
Delphi 7 - Aula 02 - Preço do Automóvel

Prof. Gilbert Azevedo da Silva

I. Objetivos

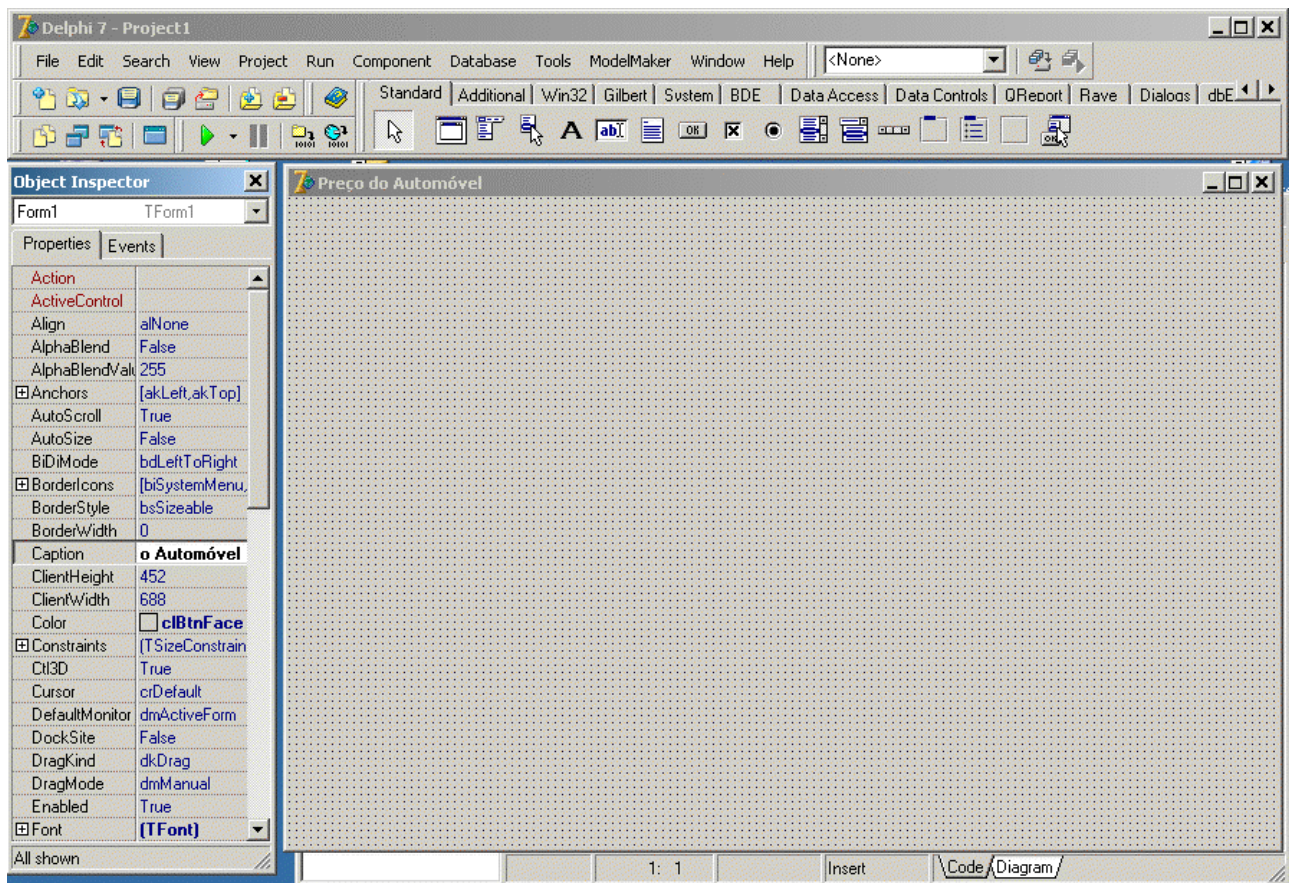
- Conhecer os tipos de propriedades dos objetos,
- Utilizar a ajuda do Delphi (help),
- Utilizar os componentes: GroupBox, ComboBox, ListBox, RadioGroup, RadioButton e CheckBox.

II. Teoria

1. Propriedades dos Objetos

É muito importante para o programador distinguir as etapas de desenvolvimento de um aplicativo. Quando o Delphi é aberto, o aplicativo está na fase de projeto. Nesta etapa, o programador constrói a interface gráfica (formulários) e escreve o código da aplicação (unidades), definindo os comandos que são executados quando o usuário interagir com o programa.

Na etapa de projeto, o programador pode alterar as propriedades dos objetos utilizando o inspetor de objetos (Object Inspector). A figura abaixo, por exemplo, mostra a edição da propriedade Caption da janela do aplicativo. (Este procedimento foi utilizado no exemplo da Aula 01).



