [@breno_ferreira](https://twitter.com/breno_ferreira)
Blog: <http://brenoferreira.wordpress.com>

From: me@juanlopes.net
Date: Thu, 10 Nov 2011 13:12:52 -0200
Subject: Re: [dotnetarchitects] Opinião sobre modelagem de objetos
To: dotnetarchitects@googlegroups.com
[Quoted text hidden]
[Quoted text hidden]

Juan Lopes <me@juanlopes.net>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Thu, Nov 10, 2011 at 1:17 PM

Extension Methods são métodos estáticos. Pense nisso quando for usá-los.

2011/11/10 Breno Ferreira <bferreira@windowslive.com>
[Quoted text hidden]
[Quoted text hidden]

dm.gandini@gmail.com <dm.gandini@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Thu, Nov 10, 2011 at 1:20 PM

@Quaiato Eu sinto que esse caso cheira a Strategy. Pq a deixa mas isolado cada comportamento enquanto State eu vejo mais como fazendo um pedido indo de um lado pro outro. O exemplo classico do Strategy é calculo de imposto de renda com um com PF e outra PJ.

@Juan para ser mega chato e porco. Eu copiaria o comportamento do JAVA usando um extension method para o tipo da enum =P .Net 1 X java 1

@Chatos e cri cri de plantão

Extinção, segundo BOCK, FURTADO e TRASSI (2002, p.52) “é um procedimento no qual uma resposta deixa abruptamente de ser reforçada. Como consequência, a resposta diminuirá de frequência e até mesmo poderá deixar de ser emitida”.

<http://en.wikipedia.org/wiki/Behaviorism>

Pare de ofender as pessoas com trolagens. Se ninguém se interessar a thread morre

Deivison M. Gandini
dm.gandini@gmail.com | msn: dm_gandini@hotmail.com
Before printing, think about your responsibility and commitment to the ENVIRONMENT!

Antes de imprimir pense em sua responsabilidade e compromisso com o MEIO AMBIENTE!

2011/11/10 Juan Lopes <me@juanlopes.net>

[Quoted text hidden]

[Quoted text hidden]

Breno Ferreira <bferreira@windowslive.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Thu, Nov 10, 2011 at 1:23 PM

Por isso disse que não são uma solução elegante ;)
E, sinceramente, acho que se Enums precisam de comportamento, utilizar Enums não é o mais apropriado.
Classes são mais apropriadas p/ esses casos.

Breno Ferreira
Twitter: [@breno_ferreira](https://twitter.com/breno_ferreira)
Blog: <http://brenoferreira.wordpress.com>

From: me@juanlopes.net

Date: Thu, 10 Nov 2011 14:17:06 -0200

[Quoted text hidden]

[Quoted text hidden]

Renato Cantarino <renato.cantarino@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Thu, Nov 10, 2011 at 1:23 PM

ui Malvadao.
Jogo purpurina no grupo.

[Quoted text hidden]

[Quoted text hidden]

dm.gandini@gmail.com <dm.gandini@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Thu, Nov 10, 2011 at 1:24 PM

Olha a il de uma enum. São constantes por isso não vejo problema de colocar um extension method pra elas. É tudo o Framework quebrando galhos pra nós

```
.class private auto ansi sealed BoundaryType
    extends [mscorlib]System.Enum
{
    .field public static literal valuetype System.Net.BoundaryType Chunked = int32(1)

    .field public static literal valuetype System.Net.BoundaryType ContentLength =
int32(0)

    .field public static literal valuetype System.Net.BoundaryType Invalid = int32(5)

    .field public static literal valuetype System.Net.BoundaryType Multipart = int32(3)

    .field public static literal valuetype System.Net.BoundaryType None = int32(4)

    .field public specialname rtspecialname int32 value__

}
```

Deivison M. Gandini

dm.gandini@gmail.com | msn: dm_gandini@hotmail.com

Before printing, think about your responsibility and commitment to the ENVIRONMENT!

Antes de imprimir pense em sua responsabilidade e compromisso com o MEIO AMBIENTE!

