

Curso C# Conceitos Básicos

Diogo Cezar Teixeira Batista

xgordo@gmail.com

Universidade Tecnológica Federal do Paraná
Campus Cornélio Procópio
UTFPR-CP

Cornélio Procópio - 2008

Agenda I

1 Aprofundando em Windows Forms

- Aplicações MDI
 - Adicionando uma nova janela filha
 - Fechando uma janela filha
- Aplicações SDI
 - Comandos Show() e ShowDialog()
- Passando valores entre Forms
- Posicionando e estado dos formulários na tela
- Controlando os eventos dos formulários
- Aplicações de exemplo
- Exercícios para praticar
- Exercício para entregar

Aprofundando em Windows Forms

- Programas dificilmente utilizam somente 1 form;
- Basicamente podemos ter os seguintes tipos de interface Windows:
 - MDI (*Multiple Document Interface*): Múltiplos documentos : o Word por exemplo;
 - SDI (*Single Document Interface*): Apenas um documento : a calculadora por exemplo;
 - Janelas modais. Exemplo: As janelas informativas conhecidas como diálogos.

1. *Journal of the American Medical Association*, 2000; 283: 2689-2696.

- Têm por objetivo criar várias instâncias;
- Instâncias filhas dentro de uma aplicação principal;
- Um projeto se torna MDI a seguinte configuração:
 - Defini-se o form que será a aplicação principal alterando-se a propriedade `isMdiContainer = true`

Adicionando uma nova janela filha I

Definição

Para adicionar uma nova janela filha, devemos inserir um novo formulário ao projeto: menu *File/Add* → *New Item* → selecione *Windows Forms*.

- Ao instanciarmos uma nova janela filha utilizamos a seguinte codificação:

Código 1: Janela filha (MDI)

```
1 wndFilha myWnd = new wndFilha( );  
2 myWnd.MdiParent = this;  
3 myWnd.Text = "Janela Filha";  
4 myWnd.Show();
```

Adicionando uma nova janela filha II

O que se fez aqui?

- Nova instância do formulário (filho);
- Definimos o pai da janela filha (*this*);
- Mostramos a janela filha (*show*);

Fechando uma janela filha

- Para fechar uma janela filha MDI ativa utiliza-se a seguinte notação:

Código 2: Fechar janela filha (MDI)

```
1 if (this.MdiChildren.Length != 0)
2 this.ActiveMdiChild.Close( );
```

- ① Se existir uma janela MDI, então...
- ② Feche a janela MDI que está ativada.

Aplicações SDI

- Projeto *windows forms*: padrão SDI.
- Para adicionar um formulário:
 - ① Na janela *Solution Explorer*, clique com o botão direito no nome do projeto, selecione *Add* e clique em *Windows Form*;
 - ② Digite um nome para o novo formulário e clique em *Add*.
- Para renomear um formulário: clique sobre o nome do *Form* no *Solution Explorer* e clique em *Rename*;
 - Se a uma mensagem de confirmação aparecer, clique em Sim. Isso vai atualizar todas as referencias ao *form* do projeto para o novo nome.
- Para se exibir um formulário, deve-se criar uma instância (objeto) do mesmo.

Comandos Show() e ShowDialog()

Ambos os comandos são utilizados para abrir um formulário, entretanto:

- *Show()*: É possível navegar entre os outros formulários que estão abertos;
- *ShowDialog()*: Impede que se navegue entre outros formulários abertos enquanto o mesmo não seja fechado.
- Exemplo de utilização dos comandos *Show()* e *ShowDialog()*:

Código 3: Código para exibir formulário

```
1 frmVermelho Vermelho = new frmVermelho();  
2 Vermelho.Show();
```

Código 4: Código para exibir formulário com show dialog

```
1 frmVerde Verde = new frmVerde();  
2 Verde.ShowDialog();
```

Passando valores entre Forms

- É possível alterar atributos de um formulário, estando em outro;
- Modificador de acesso *public*;
- Pode-se passar objetos entre formulários.

Ilustrando

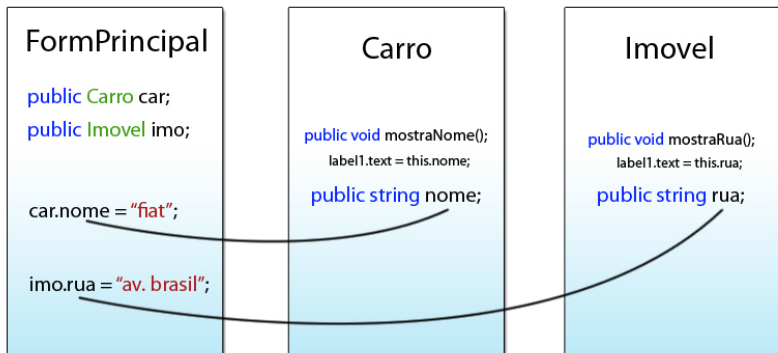


Figura: Ilustrando como é possível o compartilhamento de valores entre forms

Posicionando os formulários na tela *StartPosition*

- Estética do projeto;
- Propriedade *StartPosition*:
 - *CenterScreen*: centro da tela;
 - *Manual*: Deve-se especificar manualmente os valores:
 - Posicionamento x, y do formulário na tela.