Quando o comando Run (F9) do Delphi é selecionado, o aplicativo em desenvolvimento passa da etapa de projeto para a etapa de execução. Durante esta etapa não é possível alterar propriedades utilizando o inspetor de objetos. Na execução, quaisquer alterações nas propriedades dos objetos devem ser programadas utilizando o editor de códigos. A sintaxe utilizada para acessar uma propriedade é mostrada abaixo:

Objeto.Propriedade

O exemplo a seguir mostra a atribuição da string "Olá, Mundo!" à propriedade Text do objeto Edit1. Quando este comando é executado, o conteúdo da caixa de texto recebe a string dada. Neste caso, uma operação de escrita é realizada na propriedade, ou seja, a propriedade do objeto está recebendo um novo valor.

```
Edit1.Text := 'Clique Aqui';
```

O comando abaixo, por sua vez, mostra uma tela de mensagem com a string escrita na caixa de textos. Neste caso, uma operação de leitura é realizada na propriedade do objeto, retornando o seu valor atual.

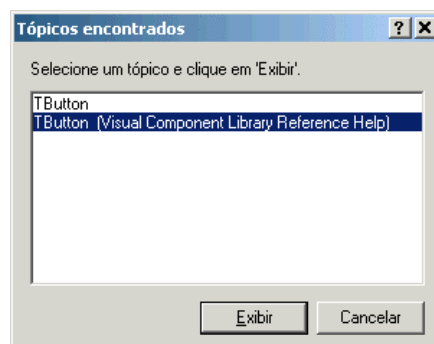
```
ShowMessage (Edit1.Text);
```

É importante perceber a diferença entre as operações de escrita e leituras das propriedades. Enquanto a primeira operação atribui um novo valor a uma determinada propriedade de um objeto; a segunda, apenas retorna o seu valor atual. A maioria das propriedades dos objetos é do tipo leitura e escrita, ou seja, permitem que ambas as operações sejam realizadas. Entretanto, algumas propriedades podem realizar apenas uma das operações sendo, portanto, dos tipos somente-leitura (read-only) ou somente-escrita (write-only).

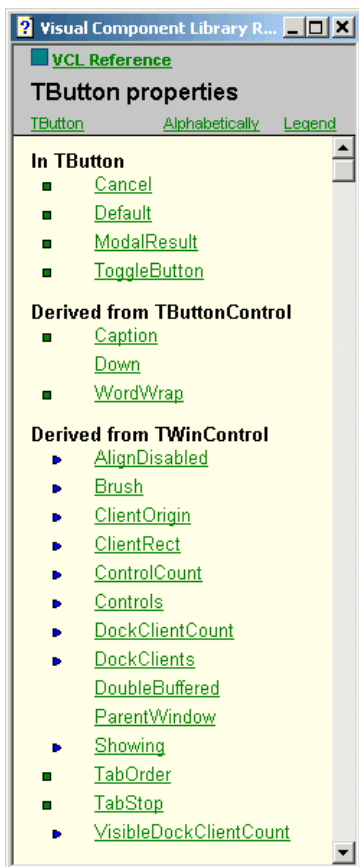
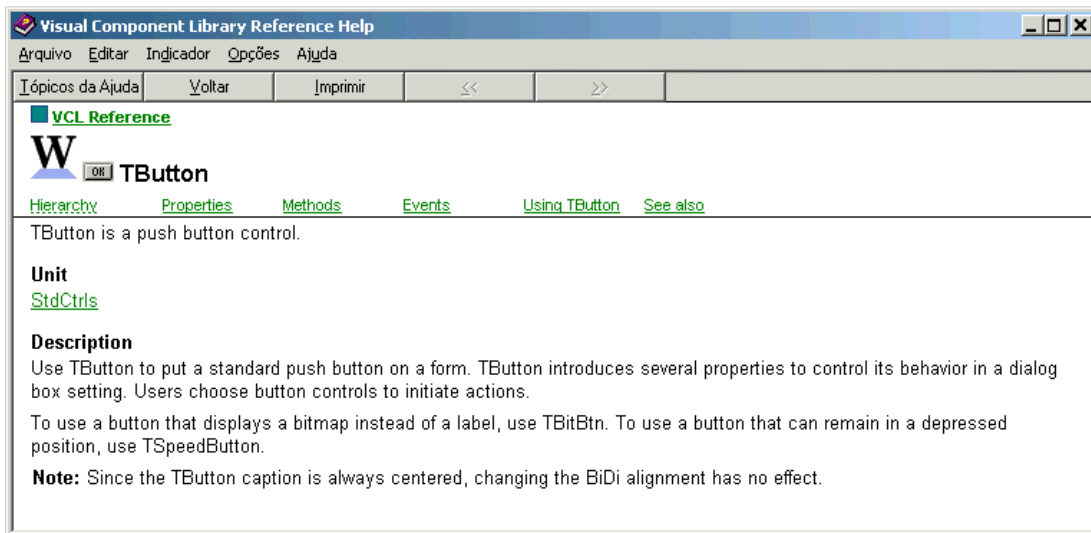
As propriedades são também classificadas quanto à visibilidade em publicadas, públicas e protegidas. Todas as propriedades apresentadas no inspetor de objetos são ditas publicadas (published), podendo ser manipuladas durante as etapas de projeto (através do inspetor de objetos) e de execução da aplicação (através de código). As propriedades públicas (public) não são apresentadas no inspetor de objetos e podem ser manipuladas apenas através de código. As protegidas são utilizadas apenas em casos mais avançados da programação Delphi quando novos tipos de objetos estão sendo criados. É importante notar que todas as propriedades publicadas permitem as operações de escrita e leitura.