2011/11/10 Juan Lopes <me@juanlopes.net>

Extension Methods são métodos estáticos. Pense nisso quando for usá-los.

[Quoted text hidden]

[Quoted text hidden]

Pedro Reys <pedroreys@gmail.com>

Thu, Nov 10, 2011 at 1:25 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Nós também já sentimos falta de ter enumerations parecidos com o que Java possui.

Terminamos por [escrever a nossa](#).

- Pedro Reys

2011/11/10 dm.gandini@gmail.com <dm.gandini@gmail.com>

[Quoted text hidden]

[Quoted text hidden]

Juan Lopes <me@juanlopes.net>

Thu, Nov 10, 2011 at 1:25 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Mais uma vez, se vocês acham que extension methods são solução para adicionar comportamento a uma classe, estão falhando em entender encapsulamento.

2011/11/10 dm.gandini@gmail.com <dm.gandini@gmail.com>

@Quaiato Eu sinto que esse caso cheira a Strategy. Pq a deixa mas isolado cada comportamento enquanto State eu vejo mais como fazendo um pedido indo de um lado pro outro. O exemplo classico do Strategy é calculo de imposto de renda com um com PF e outra PJ.

[Quoted text hidden]

[Quoted text hidden]

Juan Lopes <me@juanlopes.net>

Thu, Nov 10, 2011 at 1:26 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Concordo com esse approach, Pedro. Apesar de isso ser muita coisa, menos "enumerável" :P

2011/11/10 Pedro Reys <pedroreys@gmail.com>

[Quoted text hidden]

[Quoted text hidden]

Roberta Arcoverde <roberta.arcverde@gmail.com>

Thu, Nov 10, 2011 at 1:28 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Eu estou com o Juan, acho que enums em Java são decentes do que em C#. Além de menos verbosos.

[Quoted text hidden]

—

Roberta Lopes Arcoverde

[Quoted text hidden]

Vinicius Quaiato <vinicius.quaiato@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Thu, Nov 10, 2011 at 1:28 PM

DEPENDE.

[Quoted text hidden]

Roberta Arcoverde <roberta.arcoverde@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Thu, Nov 10, 2011 at 1:28 PM

são mais decentes*

2011/11/10 Roberta Arcoverde <roberta.arcoverde@gmail.com>

[Quoted text hidden]

[Quoted text hidden]

dm.gandini@gmail.com <dm.gandini@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Thu, Nov 10, 2011 at 1:28 PM

Eu acho que um é um meio de colocar um "comportamento" em um value type.

[Quoted text hidden]

Vinicius Quaiato <vinicius.quaiato@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Thu, Nov 10, 2011 at 1:30 PM

Acho que a galera ainda está pecando em entender encapsulamento E herança/polimorfismo E composição.

[Quoted text hidden]

Juan Lopes <me@juanlopes.net>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Thu, Nov 10, 2011 at 1:30 PM

E como você acessa o estado interno do objeto a partir de um extension method? Ou deixa tudo público?

Por isso falo sobre encapsulamento.

[Quoted text hidden]

[Quoted text hidden]

dm.gandini@gmail.com <dm.gandini@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Thu, Nov 10, 2011 at 1:34 PM

Da pra ter membros privados dentro de uma enum em Java? #desculpeAlgnorancia

[Quoted text hidden]

Juan Lopes <me@juanlopes.net>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Thu, Nov 10, 2011 at 1:35 PM

Sim.

[Quoted text hidden]

Breno Ferreira <bferreira@windowslive.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Thu, Nov 10, 2011 at 1:36 PM

Como o Quaiato falou, Enums são para armazenar valores (ou seja, são uma maneira "bonita" de dar nomes a inteiros). Se seus enums tão precisando de comportamento, acho q tem alguma coisa errada, e Enums

não são a solução.

