

Módulo

3

Object Pascal I

INTRODUÇÃO

Essa lição tem por objetivo apresentar a linguagem que o Delphi utiliza: Object Pascal. Tenha em mente que o Delphi não é uma linguagem, mas sim uma Ferramenta de Desenvolvimento que utiliza a linguagem Object Pascal.

CONCEITOS DE PROGRAMAÇÃO ORIENTADA A OBJETOS

Antes de partir para a linguagem propriamente dita, vamos aprender alguns conceitos básicos de Programação Orientada a Objetos.

Classe: Definição de tipo dos objetos, modelo de objeto.

Objeto: Instância de classe, variável cujo tipo é uma classe.

Atributos: Variáveis de instância. São os dados de um objeto.

Métodos: Funções e procedimentos de um objeto.

Propriedades: Apelido usado para evitar o acesso direto aos atributos de um objeto, onde podemos especificar métodos que serão usados para ler e atribuir seus valores a esses atributos.

Mensagens: Chamada de métodos, leitura e atribuição de propriedades.

Encapsulamento: Conjunto de técnicas usadas para limitar o acesso aos atributos e métodos internos de um objeto.

Herança: Possibilidade de criar uma classe descendente de outra, aproveitando seus métodos, atributos e propriedades.

Ancestral: Super classe ou classe de base, a partir da qual outras classes podem ser criadas.

Descendente: Subclasse.

Hierarquia de Classes: Conjunto de classes ancestrais e descendentes, geralmente representadas em uma árvore hierárquica.

Polimorfismo: Capacidade de redefinir métodos e propriedades de uma classe em seus descendentes.

ESTRUTURA DE UNITS

Vamos examinar o código gerado para um novo Form, identificando as principais seções de uma Unit típica. Abra o Delphi e crie uma nova aplicação. Observe na Unit principal as seguintes cláusulas.

Unit: A primeira declaração de uma unit é seu identificador, que é igual ao nome do arquivo.

Interface: Seção interface, onde ficam declarações que podem ser usadas por outras Units.

Uses: Na cláusula uses fica a lista de Units usadas.

Type: Na cláusula type fica a definição de tipos, aqui temos a declaração da classe do Form.

Var: Na cláusula var são declaradas as variáveis, aqui temos a declaração da instância do Form.

Implementation: Na seção implementation ficam as definições dos métodos.

End: Toda Unit termina com um end a partir de onde qualquer texto é ignorado.

VARIÁVEIS

No Delphi, toda variável tem que ser declarada antes de ser utilizada. As declarações podem ser feitas após a palavra reservada `var`, onde são indicados o nome e o tipo da variável. Os nomes de variáveis não podem ter acentos, espaços ou caracteres especiais como `&`, `$` ou `%` e o primeiro caractere de um nome de variável tem que ser uma letra ou um sublinhado (`_`).

VARIÁVEIS GLOBAIS

As variáveis abaixo são globais, declaradas na Interface da Unit. Podem ser acessadas por qualquer Unit usuária.

```
var
  I: Integer;
  Usuario: string;
```

```
A, B, Soma: Double;  
Ok: Boolean;
```

VARIÁVEIS LOCAIS

As variáveis abaixo são locais ao método, ou seja, elas só existem dentro do método, não podem ser acessadas de fora, mesmo que seja na mesma Unit. Na verdade essas variáveis são criadas quando o método é chamado e são destruídas quando ele é encerrado, seu valor não é persistente.

```
procedure TFrmExemplo.BtnTrocarClick(Sender: TObject);  
var  
    Aux: string;  
begin  
    Aux := EdtA.Text;  
    EdtA.Text := EdtB.Text;  
    EdtB.Text := Aux;  
end;
```

ATRIBUTOS

Os atributos são variáveis de instância. Para declarar um atributo em uma classe basta definir o identificador e o tipo do atributo na declaração da classe, feita na seção type da Interface da Unit, como abaixo.

```
type  
    TFrmSomar = class(TForm)  
    private {Private declarations}  
        A, B: Double;  
    public {Public declarations}  
        Soma: Double;  
    end;
```

ENCAPSULAMENTO

Os principais níveis de visibilidade dos atributos e métodos de uma classe são mostrados abaixo.