2. Ajuda do Delphi (Help)

Para conhecer tudo a respeito de um objeto, incluindo suas propriedades, simplesmente clique no objeto desejado na paleta de componentes do Delphi e pressione F1. Uma janela semelhante à apresentada abaixo é aberta. Selecione sempre o item que mostrar algo parecido com Visual Component Library ou VCL ou Windows. Esta ajuda refere-se aos objetos utilizados na programação Windows; o outro item normalmente refere-se aos objetos da biblioteca CLX utilizados na programação Windows e Linux.



Clique em Exibir para mostrar a ajuda do objeto selecionado. Para este exemplo foi escolhido o objeto botão (TButton). A ajuda abre uma nova tela, mostrada a seguir, onde pode ser encontrado muito sobre o referido objeto. Os principais atalhos nesta tela são: Properties – mostra as propriedades do objeto, relacionadas às suas características, Methods – mostra os métodos do objeto, relacionados às ações que o objeto realiza e Events - mostra os eventos que o objeto suporta, relacionados às ações que um usuário pode realizar no objeto. A ajuda mostra ainda uma breve descrição da funcionalidade do objeto.



Clique em Properties para verificar as propriedades deste objeto. A janela de propriedades, ao lado, é exibida. As propriedades assinaladas por um quadrado verde são publicadas; as demais propriedades são disponíveis somente em tempo de execução, conforme citado anteriormente. As propriedades assinaladas com uma seta azul são somente leitura, ou seja, não permitem que você atribua de novos valores a ela.

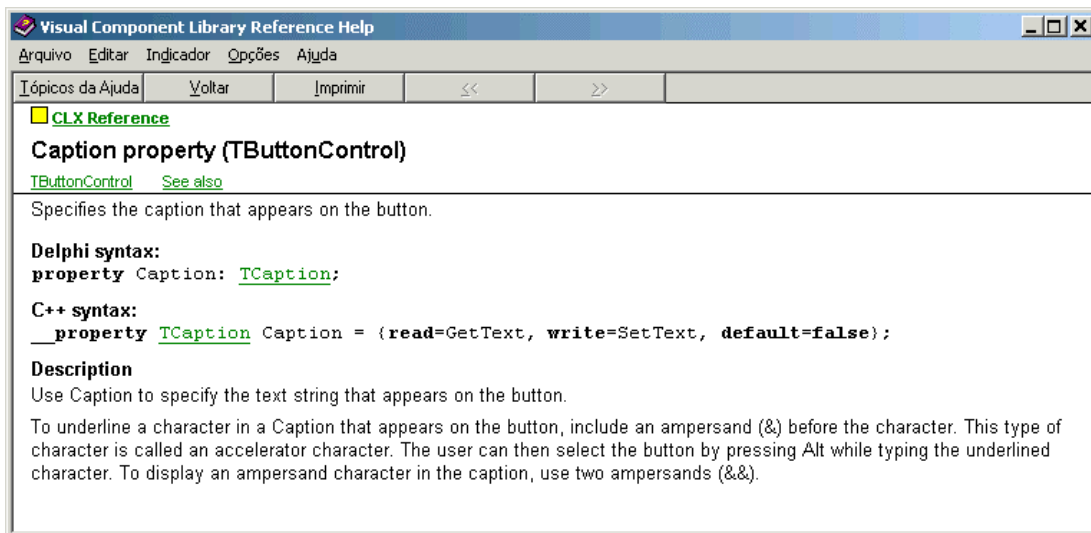
Os objetos possuem normalmente várias propriedades, sendo que a maioria delas é normalmente definida em classes ancestrais. Como visto ao lado, somente quatro das propriedades de um botão são específicas a ele (TButton), as demais ele herda de classes ancestrais como TButtonControl e TWinControl. Entretanto, todas as propriedades mostradas são propriedades do botão e podem ser usadas em sua configuração.