Breno Ferreira

Twitter: [@breno_ferreira](#)

Blog: <http://brenoferreira.wordpress.com>

From: me@juanlopes.net

Date: Thu, 10 Nov 2011 14:30:54 -0200

Subject: Re: [dotnetarchitects] Re: Opinião sobre modelagem de objetos

To: dotnetarchitects@googlegroups.com

[Quoted text hidden]

[Quoted text hidden]

Vinicius Quaiato <vinicius.quaiato@gmail.com>

Thu, Nov 10, 2011 at 1:37 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

novamente: por isso temos classes :P

[Quoted text hidden]

Vinicius Quaiato <vinicius.quaiato@gmail.com>

Thu, Nov 10, 2011 at 1:37 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Até que enfim!

[Quoted text hidden]

Juan Lopes <me@juanlopes.net>

Thu, Nov 10, 2011 at 1:38 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Então por que temos delegates se podemos ter one method interfaces?

Por que temos value objects se podemos ter só reference objects?

Estamos falando de linguagem, Quaiato.

2011/11/10 Vinicius Quaiato <vinicius.quaiato@gmail.com>

novamente: por isso temos classes :P

[Quoted text hidden]

[Quoted text hidden]

Breno Ferreira <bferreira@windowslive.com>

Thu, Nov 10, 2011 at 1:41 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Sinceramente, não entendi o q delegates tem a ver com interfaces?

Breno Ferreira

Twitter: [@breno_ferreira](#)

Blog: <http://brenoferreira.wordpress.com>

From: me@juanlopes.net

Date: Thu, 10 Nov 2011 14:38:28 -0200

Subject: Re: [dotnetarchitects] Re: Opinião sobre modelagem de objetos

To: dotnetarchitects@googlegroups.com

[Quoted text hidden]

[Quoted text hidden]

dm.gandini@gmail.com <dm.gandini@gmail.com>

Thu, Nov 10, 2011 at 1:42 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

eu li

<http://download.oracle.com/javase/tutorial/java/javaOO/enum.html>

Tem até uma main dentro de enum. oO. Da pra fazer mas não usaria não.

Deivison M. Gandini

dm.gandini@gmail.com | msn: dm_gandini@hotmail.com

Before printing, think about your responsibility and commitment to the ENVIRONMENT!

Antes de imprimir pense em sua responsabilidade e compromisso com o MEIO AMBIENTE!

2011/11/10 Vinicius Quaiato <vinicius.quaiato@gmail.com>

Até que enfim!

[Quoted text hidden]

[Quoted text hidden]

Juan Lopes <me@juanlopes.net>

Thu, Nov 10, 2011 at 1:44 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Delegates são single-method interfaces com açúcar sintático. O que exatamente você não entendeu?

2011/11/10 Breno Ferreira <bferreira@windowslive.com>

[Quoted text hidden]

[Quoted text hidden]

Daniel Freire <daniel.chc@gmail.com>

Thu, Nov 10, 2011 at 1:45 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

E pq codificar comportamentos num enum e não em uma classe?

[Quoted text hidden]

—

Daniel C. Freire

(31) 8467-2331

@danielfreire

[Quoted text hidden]

dm.gandini@gmail.com <dm.gandini@gmail.com>

Thu, Nov 10, 2011 at 1:45 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Para que do...while se pode usar um goto LOL

Deivison M. Gandini

dm.gandini@gmail.com | msn: dm_gandini@hotmail.com

Before printing, think about your responsibility and commitment to the ENVIRONMENT!

Antes de imprimir pense em sua responsabilidade e compromisso com o MEIO AMBIENTE!

2011/11/10 Breno Ferreira <bferreira@windowslive.com>

[Quoted text hidden]

[Quoted text hidden]

Juan Lopes <me@juanlopes.net>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Thu, Nov 10, 2011 at 1:50 PM

Por que codificar comportamentos numa classe em C# e não em funções em ANSI C que recebem como parâmetro o ponteiro para uma struct alocada dinamicamente?

É fácil não ver uso para algo que você nunca usou.

2011/11/10 Daniel Freire <daniel.chc@gmail.com>

[Quoted text hidden]

--

Daniel C. Freire
(31) 8467-2331
@danielfreire

--

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]

Vinicius Quaiato <vinicius.quaiato@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Thu, Nov 10, 2011 at 1:52 PM

mas Juan por que você está tão reativo?