Estado inicial do formulário

- *Normal*: Abre o formulário em seu estado inicial;
- *Minimized*: Abre o formulário minimizado na barra de tarefas;
- *Maximized*: Abre o formulário maximizado.

Controlando os eventos dos formulários

- Ações atribuídas ao comportamento do formulário;
- Ao ocorrer um evento, um bloco de código é executado (*Manipulador de Evento*);
- Nomeação padrão combina o nome do objeto com o evento correspondente ligando-os por um underline:
 - exemplo: `button1_Click`; `form1_Load`.

Aplicações de exemplo

Para fixar o conteúdo estudado, vamos analisar as seguintes aplicações:

- Aplicação com janelas MDI (MDIApplication);
- Aplicação com formulários Verde e Vermelho (VerdeVermelho);
- Aplicação com posicionamento das janelas (PosicaoForm);
- Aplicação com eventos dos formulários (EventosForm);
- Aplicação com interação de objetos entre formulários (InteracaoFormularios);

Exercícios para praticar

- ❶ Em uma aplicação MDI, implemente n filhos para uma aplicação, cada filho dessa aplicação será um bloco de notas que deverá abrir e salvar arquivos de locais selecionados pelo usuário. O programa deve mostrar o número do filho, e ordenar as janelas conforme a aplicação MDI de exemplo.
- ❷ Faça uma aplicação que altere através de um formulário principal a cor, o tamanho, o posicionamento e um label de um outro formulário no projeto.
- ❸ Utilizando o primeiro exercício e os conceitos de eventos, implemente:
 - Mensagem de boas vindas ao abrir o programa;
 - Confirmação quando o usuário tentar sair do programa;

Exercício para entregar I

Uma empresa solicitou a criação de um programa que vai registrar em um único arquivo texto seus clientes, clientes preferenciais e seus fornecedores. A estrutura para a criação do arquivo pode ser analisada no seguinte diagrama de classes:

Exercício para entregar II

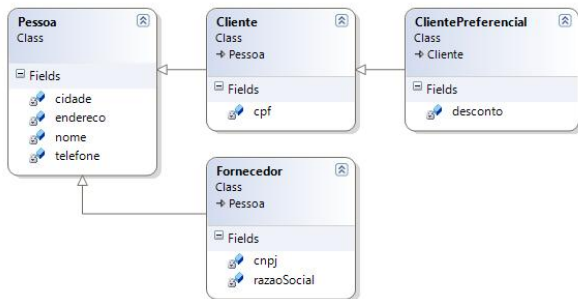


Figura: Diagrama de classes a ser implementado

Exercício para entregar III

Existe um projeto criado pelo analista de sistema disponível para facilitar a implementação do projeto, esse arquivo se chama ExWindowApp.rar (disponível no moodle); O arquivo de saída deve ser formatado da seguinte maneira:

Código 5: Arquivo texto utilizado pela empresa

```
1 R:F::
2 cnpf:000.000.000-00/00;
3 nome:bebidas gel dist;
4 razao_social:coca-cola s/a;
5 cidade:cornélio procópio;
6 endereco:rua pedro gomes, 425;
7 telefone:043-3523-3340;
8 --
9 cnpf:111.121.242-24/21;
10 nome:padaria pao bao;
11 razao_social:pao bao ltda;
```

Exercício para entregar IV

```
12  cidade:paraguacu;  
13  endereco:rua alfredo quincas , 49;  
14  telefone:043-8897-9889;  
15  --  
16  R:C::  
17  cpf:009.880.000-29;  
18  nome:pedro bozzeli;  
19  cidade:paraguacu;  
20  endereco:julio gomes , 8849;  
21  telefone:043-8878-7789;  
22  --  
23  R:CP::  
24  cpf:059.855.064-49;  
25  nome:pedro rico;  
26  cidade:londrina;  
27  endereco:jk , 1232;  
28  telefone:043-3345-9789;  
29  desconto:45.50;  
30  --
```

Exercício para entregar V

R:F:: indica que os próximos registros serão fornecedores, – indica a separação entre registros, R:C:: indica o registro de clientes e R:CP :: indica o registro de clientes preferenciais.

É importante que o programa seja feito em múltiplos formulários, um formulário para cadastro de fornecedores, outro para cadastro de clientes e outro para o cadastro de clientes preferenciais. O programa deverá trabalhar com listas de objetos. O programa deve ter ainda a opção: salvar dados, a qual deverá localizar um local para gerar o arquivo texto conforme a formatação solicitada.

Esse projeto engloba a maioria das funcionalidades estudadas até o momento e valerá 20 pontos na nota dos exercícios.

Data da entrega: 17 de outubro.

Dica: utilize os projetos já montados e reaproveite seus códigos.