Classes são estruturas que definem as características dos objetos. Por exemplo, imagine as classes Ser Vivo, Animal, Mamífero e Homem. A classe Ser Vivo define as propriedades de um ser vivo, como, por exemplo, a capacidade de reproduzir-se. A classe Animal descende de Ser Vivo, ou seja, possui todas as suas características e adiciona suas próprias especificidades. De forma análoga, a classe Mamífero descende de Animal e a classe Homem descende de Mamífero. Você, por exemplo, pode ser visto como um objeto da classe Homem, possuindo todas as características definidas nesta classe e nas classes ancestrais a ela. Todos nós somos homens, mamíferos, animais e seres vivos. A ordem inversa pode não ser verdadeira. Um ser vivo pode não ser um homem, por exemplo.

Os objetos utilizados na programação em Delphi (e em qualquer linguagem de programação orientada ao objeto) obedecem a uma estrutura semelhante a esta. A definição de classes de objetos simplifica a especialização e a padronização dos mesmos. Como será visto adiante, muitos objetos possuem propriedades em comum e este fato é justificado pelo motivo de terem ancestrais também comuns.

Voltando à janela da ajuda. Ao clicar em uma propriedade, uma nova janela é apresentada (vista a seguir). Esta tela mostra qual a funcionalidade da propriedade selecionada, qual o tipo de informação que ela armazena, além de uma breve descrição sobre seu funcionamento. Tópicos e exemplos relacionados ao tema são comumente acessíveis através dos atalhos See Also e Examples apresentados na parte superior das janelas de ajuda.

Navegar nos atalhos referente aos botões, e aos demais objetos, certamente lhe levará a uma infinidade de informação que pode ser útil na realização de um determinado aplicativo. A ajuda é uma ferramenta bastante útil para o conhecimento das propriedades dos objetos e o aprendizado da programação em Delphi. Como um grande número de objetos é disponível - e um número ainda maior de propriedades está relacionado a eles - é impossível conhecer todas as suas características e funcionalidades, sendo portanto imprescindível recorrer à ajuda para se conhecer sempre um pouco mais desta ferramenta de programação.



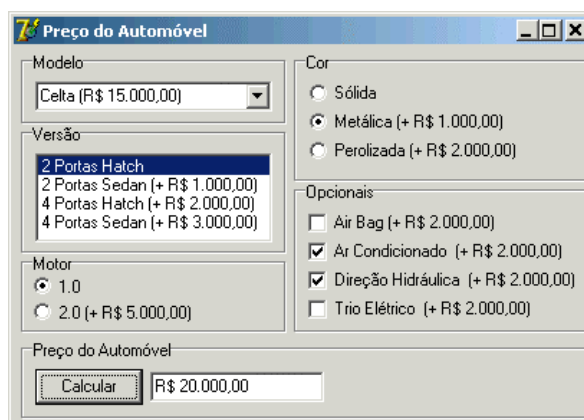
III. Exemplo

1. Objetivo

O exemplo desta aula mostra o funcionamento básico dos componentes GroupBox – que é um container para outros objetos; ComboBox, ListBox, RadioGroup – que são utilizados para listar opções; RadioButton – que é um botão de seleção e CheckBox – que é uma caixa de verificação.

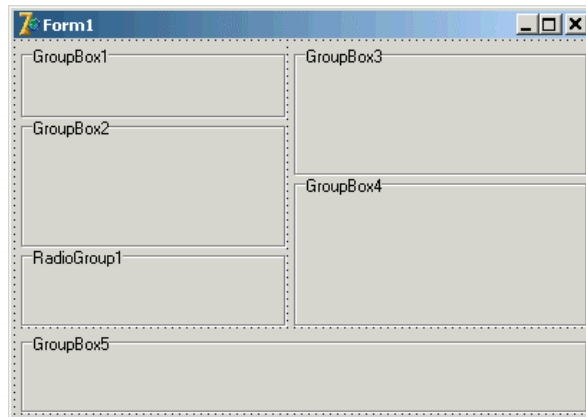
O objetivo da aplicação é calcular o preço final de um automóvel baseado na seleção do modelo, versão, motor, cor e opcionais do veículo. O preço final do automóvel deve ser calculado quando o botão Calcular for clicado. Os valores e opções apresentados no exemplo são meramente didáticos.

A janela do aplicativo é apresentada abaixo. A seleção do modelo é realizada através de um objeto ComboBox que mostra a lista de veículos disponíveis, informando o seu preço básico (versão 2 portas Hatch, motor 1.0, cor sólida e sem opcionais). As seleções da versão e do motor são feitas através dos componentes ListBox e RadioGroup, respectivamente. A seleção da cor é realizada através de objetos RadioButton e os opcionais, através de CheckBox. O preço final do automóvel é dado pelo preço básico acrescido dos valores apresentados para cada seleção realizada (versão, motor, cor e opcionais).



