



Fabrício Cabral <fabriciofx@gmail.com>

Instanciar um membro de uma classe no corpo ou na assinatura de um construtor?

27 messages

Fabrício Cabral <fabriciofx@gmail.com>

Wed, Jul 18, 2012 at 2:54 AM

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

Olá todos!

Tenho uma dúvida um tanto quanto "boba", mas que talvez renda uma boa discussão. Imagine que eu tenho uma classe chamada Pessoa e uma outra chamada CPF. Uma Pessoa possui um nome e um único (no meu domínio) CPF. Pessoa é uma entidade e CPF um objeto de valor e ambos são objetos do meu domínio. A assinatura do construtor é CPF(string numeroCpf), onde numeroCpf é o número do CPF no formato de uma string. Antes de instanciar esse objeto, a classe faz uma validação para determinar se esse número é válido. Se não for, dispara uma exceção. Até aí, tudo normal, certo?

Bom, a dúvida é a seguinte: como deveria ser a assinatura da classe Pessoa? Deveria ser:

1) Pessoa(string nome, string numeroCpf) e dentro do construtor eu faria um
cpf = new CPF(numeroCpf) e para instanciar um objeto da classe Pessoa eu faria
um: Pessoa p = new Pessoa("João", "111.222.333-44");

ou

2) Pessoa(string nome, CPF cpf) e eu instanciaria um objeto da classe Pessoa
faria um: Pessoa p = new Pessoa("João", new CPF("111.222.333-44"));

Agora não basta apenas dizer se 1) é melhor que 2) ou vice-versa. Tem que explicar por qual razão, motivo ou circunstância um melhor que o outro. Principalmente no que tange a questão do design do sistema.

Agradeço a atenção,

--

-fx

Lemol Moraes Lutonda <leza.ml@fecrd.cujae.edu.cu>

Wed, Jul 18, 2012 at 2:03 AM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Olá,

Em essas condições não vejo diferenças nem problemas em usar um ou outro, e provavelmente eu aplicaria polimorfismo para usar as duas assinaturas.

Atenciosamente,

Leza Moraes Lutonda, Lemol-C

<http://lemolsoft.webs.com><http://github.com/lemol><http://twitter.com/lemolsoft>

EI 18-Jul-12 2:54 AM, Fabrício Cabral escribió:

[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>

Participe en la XVI Convencion Cientifica de Ingenieria y Arquitectura del 26 al 30 de noviembre de 2012. La Habana, Cuba: <http://ccia.cujae.edu.cu>

Consulte la enciclopedia colaborativa cubana. <http://www.ecured.cu>

--
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>

edmilson hora <edmilson_hora@yahoo.com.br>

Wed, Jul 18, 2012 at 7:53 AM

Reply-To: dotnetarchitects@googlegroups.com

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

Fabricio, as duas maneiras são idênticas (a única diferença é onde está instanciando CPF), porém em minha opinião, abandone a ideia de construtor parametrizado para as ENTIDADES do domínio a classe Pessoa numa situação real terá muitos atributos e ficar mudando ou enchendo a classe de construtores não é legal. Os construtores caem melhor para Os VO's como é o caso de CPF(string numero) e para as classes de Serviços como ==> PessoaServico(Pessoa pessoa).

crie suas classes de Entidades de domínio da maneira mais simples possível o uso delas lá na frente será muito mais flexível.

```
classe Pessoa
{
    public int Id {get; set;}
    public string Nome {get; set;}
    public CPF Cpf {get; set;}
    public Endereco Endereco {get; set;}
}
```

Quanto a desing do sistema, as Entidades que não tem comportamentos como é o caso do exemplo, o desing é esse mesmo, classes simples, sem inventar muito, quando aparecerem as Entidades que eventualmente tenham comportamentos, aí sim o desing é modelado de acordo com as regras solicitadas.

De: Fabrício Cabral <fabriciofx@gmail.com>

Para: .Net Architects <dotnetarchitects@googlegroups.com>

Enviadas: Quarta-feira, 18 de Julho de 2012 2:54

Assunto: [dotnetarchitects] Instanciar um membro de uma classe no corpo ou na assinatura de um construtor?

