

Retirei esta tabela do MSDN, e acho(e uso!) interessante essas regras pois ajudam a manter o código mais organizado !!!

Tá certo que nem todos os componentes são iguais e alguns nem existam no java, mas muitos caem como uma luva, como o lbl(Label), btn(Button), txt(TextBox), frm(Form{na verdade é frame ne =P}) e etc ...

Ta aí a tabelinha bonitinha das regras que ajudam a manter um código limpo.

Componente	Prefixo	Exemplo
Form	frm	frmPrincipal
Label	lbl	lblValor
LinkLabel	lnk	lnkEmail
Button	btn	btnSair
TextBox	txt	txtSalario
Menu	mnu	mnuFileOpen
CheckBox	chk	chkConfirma
RadioButton	rad	radCasado
GroupBox	grp	grpEstCivil
PictureBox	pic	picAvatar
Panel	pnl	pnlEstCivil
DataGrid	grd	grdQueryResult
ListBox	lst	lstPolicyCodes
CheckedListBox	clb	clbOptions
ComboBox	cbo	cboEnglish
ListView	lvw	lvwHeadings
TreeView	tre	treOrganization
TabControl	tbc	tbcOptions
DateTimePicker	dtp	dtpPublished
MonthCalendar	mcl	mclPeriod
HScrollBar	hsb	hsbMove
VScrollBar	vsb	vsbMove
Timer	tmr	tmrAlarm
Splitter	spt	sptDivision
DomainUpDown	upd	updPages
NumericUpDown	nud	nudPieces
TrackBar	trb	trbIndex
ProgressBar	prg	prgLoadFile
RichTextBox	rtf	rtfReport
ImageList	ils	ilsAllIcons
HelpProvider	hlp	hlpOptions
ToolTip	tip	tipIcons
ContextMenu	cmn	cmnOpen
ToolBar	tlb	tlbActions
StatusBar	sta	staDateTime
NotifyIcon	nti	ntiOpen
OpenFileDialog	ofd	ofdImage
SaveFileDialog	sfd	sfdImage

FontDialog	ftd	ftdText
ColorDialog	cld	cldText
PrintDialog	ptd	ptdText
PrintPreviewDialog	ppd	ppdText
PrintPreviewControl	ppc	ppcText
ErrorProvider	err	errOpen
PrintDocument	prn	prnText
PageSetup Dialog	psd	psdReport
CrystalReportViewer	rpt	rptSales

Data

Componente	Prefixo	Exemplo
DataSet	dts	dtsProducts
OleDbDataAdapter	oda	odaClients
OleDbConnection	ocn	ocnClients
OleDbCommand	ocm	ocmConsult
SqlDataAdapter	sda	sdaClients
SqlConnection	scn	scnClients
SqlCommand	scm	scmConsult
DataView	dtv	dtvConsult

Como tirei essas regras da msdn(Microsoft Developers Networks) elas se aplicam mais em C# e VB.NET. Mas são sim adaptáveis ao Java.

Outro importante tópico a ser debatido neste post.

Segue abaixo algumas maneiras e metodologias que ao meu ponto de vista são ótimas não só quando em equipe, mas a facilidade de manutenção e qualidade do código.

Primeiro, vou falar sobre o padrão de nomenclatura que são os seguintes:

Pascal case

A primeira letra de cada palavra é maiuscula e as restantes minusculas:BackColor

Camel case

Primeira palavra minuscula e as próximas são em Pascal case:backColor

Uppercase

Maiuscula são as letras de palavras que foram abreviadas:System.IO, System.Web.UI, IDisposable

Lembrando que o .Net é Case Sensitivity, ou seja, sabe diferenciar maiúsculo de minúsculo: calcula(); Calcula();

Naming Guidelines

Abreviações

Não utilizar abreviações como: OpenWin() ao invés de usar OpenWindow(), são apenas poucas letras e isso faz muita diferença quando outro desenvolvedor ou até você mesmo for ler o código para alterações ou correção de bugs.

Namespace

Deve ser sempre em Pascal case, pois também dará final a uma DLL (DiarioFonte.Blog.Core.dll).

NomeEmpresa.NomeTecnologia[.Feature][.Design]

DiarioFonte.Blog.Core

Classe

Também deve ser sempre em Pascal case, pois ao referenciar uma variável como está classe, fará toda a diferença.

```
public Class accessControl{}  
accessControl acesso = new accessControl();
```

Agora vejamos:

```
public Class AccessControl{}  
AccessControl acesso = new AccessControl();
```

Com certeza a segunda forma é muito mais legível, e imagine instanciarmos: sqlConnection, textBox... não tem cara de classe, concordam?

Interface

Sempre use o "I" para iniciar a Interface, assim em seu código fica muito fácil tanto implementá-la quanto utilizar em seu código. (No caso, o I Seria UpperCase, porém é apenas uma letra, e o restante utiliza o padrão de nomenclatura da Classe)

```
public interface IControlManager{}  
public Class AccessControlManager : IControlManager {}
```

Enum

Pascal case. Compare: AccessControl.tipoAcesso.administrdor,
AccessControl.TipoAcesso.Administrador. Com certeza o segundo é muito mais profissional!

Parametros

Use sempre Camel Case.

```
public DataTable EnviaEmail(string de, string para, bool preAutenticar){}
```

Métodos

Métodos devem ser escritos em Pascal Case. accessControl.EnviaEmail("igor@xxxx.xxx", _sendTo, false);

Propriedades

Pascal Case, pois assim, usariamos a propriedade exatamente assim: accessControl.Usuario

Campos

Campos para utilização apenas interna da classe devem ser escritos como Camel Case, e se for um campo que será utilizado por uma Propriedade, deve-se iniciar com um _ (Underline), como no exemplo seguinte:

```
private string _usuario;  
public string Usuario  
{  
    GET{return _usuario;}  
    SET{_usuario = value;}  
}
```

Galera, Espero que gostem deste artigo!

Para a próxima, vou tentar escrever algo sobre Boas Maneiras como Comentários...

Ou quem sabe algo sobre Métodos Ágeis, tipo, um pouco sobre as tecnologias como TDD, Continuous Integration...

Ou ainda algo como tecnologias para facilitar o trabalho em equipe..

Aguardem ansiosos, pois há muito a aprender!