O Breno perguntou o pq de fazer isso... nos mostre o pq...

Pois para mim é exatamente o mesmo que eu disse sobre usar enum para distinguir tipos: não há necessidade.

Show us the light.

[Quoted text hidden]

[Quoted text hidden]

Daniel Freire <daniel.chc@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Thu, Nov 10, 2011 at 1:52 PM

É justamente isso que eu gostaria de saber. Pq?

[Quoted text hidden]

--

Daniel C. Freire
[Quoted text hidden]

Fernando Munoz <fernando017@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Thu, Nov 10, 2011 at 1:52 PM

Juan concordo que é fácil criticar o uso de algo que nunca usamos, mas por favor, nos fale um exemplo de ENUM que precise de comportamento.
Acho que muita gente está curiosa para saber...

Em 10 de novembro de 2011 14:50, Juan Lopes <me@juanlopes.net> escreveu:

Por que codificar comportamentos numa classe em C# e não em funções em ANSI C que recebem como

parâmetro o ponteiro para uma struct alocada dinamicamente?

É fácil não ver uso para algo que você nunca usou.

2011/11/10 Daniel Freire <daniel.chc@gmail.com>

E pq codificar comportamentos num enum e não em uma classe?

2011/11/10 dm.gandini@gmail.com <dm.gandini@gmail.com>

eu li

<http://download.oracle.com/javase/tutorial/java/javaOO/enum.html>

Tem até uma main dentro de enum. oO. Da pra fazer mas não usaria não.

Deivison M. Gandini

dm.gandini@gmail.com | msn: dm_gandini@hotmail.com

Before printing, think about your responsibility and commitment to the ENVIRONMENT!

Antes de imprimir pense em sua responsabilidade e compromisso com o MEIO AMBIENTE!

2011/11/10 Vinicius Quaiato <vinicius.quaiato@gmail.com>

Até que enfim!

On Thu, Nov 10, 2011 at 2:36 PM, Breno Ferreira <bferreira@windowslive.com> wrote:

Como o Quaiato falou, Enums são para armazenar valores (ou seja, são uma maneira "bonita" de dar nomes a inteiros). Se seus enums tão precisando de comportamento, acho q tem alguma coisa errada, e Enums não são a solução.

Breno Ferreira

Twitter: [@breno_ferreira](https://twitter.com/breno_ferreira)

Blog: <http://brenoferreira.wordpress.com>

[Quoted text hidden]

[Quoted text hidden]

[Quoted text hidden]

[Quoted text hidden]

[Quoted text hidden]

[Quoted text hidden]

[Quoted text hidden]

Vinicius Quaiato <vinicius.quaiato@gmail.com>

Thu, Nov 10, 2011 at 1:53 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

você já foi drenado pelo lado negro da força. JAVISH!

[Quoted text hidden]

Breno Ferreira <bferreira@windowslive.com>

Thu, Nov 10, 2011 at 1:53 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Sinceramente Juan, não vejo Delegates dessa maneira, mas sim tipo function-pointers.

Breno Ferreira

Twitter: [@breno_ferreira](https://twitter.com/breno_ferreira)

Blog: <http://brenoferreira.wordpress.com>

From: me@juanlopes.net

Date: Thu, 10 Nov 2011 14:44:35 -0200

[Quoted text hidden]

[Quoted text hidden]

Roberta Arcoverde <roberta.arcoverde@gmail.com>

Thu, Nov 10, 2011 at 1:59 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

ué, e todos estes exemplos que vocês citaram onde extension methods seriam usados em c# pra atrelar comportamento a enums?

o ayende [já precisou de algo do tipo](#).