[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]

Joao Bosco <joao.bosco.seixas@gmail.com>

Wed, Jul 18, 2012 at 9:10 AM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Olha, eu vejo muita diferença entre usar uma maneira ou outra.

Talvez pelo fato de se tratar de Pessoa e CPF, nós não estamos dando a devida atenção, mas se extrapolarmos para outros casos podemos ver que isso pode trazer algum desconforto.

1- Se o objeto CPF não fizer sentido para o sistema fora do agregado pessoa, então acho que você deve passar a "string" e dentro do objeto pessoa você instância ele corretamente. Pessoa(string nome, string numeroCpf)

2- Se o objeto CPF faz sentido para todo o sistema, não vejo problema em receber o CPF como parametro. Pessoa(string nome, CPF cpf). Mas é preciso ter cuidado sobre quem tem a responsabilidade de criação do CPF nesse caso.

Uma terceira opção deve ser levada em consideração...

3- Criação de uma Factory para o objeto pessoa que consegue lidar com a complexidade envolvida na criação do objeto.

Talvez isso tudo pareça um exagero para pessoa e cpf, mas a questão não é essa. A questão é a criação de objetos "complexos".

[Quoted text hidden]

—

Atenciosamente,

João Bosco Seixas

Inteligência Digital

<http://www.joaoseixas.com>

@joaoboscoseixas

[Quoted text hidden]

Daniel Moreira Yokoyama <moreira.yokoyama@gmail.com>

Wed, Jul 18, 2012 at 9:39 AM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Por favor,

Me explique o que quis dizer por "usar o polimorfismo"? Não entendi onde tem o polimorfismo aí... quem assume a forma de quem? Pessoa assume forma de CPF? CPF assume forma de Pessoa? Pra mim nenhum dos dois faz sentido.

Atenciosamente,

Daniel Moreira Yokoyama.

@dmyoko

<http://twitter.com/dmyoko>

Blogs:

Getting Sharper (C#, Arquitetura de Software e outras coisas mais)

<http://gettingsharper.wordpress.com/>

Eis o Cristo... e jaz o Cristão (Cristianismo sem Religiosidade)

<http://eisocristo.wordpress.com>

Stay Sharp!

[Quoted text hidden]

[Quoted text hidden]

Daniel Moreira Yokoyama <moreira.yokoyama@gmail.com>

Wed, Jul 18, 2012 at 9:42 AM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Se é uma dependência (required) faz todo sentido colocar o CPF no construtor, Edmilson. Entendi o que você quis dizer mas eu não colocaria no construtor todos os membros, apenas aqueles que fossem requeridos e que não podem ter um valor default. CPF é um bom exemplo disto.

Atenciosamente,

Daniel Moreira Yokoyama.

@dmyoko

<http://twitter.com/dmyoko>

Blogs:

Getting Sharper (C#, Arquitetura de Software e outras coisas mais)

<http://gettingsharper.wordpress.com/>

Eis o Cristo... e jaz o Cristão (Cristianismo sem Religiosidade)

<http://eisocristo.wordpress.com>

Stay Sharp!

Em 18 de julho de 2012 07:53, edmilson hora <edmilson_hora@yahoo.com.br> escreveu:

[Quoted text hidden]

[Quoted text hidden]

Daniel Moreira Yokoyama <moreira.yokoyama@gmail.com>

Wed, Jul 18, 2012 at 9:49 AM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Concordo, João.

Mas eu diria o seguinte:

CPF faz parte do seu domínio. String não é seu domínio, é só uma forma de representar uma informação sem semântica.

Então, o seu construtor tem que ter, no mínimo, conformidade com o seu domínio: Pessoa recebe uma instância de CPF.

Isso é o mínimo esperado para que o seu domínio seja mais harmonioso.

Aí, você até pode, para facilitar sua vida de uma forma geral, criar um outro construtor que receba string (você perde semântica, mas você pode ganhar expressividade usando `p = new Pessoa(numeroCPF = "12345678")`).

Dando nome ao parâmetro no construtor faz com que o código faça sentido, quem lê o código continua sabendo qual sua intenção (intention revealing), ao invés de simplesmente ver uma string solta.