Nível	Visibilidade
Private	Os itens declarados nesse nível só podem ser acessados na mesma unit.
Public	Nesse nível, qualquer unit usuária poderá acessar o item.
Protected	Os itens só poderão ser acessados em outra unit se for em uma classe descendente.
Published	É o nível default, igual ao Public, mas define propriedades e eventos usados em tempo de projeto.

CLASSES

Classes são tipos de objetos, uma classe é declarada na cláusula type da seção interface e os métodos são definidos na seção implementation. Examine o código de um Form para identificar os elementos de sua classe.

```
interface  
type  
    TFrmSomar = class(TForm)  
        EdtA: TEdit;  
        EdtB: TEdit;  
        BtnSoma: TButton;  
        procedure BtnSomaClick(Sender: TObject);  
    private  
        { Private declarations }  
    public  
        { Public declarations }  
    end;  
  
implementation
```

```
procedure TFrmSoma.BtnSomaClick(Sender: TObject);
begin
    ShowMessage(EdtA.Text + EditB.Text);
end;
```

OBJETOS

Um Objeto é tratado como uma variável cujo tipo é uma classe. A declaração de objetos é igual à declaração de uma variável simples, tendo no lugar do tipo a classe do objeto.

```
var
    FrmSomar: TFrmSomar;
```

LITERAIS

Valores literais são valores usados em atribuições e expressões. Cada tipo tem uma sintaxe diferente.

Tipo	Definição
Inteiro	Seqüência de dígitos decimais (0 a 9), sinalizados ou não.
Inteiro Hexadecimal	Seqüência de dígitos hexadecimais (0 a F), precedidos por um cifrão (\$).
Real	Igual ao tipo Inteiro, mas pode usar separador decimal e notação científica.
Caractere	Letra entre apóstrofes ou o caracter # seguido de um número inteiro entre 0 e 255 (ASCII).
String	Seqüência de caracteres delimitados por apóstrofes.

CONSTANTES

São declaradas na seção const, podem ser usadas como variáveis, mas não podem ser alteradas. Geralmente o nome das constantes é escrito em letras maiúsculas e na declaração dessas constantes não é indicado o tipo.

```
const
    G = 3.94851265E-19;
    NUM_CHARS = '0123456789';
    CR = #13;
    SPACE = ' ';
    MAX_VALUE = $FFFFFFFF;
```

CONSTANTES TIPADAS

Na verdade, constantes tipadas são variáveis inicializadas com valor persistente, que podem ser alteradas normalmente, como qualquer variável. A única diferença de sintaxe entre constantes tipadas e simples é que o tipo da constante é indicado explicitamente na declaração. Se uma constante tipada for declarada localmente, ela não será destruída quando o método for encerrado. Para diferenciar das constantes normais, costuma-se declarar estas com letras de caso variável, como abaixo.

```
const
    Cont: Integer = 1;
    Peso: Double = 50.5;
    Empresa: string = 'Linux Informática';
```

INSTRUÇÕES

Os programas são compostos por instruções, que são linhas de código executável. Exemplos de instruções simples são atribuições, mensagens entre objetos, chamadas de procedimentos, funções e métodos, como mostradas abaixo. As instruções podem ser divididas em várias linhas, o que indica o fim de uma instrução é o ponto e vírgula no final. Quando uma instrução é quebrada, costuma-se dar dois espaços antes das próximas linhas, para melhorar a leitura do código.

```
Caption := 'Linux Informática!';
Form2.ShowModal;
Application.MessageBox('Você executou uma operação ilegal, o programa será finalizado.', 'Falha geral', MB_ICONERROR);
```