[Quoted text hidden]

--

Roberta Lopes Arcoverde

[Quoted text hidden]

Winston Pacheco Junior <winston.pacheco@gmail.com>

Thu, Nov 10, 2011 at 1:59 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Pedro,

Já usei muito esse tipo de classe (lá a gente chamava de Model de Domínio) que você mandou mas a gente nunca deixava os construtores como públicos. Se o cara pode instanciar não dá pra garantir os valores né? Sei lá, a gente só usava pra domínio (como um enumeration), antes de ter attributes no .net (ou talvez antes de eu conhecer os attributes, lá nos idos de 200x.

[Quoted text hidden]

Juan Lopes <me@juanlopes.net>

Thu, Nov 10, 2011 at 2:02 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Exemplo de cabeça, fictício

```
public enum Estado {
    RJ(Regiao.SUDESTE) {
        @Override
        public void applyAccent(Person person) {
            person.replaceSoundOf('s', 'x');
        }
    },

    SP(Regiao.SUDESTE) {
        @Override
        public void applyAccent(Person person) {
            person.doubleDurationOf('r');
        }
    },

    BA(Regiao.NORDESTE) {
        @Override
        public void applyAccent(Person person) {
            person.speaksSinging();
        }
    }
};

private final Regiao regiao;
private Estado(Regiao regiao) {
    this.regiao = regiao;
```

```
}
```

```
public abstract void applyAccent(Person person);
```

[Quoted text hidden]

[Quoted text hidden]

Vinicius Quaiato <vinicius.quaiato@gmail.com>

Thu, Nov 10, 2011 at 2:05 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

ORLY?

[Quoted text hidden]

dm.gandini@gmail.com <dm.gandini@gmail.com>

Thu, Nov 10, 2011 at 2:08 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Eu leio nisso uma enum e uma classe. E uma classe que viola o principio de open-close por sinal. Mesmo que o .Net tivesse isso eu ainda acho bacana deixar separado um do outro.

Agora que eu dou o braço a torcer é uma enum que comporta mais de um valor. Ai matou a pau. Isso sim bateu uma invejinha

[Quoted text hidden]

Juan Lopes <me@juanlopes.net>

Thu, Nov 10, 2011 at 2:08 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

É uma forma de ver. Mas lembre-se que delegates em .net podem ter closures associadas. Simple function pointers não funcionam assim.

[Quoted text hidden]

Juan Lopes <me@juanlopes.net>

Thu, Nov 10, 2011 at 2:13 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Aliás, a própria Microsoft compara os dois na MSDN.

<http://msdn.microsoft.com/en-us/library/ms173173.aspx>

[Quoted text hidden]

[Quoted text hidden]

Juan Lopes <me@juanlopes.net>

Thu, Nov 10, 2011 at 2:14 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Viola o open-closed? Uma classe abstrata que viola o OCP?

[Quoted text hidden]

dm.gandini@gmail.com <dm.gandini@gmail.com>

Thu, Nov 10, 2011 at 2:22 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Me coloca o espirito santo e seu comportamento dentro do `Estado` sem modifica-la?

Usando uma estratégia da vida eu colocava mais um tipo na enum (que pra mim não é classe) e criaria mais uma classe responsável pelo comportamento.

Como eu disse eu vejo duas coisas ai e duas responsabilidades. Não vi vantagem em quebrar esse paradigma ainda.

[Quoted text hidden]

Juan Lopes <me@juanlopes.net>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Thu, Nov 10, 2011 at 2:24 PM

Os valores de um enum são campos estáticos do mesmo tipo do enum. Mudar os campos estáticos não é mudar o comportamento da classe.

[Quoted text hidden]

dm.gandini@gmail.com <dm.gandini@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Thu, Nov 10, 2011 at 2:35 PM

Eu penso a mesma coisa pra uma .net Enum que não tem métodos. Agora quando tem métodos junto com a enum, como eu te disse, eu vejo que existem duas coisas vivendo juntas. Que no meu universo não é normal.

Logo eu nem posso usar estados no meu cadastro sem sem tem que colocar um sotaque novo.

Cara, não leva pro lado pessoal só estou explorando o assunto e aprendendo com vc.

[Quoted text hidden]

Juan Lopes <me@juanlopes.net>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Thu, Nov 10, 2011 at 2:43 PM

Pois é, então agora imagine que você tem uma classe pessoa que tem uma situação. Essa situação pode ser "Ativo", "Inativo", "Pendente", etc., que são enums.