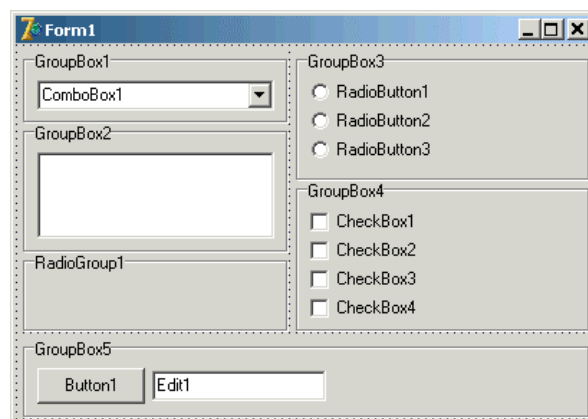
2. Inserindo Objetos no Formulário

Insira no formulário 5 objetos GroupBox e um RadioGroup conforme mostrado na janela a seguir. Organize os objetos utilizando o mouse ou as propriedades Left e Top para a posição do objeto no Form, Height e Width para a altura e largura do objeto, respectivamente.



Os objetos GroupBox são containers para objetos, permitindo que outros objetos sejam incluídos no seu interior com o objetivo de compor a interface gráfica dos aplicativos. Para concluir a etapa de inclusão de objetos, insira os componentes listados na tabela abaixo dentro do GroupBox especificado. O formulário da aplicação deve ficar conforme mostrado na figura.

GroupBox	Objeto
GroupBox1	ComboBox
GroupBox2	ListBox
GroupBox3	RadioButton1 RadioButton2 RadioButton3
GroupBox4	CheckBox1 CheckBox2 CheckBox3 CheckBox4
GroupBox5	Button1 Edit1



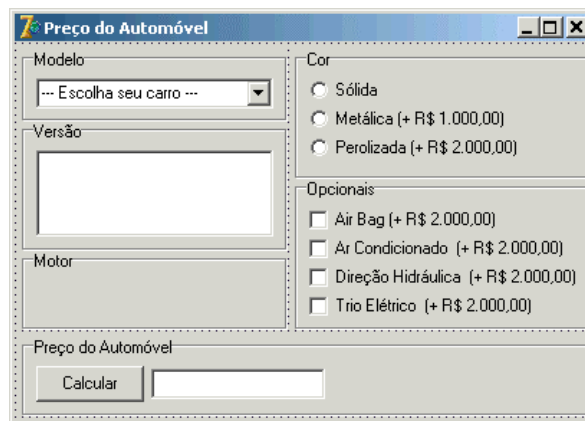
3. Alterando Propriedades dos Objetos

Altere as propriedades Caption e Text dos objetos de acordo com a tabela abaixo

Objeto	Propriedade	Valor
Button1	Caption	Calcular
Form1	Caption	Preço do Automóvel
GroupBox1	Caption	Modelo
GroupBox2	Caption	Versão
GroupBox3	Caption	Cor
GroupBox4	Caption	Opcionais
GroupBox5	Caption	Preço do Automóvel
RadioGroup1	Caption	Motor
ComboBox1	Text	--- Escolha seu carro ---
Edit	Text	“vazio”

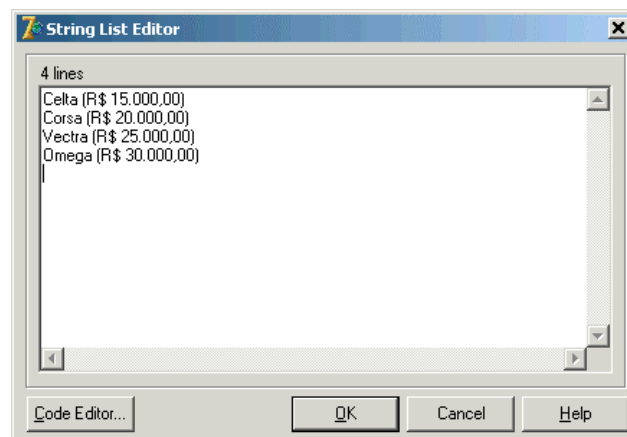
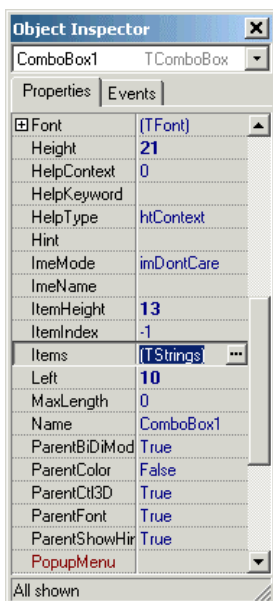
Altere a propriedades Caption dos RadioButtons e dos CheckBoxes de acordo com a tabela abaixo. O formulário da aplicação deve ficar conforme mostrado na figura.