Mas, você pode acabar descobrindo que manter esta string neste construtor pode dar trabalho para manter. Isto pode acontecer. E aí cabe a você descobrir se ainda vale a pena manter a versão string ou não.

Uma coisa é fato: se CPF é uma classe do domínio, então é fundamental que o construtor base receba uma instância dele. O resto é opcional.

Atenciosamente,

Daniel Moreira Yokoyama.
@dmyoko
<http://twitter.com/dmyoko>

Blogs:

Getting Sharper (C#, Arquitetura de Software e outras coisas mais)
<http://gettingsharper.wordpress.com/>

Eis o Cristo... e jaz o Cristão (Cristianismo sem Religiosidade)
<http://eisocristo.wordpress.com>

Stay Sharp!

[Quoted text hidden]

[Quoted text hidden]

Joao Bosco <joao.bosco.seixas@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Wed, Jul 18, 2012 at 9:55 AM

Exato Daniel!

A busca pela expressividade é um dos melhores guias nesses casos.

[Quoted text hidden]

[Quoted text hidden]

Lemol Moraes Lutonda <leza.ml@fecrd.cujae.edu.cu>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Wed, Jul 18, 2012 at 11:03 AM

Fazendo o constructor polimórfico, i.e, que tenha as duas assinaturas.

El 18-Jul-12 9:39 AM, Daniel Moreira Yokoyama escribió:

[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>

Participe en la XVI Convención Científica de Ingeniería y Arquitectura del 26 al 30 de noviembre de 2012. La Habana, Cuba: <http://ccia.cujae.edu.cu>

Consulte la enciclopedia colaborativa cubana. <http://www.ecured.cu>

[Quoted text hidden]

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

Wed, Jul 18, 2012 at 12:26 PM

To: dotnetarchitects@googlegroups.com

Apesar de semanticamente fazer sentido a sua afirmação, Lemol, você está sendo ambíguo.

Polimorfismo tem um outro sentido quando estamos falando de objetos.

2012/7/18 Lemol Morais Lutonda <leza.ml@fecrd.cujae.edu.cu>

[Quoted text hidden]

—
<https://github.com/juanlpopes>

[Quoted text hidden]

Lemol Morais Lutonda <leza.ml@fecrd.cujae.edu.cu>

Wed, Jul 18, 2012 at 11:36 AM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Mas também significa ter métodos com mesmo nome mas assinaturas diferentes dentro da mesma classe.

El 18-Jul-12 12:26 PM, Juan Lopes escribió:

[Quoted text hidden]

[Quoted text hidden]

Juan Lopes <me@juanlopes.net>

Wed, Jul 18, 2012 at 12:40 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Sim, polimorfismo ad-hoc. Mas afim de evitar ambiguidades, as pessoas geralmente chamam de method overloading.

[Quoted text hidden]

Lemol Morais Lutonda <leza.ml@fecrd.cujae.edu.cu>

Wed, Jul 18, 2012 at 11:50 AM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

OK.

Realmente seria mais facil haver dito sobrecarga do constructor.

El 18-Jul-12 12:40 PM, Juan Lopes escribió:

[Quoted text hidden]

[Quoted text hidden]

Daniel Moreira Yokoyama <moreira.yokoyama@gmail.com>

Wed, Jul 18, 2012 at 1:26 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Eu sinceramente nunca entendi por que overloading é chamado por polimorfismo. Para mim uma coisa não tem nada a ver com a outra. Eu nem saquei isso na mensagem original do Lemol, mas agora que vocês falaram, eu me pergunto, não é a primeira vez que vejo pessoas relacionarem overloading com polimorfismo.

Sei lá, o Juan falou que é ambíguo, pra mim nunca foi. Sempre associei isto a péssimos professores que não sabem explicar polimorfismo em OOP (através de overriding, não overloading...).

Sei que isto é off topic, mas eu realmente sempre quis entender esta confusão que tanta gente faz.

Atenciosamente,

Daniel Moreira Yokoyama.
@dmyoko

<http://twitter.com/dmyoko>

Blogs:

Getting Sharper (C#, Arquitetura de Software e outras coisas mais)

<http://gettingsharper.wordpress.com/>

Eis o Cristo... e jaz o Cristão (Cristianismo sem Religiosidade)

<http://eisocristo.wordpress.com>

Stay Sharp!