Prefere isso:

```
class Pessoa {
    Situacao situacao;
    public decimal calcularMultaPorAtrasoDe(int dias) {
        if (situacao == Situacao.Ativo)
            return 0.20 * dias;
        else if (situacao == Situacao.Inativo)
            return 0.30 * dias;
        else if (situacao == Situacao.Pendente)
            return 0.40 * dias;
    }
}
```

Ou isso:

```
enum Situacao {
    Ativo(0.20), Inativo(0.30), Pendente(0.40);

    decimal multaPorDia;
    Situacao(decimal multaPorDia) {
        this.multaPorDia = multaPorDia;
    }

    public decimal calcularMultaPorAtrasoDe(int dias) {
        return multaPorDia * dias;
    }
}

class Pessoa {
    Situacao situacao;
    public decimal calcularMultaPorAtrasoDe(int dias) {
        return situacao.calcularMultaPorAtrasoDe(dias);
    }
}
```

[Quoted text hidden]

[Quoted text hidden]

Vinicius Quaiato <vinicius.quaiato@gmail.com>

Thu, Nov 10, 2011 at 2:47 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Pra mim dá na mesma do state que eu postei :P

[Quoted text hidden]

Juan Lopes <me@juanlopes.net>

Thu, Nov 10, 2011 at 2:48 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Sim, dá, mas com ajuda da linguagem é melhor, não?

Como eu disse, delegates e one method interfaces dão na mesma, mas a sintaxe de delegates é menos verbosa.

[Quoted text hidden]

[Quoted text hidden]

Breno Ferreira <bferreira@windowslive.com>

Thu, Nov 10, 2011 at 3:00 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Aplicando o que o Quaiato escreveu no blog dele:

```
public abstract class Situacao
{
    public abstract decimal AplicaMulta(int dias);
}
public class Ativo : Situacao
{
    public override decimal AplicaMulta(int dias);
    {
        return .20 * dias
    }
}
public class Pendente : Situacao
{
    public override decimal AplicaMulta(int dias);
    {
        return .40 * dias
    }
}
public class Inativo : Situacao
{
    public override decimal AplicaMulta(int dias);
    {
        return .30 * dias
    }
}
public class Conta
{
    public Situacao { get; set; }

    public decimal CalcularMulta()
    {
        return this.Situacao.AplicaMulta();
    }
}
```

Ao meu ver, acho essa uma implementação mais apropriada com a realidade, pois "Situação" é algo a mais

que apenas guardar um int, neste caso, Situação seria uma classe e não um Enum. Mas, com a maneira que é feita em Java (que não conhecia, mas achei legal!), não acho feia. E ao q me parece em Java, Enums se comportam mais como objetos do que um simples inteiro representado por um nome.

Breno Ferreira

Twitter: [@breno_ferreira](#)

Blog: <http://brenoferreira.wordpress.com>

From: me@juanlopes.net

Date: Thu, 10 Nov 2011 15:48:39 -0200

[Quoted text hidden]

[Quoted text hidden]

dm.gandini@gmail.com <dm.gandini@gmail.com>

Thu, Nov 10, 2011 at 3:00 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Juan, eu entendi o seu ponto de vista. Seu jeito é bacana.

Mas usaria uma classe + a enum, Principalmente em uma enum mais utilizada no sistema.

[Quoted text hidden]

Juan Lopes <me@juanlopes.net>

Thu, Nov 10, 2011 at 3:03 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Ou, aplicando a reversal russa,

Na Russia, digo, no .NET, os enums que se comportam mais como tipos numéricos que como objetos.