Objeto	Propriedade	Valor
RadioButton1	Caption	Sólida
RadioButton2	Caption	Metálica (+ R\$ 1.000,00)
RadioButton3	Caption	Perolizada (+ R\$ 2.000,00)
CheckBox1	Caption	Air Bag (+ R\$ 2.000,00)
CheckBox2	Caption	Ar Condicionado (+ R\$ 2.000,00)
CheckBox3	Caption	Direção Hidráulica (+ R\$ 2.000,00)
CheckBox4	Caption	Trio Elétrico (+ R\$ 2.000,00)

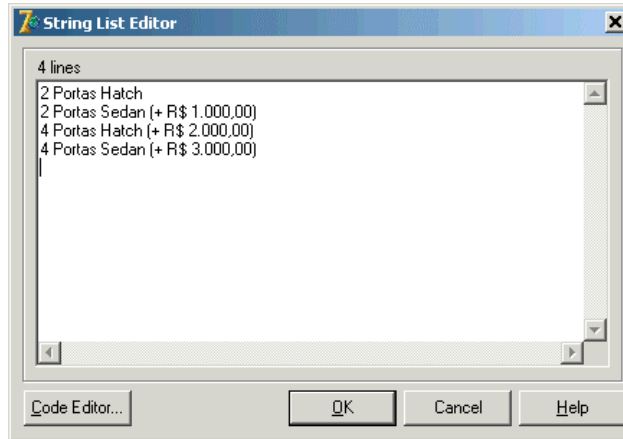


Os objetos ComboBox, ListBox e RadioGroup possuem funcionalidade semelhante. Esses objetos mantêm uma lista de itens selecionáveis dada pela propriedade Items. A principal diferença entre eles consiste na forma gráfica de apresentação desta lista. Outra propriedade comum aos três é ItemIndex que indica qual item da lista está selecionado. Esta propriedade tem valor -1 quando nenhum item é selecionado, 0 quando o primeiro item é selecionado, 1 para o segundo item, 2 para o terceiro e assim por diante.

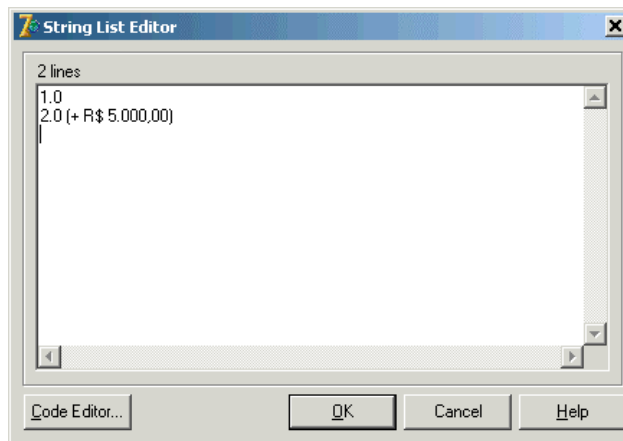
Selecione a propriedade Items do ComboBox1 (clique no botão com reticências ao lado da propriedade Items no inspetor de objetos) e preencha a lista de acordo com a figura abaixo. Pressione OK para finalizar a edição da lista.



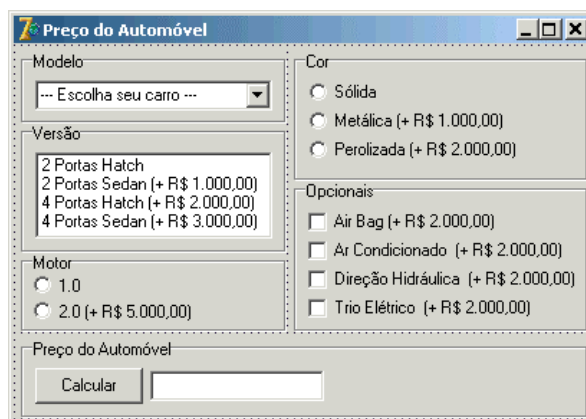
Selecione a propriedade Items do ListBox1 e preencha a lista de acordo com a figura abaixo. Pressione OK para finalizar a edição da lista.



Selecione a propriedade Items do RadioGroup1 e preencha a lista de acordo com a figura abaixo. Pressione OK para finalizar a edição da lista.



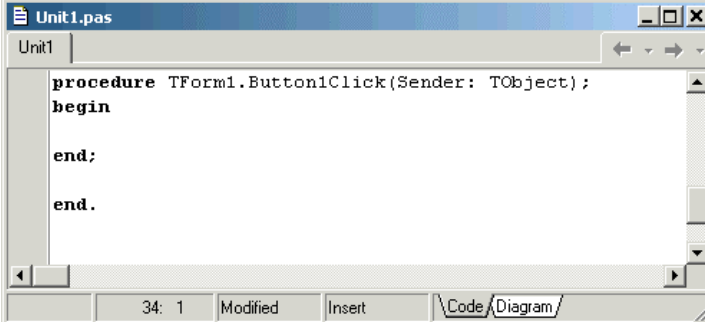
O formulário da aplicação está completo e deve parecer conforme mostrado na figura abaixo.