[Quoted text hidden]

[Quoted text hidden]

William Thiago <williamthiago@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Wed, Jul 18, 2012 at 9:41 AM

Acredito que ele queria dizer overload das assinaturas...

[Quoted text hidden]

Rafael Marques <rafaelomarques@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Wed, Jul 18, 2012 at 12:11 PM

Eu usaria CPF no construtor... caso contrario, nem criaria uma classe chamada CPF...
Na minha humilde opiniao, a maior funcao de criar tiny types e a expressividade...

[Quoted text hidden]

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

Wed, Jul 18, 2012 at 2:07 PM

+1

Intention Revealing!

Atenciosamente,

Daniel Moreira Yokoyama.
@dmyoko
<http://twitter.com/dmyoko>

Blogs:

Getting Sharper (C#, Arquitetura de Software e outras coisas mais)

<http://gettingsharper.wordpress.com/>

Eis o Cristo... e jaz o Cristão (Cristianismo sem Religiosidade)

<http://eisocristo.wordpress.com>

Stay Sharp!

[Quoted text hidden]

[Quoted text hidden]

Lemol Moraes Lutonda <leza.ml@fecrd.cujae.edu.cu>

Wed, Jul 18, 2012 at 1:23 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

E em todos casos named parameter é uma boa ideia para este fim.

Atenciosamente,

Leza Moraes Lutonda, Lemol-C

<http://lemolsoft.webs.com>

<http://github.com/lemol>

<http://twitter.com/lemolsoft>

El 18-Jul-12 2:07 PM, Daniel Moreira Yokoyama escribió:

[Quoted text hidden]

[Quoted text hidden]

Daniel Moreira Yokoyama <moreira.yokoyama@gmail.com>

Wed, Jul 18, 2012 at 2:29 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Sim... mas a classe CPF pode incluir, por exemplo, validação.

Atenciosamente,

Daniel Moreira Yokoyama.

@dmyoko

<http://twitter.com/dmyoko>

Blogs:

Getting Sharper (C#, Arquitetura de Software e outras coisas mais)

<http://gettingsharper.wordpress.com/>

Eis o Cristo... e jaz o Cristão (Cristianismo sem Religiosidade)

<http://eisocristo.wordpress.com>

Stay Sharp!

[Quoted text hidden]

[Quoted text hidden]

edmilson hora <edmilson_hora@yahoo.com.br>

Wed, Jul 18, 2012 at 2:31 PM

Reply-To: dotnetarchitects@googlegroups.com

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

Alem da expressividade, tem questões como validação e/ou regras que podem ser importantes para o tiny type.

De: Rafael Marques <rafaelomarques@gmail.com>

Para: dotnetarchitects@googlegroups.com

Enviadas: Quarta-feira, 18 de Julho de 2012 12:11

Assunto: Re: [dotnetarchitects] Instanciar um membro de uma classe no corpo ou na assinatura de um construtor?

Eu usaria CPF no construtor... caso contrario, nem criaria uma classe chamada CPF...
Na minha humilde opiniao, a maior funcao de criar tiny types e a expressividade...
Em 18/07/2012 12:02, "Lemol Moraes Lutonda" <leza.ml@fecrd.cujae.edu.cu> escreveu:
Fazendo o constructor polimórfico, i.e, que tenha as duas assinaturas.

El 18-Jul-12 9:39 AM, Daniel Moreira Yokoyama escribió:

Por favor,

Me explique o que quis dizer por "usar o polimorfismo"? Não entendi onde tem o polimorfismo aí... quem assume a forma de quem? Pessoa assume forma de CPF? CPF assume forma de Pessoa? Pra mim nenhum dos dois faz sentido.

Atenciosamente,

Daniel Moreira Yokoyama.
@dmyoko
<http://twitter.com/dmyoko>

Blogs:
Getting Sharper (C#, Arquitetura de Software e outras coisas mais)
<http://gettingsharper.wordpress.com/>

Eis o Cristo... e jaz o Cristão (Cristianismo sem Religiosidade)
<http://eisocristo.wordpress.com>

Stay Sharp!