[Quoted text hidden]

[Quoted text hidden]

dm.gandini@gmail.com <dm.gandini@gmail.com>

Thu, Nov 10, 2011 at 3:06 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Na "Russia" o único Value type com comportamento é a struct

[Quoted text hidden]

Thiago Menezes Cristo <thiagomcristo@gmail.com>

Thu, Nov 10, 2011 at 3:12 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Na Russia existe as [Medvedev Girls](#). =)

Thiago de Menezes Cristo

[Quoted text hidden]

Breno Ferreira <bferreira@windowslive.com>

Thu, Nov 10, 2011 at 3:16 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Realmente, a Russia é uma país de belos atributos!

Breno Ferreira

Twitter: [@breno_ferreira](#)

Blog: <http://brenoferreira.wordpress.com>

From: thiagomcristo@gmail.com
Date: Thu, 10 Nov 2011 16:12:43 -0200
[Quoted text hidden]
[Quoted text hidden]

dm.gandini@gmail.com <dm.gandini@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Thu, Nov 10, 2011 at 3:16 PM

Umm pesquisando é o Java que diferente mesmo.
Não achei nada parecido em C++ tb.

Deivison M. Gandini
dm.gandini@gmail.com | msn: dm_gandini@hotmail.com
Before printing, think about your responsibility and commitment to the ENVIRONMENT!
Antes de imprimir pense em sua responsabilidade e compromisso com o MEIO AMBIENTE!

2011/11/10 Thiago Menezes Cristo <thiagomcristo@gmail.com>
[Quoted text hidden]

[Quoted text hidden]

dm.gandini@gmail.com <dm.gandini@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Thu, Nov 10, 2011 at 3:18 PM

Essas minas não querem concorrência, tão tirando as armas das barangas!

Deivison M. Gandini
dm.gandini@gmail.com | msn: dm_gandini@hotmail.com
Before printing, think about your responsibility and commitment to the ENVIRONMENT!
Antes de imprimir pense em sua responsabilidade e compromisso com o MEIO AMBIENTE!

[Quoted text hidden]

[Quoted text hidden]

Pedro Reys <pedroreys@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: "dotnetarchitects@googlegroups.com" <dotnetarchitects@googlegroups.com>

Thu, Nov 10, 2011 at 3:22 PM

Wiston, eu concordo com você com relação ao ctor. Atualizarei o Readme, obrigado

Sent from my phone.

[Quoted text hidden]

[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]

Pedro Reys <pedroreys@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: "dotnetarchitects@googlegroups.com" <dotnetarchitects@googlegroups.com>

Fri, Nov 18, 2011 at 5:00 PM

Patrick Lioi, meu colega aqui na Headspring escreveu [um post hoje explicando os motivos porque optamos por usar essa classe Enumeration](#) ao invés de enums..

Nos exemplos dele o construtor é privado, Winston.

- Pedro Reis

2011/11/10 Pedro Reis <pedroreys@gmail.com>

[Quoted text hidden]

[Quoted text hidden]

Danimar Ribeiro <danimaribeiro@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: ".Net Architects" <dotnetarchitects@googlegroups.com>

Mon, Nov 21, 2011 at 2:51 PM

Muito legal as duas abordagens.

A meu ponto de vista, a segunda abordagem parece mais concisa. Porém.
O que fazer se essa Situação da Pessoa, não apenas funciona para o valor da multa?

Se essa Situação deve ser verificada, para verificar se efetuo a venda para o cliente, se o cliente merece um desconto, enfim, em um sistema com várias regras aplicadas á situação da pessoa. Como faço, crio vários métodos para todos esses dentro da classe, ou dentro do enum? E se essa verificação possui várias dependências?

Como vocês fariam em casos como este?

Danimar Ribeiro

[Quoted text hidden]

Aplicando o que o Quaiato escreveu no blog dele: `public abstract class Situacao`

```
{  
    public abstract decimal AplicaMulta(int dias);  
}
```

```
public class Ativo : Situacao  
{  
    public override decimal AplicaMulta(int dias);  
    {  
        return .20 * dias  
    }  
}
```

```
public class Pendente : Situacao  
{  
    public override decimal AplicaMulta(int dias);  
    {  
        return .40 * dias  
    }  
}
```

```
public class Inativo : Situacao  
{  
    public override decimal AplicaMulta(int dias);  
    {  
        return .30 * dias  
    }  
}
```

```
public class Conta  
{  
    public Situacao { get; set; }  
}
```

```
public decimal CalcularMulta()  
{  
    return this.Situacao.AplicaMulta();  
}
```

—
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
[Quoted text hidden]

Vinicius Quaiato <vinicius.quaiato@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Mon, Nov 21, 2011 at 4:10 PM

Ou vc faz essa verificação no load do objeto, o que é mais custoso... ou faz isso na forma de consolidação.