No ComboBox, o usuário deve selecionar o modelo do automóvel. Esta seleção implica no preço inicial do veículo. Através dos ListBox, RadioGroup e RadioButtons, o usuário seleciona a versão, o motor e o tipo de cor do veículo. De acordo com a seleção, o preço do automóvel pode ou não ter acréscimos. Até aqui, apenas uma opção em cada lista pode ser selecionada. Na lista de opcionais, entretanto, o usuário pode optar por nenhum, um ou vários opcionais, isto por que a seleção utilizando CheckBox não é mutuamente exclusiva.

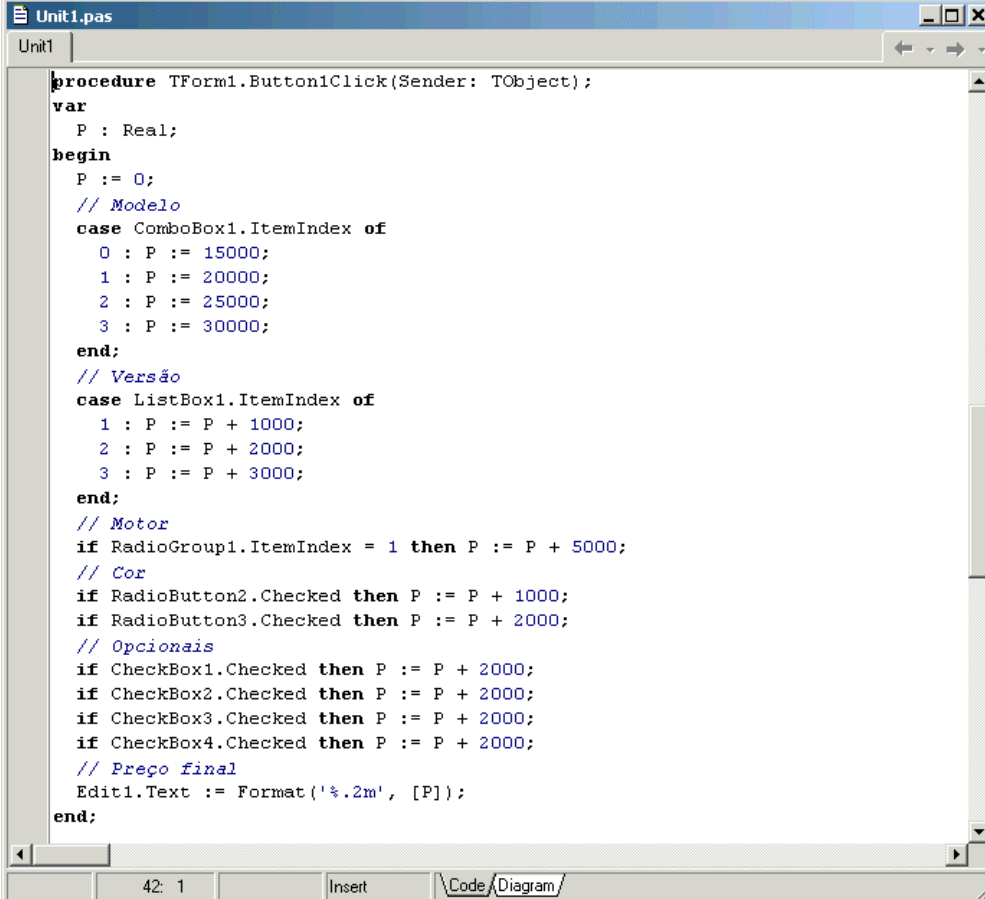
4. Programando Eventos

O passo final para construção do aplicativo consiste na programação do evento de clique no botão. Para programar este evento, dê dois cliques no botão Calcular. O Delphi passará automaticamente para a janela de edição de códigos, conforme mostrado abaixo. O procedimento `Button1Click` é incluído na unidade de códigos do aplicativo. Os comandos escritos dentro dele serão executados quando o botão Calcular for clicado pelo usuário. Neste exemplo, vamos considerar que o usuário selecionou algum dos modelos de automóvel disponível.



```
Unit1.pas
Unit1
procedure TForm1.Button1Click(Sender: TObject);
begin
end;
end.
```