Em 18 de julho de 2012 02:03, Lemol Moraes Lutonda
<leza.ml@fecrd.cujae.edu.cu> escreveu:

Olá,
Em essas condições não vejo diferenças nem problemas em usar um ou outro, e provavelmente eu aplicaria polimorfismo para usar as duas assinaturas.

Atenciosamente,

Leza Moraes Lutonda, Lemol-C
<http://lemolsoft.webs.com>

[Quoted text hidden]

Participe en la XVI Convencion Cientifica de Ingenieria y Arquitectura del
26 al 30 de noviembre de 2012. La Habana, Cuba:
<http://ccia.cujae.edu.cu>

Consulte la enciclopedia colaborativa cubana. <http://www.ecured.cu>

--

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>

Participe en la XVI Convención Científica de Ingeniería y Arquitectura del 26 al 30 de noviembre de 2012. La Habana, Cuba: <http://ccia.cujae.edu.cu>

Consulte la enciclopedia colaborativa cubana. <http://www.ecured.cu>

--

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>

[Quoted text hidden]

Felipe Aron <felipearon@gmail.com>

Wed, Jul 18, 2012 at 2:37 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

O que seria um "tiny type"?

[Quoted text hidden]

--

Analista-Programador

#FollowMe: [@felipearon](https://twitter.com/felipearon) (<http://felipearon.net>)

[Quoted text hidden]

Lemol Morais Lutonda <leza.ml@fecrd.cujae.edu.cu>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Wed, Jul 18, 2012 at 1:40 PM

Exacto, mas de qualquer forma o objecto CPF será criado no constructor.

Atenciosamente,

Leza Morais Lutonda, Lemol-C
<http://lemolsoft.webs.com>
<http://github.com/lemol>
<http://twitter.com/lemolsoft>

El 18-Jul-12 2:29 PM, Daniel Moreira Yokoyama escribió:

[Quoted text hidden]

[Quoted text hidden]

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

Wed, Jul 18, 2012 at 2:39 PM

Objetos pequenos e imutáveis que encapsulam e dão comportamento a tipos básicos. CPF é um exemplo.

2012/7/18 Felipe Aron <felipearon@gmail.com>

[Quoted text hidden]

—
<https://github.com/juanplopes>

[Quoted text hidden]

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

Wed, Jul 18, 2012 at 2:46 PM

Aron -> <http://blog.caelum.com.br/pequenos-objetos-imutaveis-e-tiny-types/>

Ultimamente estou adotando a opiniao do Daniel,
Minhas classes ja vem com toda a regra aplicada embutida.

[Quoted text hidden]

—

Att,
Renato Cantarino

[Quoted text hidden]

Felipe Aron <felipearon@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Wed, Jul 18, 2012 at 3:17 PM

Valeu pelas info galera!

[Quoted text hidden]

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

Sun, Jul 22, 2012 at 11:47 PM

Oi Fabricio,

acho que o pessoal já respondeu a sua pergunta sobre o construtor. Só queria aproveitar e dar uma dica sobre o nome do parâmetro.

No seu email você menciona que o parametro do construtor de CPF é "numeroCPF". Apenas como sugestão, você poderia usar apenas "numero", uma vez que, a classe já é CPF.

[]s
Cezar.
cezarguimaraes.com

[Quoted text hidden]

[Quoted text hidden]

[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>

—
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]

Fabício Cabral <fabriciofx@gmail.com>

Mon, Jul 23, 2012 at 11:14 AM

To: dotnetarchitects@googlegroups.com

Olá todos!

Gostaria de agradecer a todos pela colaboração na discussão acerca deste tema. Para mim foi muito proveitoso e apareceram ótimas considerações e sugestões!

Cezar, eu normalmente não uso o nome "numeroCPF". Apenas o usei para ficar mais claro para os membros da lista. Normalmente eu uso apenas "cpf" mesmo, tanto se fosse para usar como string como para um objeto CPF.

Mas bem observado da sua parte também! :)

[]'s

2012/7/22 Cezar Augusto Guimaraes Neto <cezar.alevatto@gmail.com>

[Quoted text hidden]

—
-fx