Mas nesse caso sim pode-se usar um enum para representar esse status...

On Mon, Nov 21, 2011 at 3:51 PM, Danimar Ribeiro <danimaribeiro@gmail.com> wrote:

Muito legal as duas abordagens.
A meu ponto de vista, a segunda abordagem parece mais concisa. Porém.
O que fazer se essa Situação da Pessoa, não apenas funciona para o valor da multa?
Se essa Situação deve ser verificada, para verificar se efetuo a venda para o cliente, se o cliente merece um desconto, enfim, em um sistema com várias regras aplicadas á situação da pessoa. Como faço, crio vários métodos para todos esses dentro da classe, ou dentro do enum? E se essa verificação possui várias dependências?

Como vocês fariam em casos como este?

Danimar Ribeiro

[Quoted text hidden]

[Quoted text hidden]

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>

Danimar Ribeiro <danimaribeiro@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: ".Net Architects" <dotnetarchitects@googlegroups.com>

Mon, Nov 21, 2011 at 4:15 PM

Desculpe mas não entendi direito.
Poderia explicar melhor?

Danimar Ribeiro

On 21 nov, 17:10, Vinicius Quaiato <vinicius.quaiato@gmail.com> wrote:
> Ou vc faz essa verificação no load do objeto, o que é mais custoso... ou
> faz isso na forma de consolidação.
>
> Mas nesse caso sim pode-se usar um enum para representar esse status...
>
> On Mon, Nov 21, 2011 at 3:51 PM, Danimar Ribeiro <danimaribeiro@gmail.com> wrote:

—
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
[Quoted text hidden]

Juliano Nunes <juliano.net@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Mon, Nov 21, 2011 at 5:08 PM

Comparando os códigos do Juan e do Quaiato, a impressão que eu tenho é que o "modo Java" trata-se de um "atalho de sintaxe". Isso teria algum outro benefício que não evitar uma chuva de classes dependendo da situação (no caso do modo C#)?

E usando o enum para implementações não estaria atribuindo a ele responsabilidades demais? Colocar regras de negócio dentro de um enum me soa estranho.

[Quoted text hidden]

Juan Lopes <me@juanlopes.net>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Mon, Nov 21, 2011 at 5:15 PM

Ter dados sem comportamento me parece mais estranho ainda.

Os bons programadores C# evitam enums justamente por ele geralmente levar à quebra da lei de Demeter.

Muitas vezes, um mero atalho de linguagem muda toda a forma como os programadores daquela linguagem pensam. Em C++ é prática passar parâmetros por referência. Em C# isso não é bem visto, pois sem const-correctness, passar parâmetros por referência leva a queda da coesão.

Yield return no C# é um mero atalho de linguagem para uma máquina de estados. Esse atalho mudou radicalmente a forma como programamos atualmente. Assim como delegates. Ou value types que herdam de object.

2011/11/21 Juliano Nunes <juliano.net@gmail.com>
[Quoted text hidden]

[Quoted text hidden]

Juliano Nunes <juliano.net@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Mon, Nov 21, 2011 at 5:44 PM

Mas dados não possuem comportamento (ou ao menos não deveriam), seu propósito a meu ver é ser consumido, gerado, etc. por um objeto. Por isso concordo com a abordagem de enum + classe.

Quanto ao atalho de linguagem, sei que o C# é mestre nisso (quando digo mestre é justamente por haver muitas implementações desta forma na linguagem) e pra falar a verdade, dou graças a Deus por isso. Não acho que seja uma prática ruim colocar esses "atalhos".

[Quoted text hidden]