O código para calcular o preço final do automóvel é mostrado abaixo. Partindo do preço básico do automóvel, a seleção realizada em cada objeto altera o seu valor.



```
Unit1.pas
Unit1
procedure TForm1.Button1Click(Sender: TObject);
var
  P : Real;
begin
  P := 0;
  // Modelo
  case ComboBox1.ItemIndex of
    0 : P := 15000;
    1 : P := 20000;
    2 : P := 25000;
    3 : P := 30000;
  end;
  // Versão
  case ListBox1.ItemIndex of
    1 : P := P + 1000;
    2 : P := P + 2000;
    3 : P := P + 3000;
  end;
  // Motor
  if RadioGroup1.ItemIndex = 1 then P := P + 5000;
  // Cor
  if RadioButton2.Checked then P := P + 1000;
  if RadioButton3.Checked then P := P + 2000;
  // Opcionais
  if CheckBox1.Checked then P := P + 2000;
  if CheckBox2.Checked then P := P + 2000;
  if CheckBox3.Checked then P := P + 2000;
  if CheckBox4.Checked then P := P + 2000;
  // Preço final
  Edit1.Text := Format('%.2m', [P]);
end;
```

Uma variável local – `P` – é usada para a realizar do cálculo do preço do automóvel. Para os objetos `ComboBox`, `ListBox` e `RadioGroup`, a seleção realizada pelo usuário é indicada pela propriedade `ItemIndex`. Nos objetos `RadioButton` e `CheckBox`, a seleção é indicada pela propriedade `Checked`.

O valor inicial de `P` é dado de acordo com a seleção do `ComboBox1` que indica qual o modelo selecionado. Quando a propriedade `ItemIndex` do `ComboBox1` é zero, o modelo selecionado é Celta e `P` recebe o valor 15000. Se `ItemIndex` é um, o modelo selecionado é Corsa e `P` recebe 20000. Para `ItemIndex` igual a dois, `P`

recebe 25000 e para ItemIndex igual a três, 30000. Os possíveis valores para ItemIndex são testados com um comando CASE.

A seleção da versão é realizada no ListBox. Neste caso, a propriedade ItemIndex indica qual o acréscimo realizado no valor da variável P. Os possíveis valores para ItemIndex são testados também com um comando CASE.

A seleção do motor é feita no RadioGroup. Neste caso, o valor da propriedade ItemIndex é testado com um comando IF, isto por que a verificação de um único valor é relevante para a aplicação.

Para as opções de cor e opcionais, são verificadas as propriedades Checked dos RadioButtons e CheckBoxes. Quando esses objetos estão selecionados, a propriedade Checked retorna o valor True. Os comandos IF verificam qual a cor selecionada e quais os opcionais incluídos no automóvel, fazendo o devido incremento no seu preço.

Finalmente, o preço final do veículo é convertido de Real para String e apresentado na caixa de textos Edit1. A conversão é realizada com a função Format que possibilita transformar um valor numérico para sua representação na forma monetária.

5. Executando a Aplicação

Execute e teste a aplicação, selecionado o comando Run do Delphi (F9).

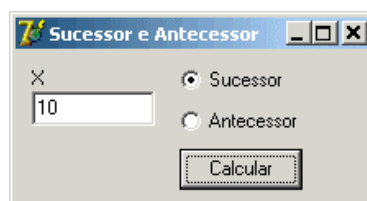
6. Salvando a Aplicação

Salve a aplicação, clicando no botão Save All da barra de ferramentas.

IV. Exercícios

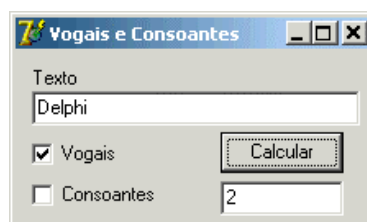
1. Sucessor e Antecessor

Implementar um aplicativo para calcular o sucessor ou antecessor de um número (X). O resultado deve ser mostrado na própria caixa de texto onde o valor inicial é dado.



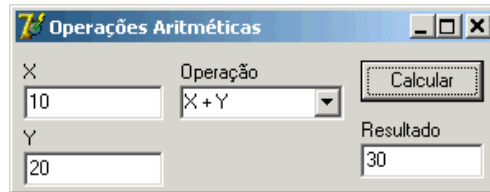
2. Vogais e Consoantes

Implementar um aplicativo para contar o número de vogais e/ou consoantes em um texto.



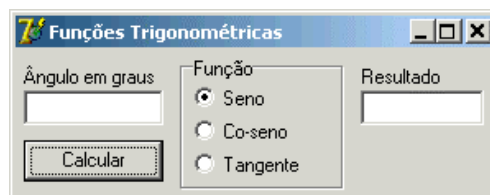
3. Operações Aritméticas

Implementar um aplicativo para calcular a adição ($X+Y$), subtração ($X-Y$), multiplicação ($X*Y$) ou divisão (X/Y) dos valores X e Y dados, de acordo com a seleção do ComboBox, quando o botão Calcular é clicado.



4. Funções Trigonômétricas

Implementar um aplicativo para calcular seno, co-seno e tangente de um ângulo dado em graus. Utilizar um RadioGroup.



5. Eleições

Implementar um aplicativo para contar os votos em uma eleição. A seleção do candidato deve ser realizada em um ListBox. A cada candidato selecionado, mostrar uma figura diferente.