Você pode usar várias instruções agrupadas em uma instrução composta, como se fosse uma só instrução. Uma instrução composta delimitada pelas palavras reservadas `begin` e `end`. Toda instrução, simples ou composta, é terminada com um ponto-e-vírgula.

```
if CheckBox1.Checked then
begin
  ShowMessage('O CheckBox será desmarcado.');
```

CheckBox1.Checked := False;

```
end;
```

ESTILO DE CODIFICAÇÃO

As instruções e todo o código de uma Unit devem ser distribuídos para facilitar o máximo a leitura. Para isso, podemos usar a indentação, geralmente de três espaços para indicar os níveis de código. Procure criar um estilo próprio, que melhor se molde à sua realidade. Se for desenvolver em grupo, é melhor que todos usem o mesmo estilo para evitar confusões.

COMENTÁRIOS

Existem 3 estilos de comentário no Delphi, como mostrado abaixo:

```
(* Comentário do Pascal Padrão *)
{ Comentário do Turbo Pascal }
// Comentário de linha do C++
```

Cuidado com as diretivas de compilação, pois elas são delimitadas por chaves e podem ser confundidas com comentários. A diretiva de compilação mostrada abaixo é incluída em todas as Units de Forms.

```
{ $R*.DFM }
```

TIPOS DE DADOS PADRÃO

O Delphi trata vários tipos de dados padrão, segue uma descrição sucinta desses tipos.

TIPOS INTEIROS

São tipos numéricos exatos, sem casas decimais. O tipo `Integer` é o tipo inteiro padrão.

Tipo	Tamanho em Bytes	Valor Mínimo	Valor Máximo
ShortInt	1	-128	127
SmallInt	2	-32768	32767
LongInt	4	-2147483648	2147483647
Byte	1	0	255
Word	2	0	65535
Integer	4	-2147483648	2147483647
Cardinal	4	0	2147483647

TIPOS REAIS

São tipos numéricos com casas decimais. O tipo `Double` é o tipo real padrão.

Tipo	Tamanho em Bytes	Valor Mínimo	Valor Máximo	Dígitos Significativos
Real	6	10-39	1038	11-12
Single	4	10-45	1038	7-8
Double	8	10-324	10308	15-16
Extended	10	10-4932	104932	19-20
Comp	8	-1018	1018	19-20
Currency	8	-1012	1012	19-20

TIPOS TEXTO

Os tipos texto podem operar com caracteres simples ou grupos de caracteres. O tipo texto padrão é o tipo `String`.

Tipo	Descrição
Char	Um único caractere ASCII.
String	Texto alocado dinamicamente. Pode ser limitado a 255 caracteres conforme configuração.
PChar	String terminada em nulo (#0), usada geralmente nas funções da API do Windows.

O operador + pode ser usado para concatenar strings e você pode usar uma variável do tipo string como uma lista de caracteres.

```
ShowMessage('5ª letra do título da janela: ' + Caption[5]);
Label1.Text := '2ª letra do Edit: ' + Edit1.Text[2];
```

Existem várias funções de manipulação de strings, veja algumas das mais importantes mostradas abaixo.

Função	Descrição
AnsiCompareText	Compara 2 strings sem sensibilidade de maiúsculas/minúsculas.
AnsiLowerCase	Converte todas as letras de uma string para minúsculas.
AnsiUpperCase	Converte todas as letras de uma string para maiúsculas.
Copy	Retorna parte de uma string.
Delete	Apaga parte de uma string.
Insert	Insere uma string em outra.
Length	Número de caracteres de uma string.
Pos	Posição de uma string em outra.
Trim	Remove todos os espaços de uma string.
TrimLeft	Remove os espaços à esquerda de uma string.
TrimRight	Remove os espaços à direita de uma string.
Format	Formata uma string com uma série de argumentos de vários tipos.

Por exemplo, para comparar o texto de dois Edits, poderíamos usar a função AnsiCompareText.

```
if AnsiCompareText(EdtA.Text, EdtB.Text) = 0 then
  ShowMessage('O texto dos dois Edits é igual.');
```

A função Format é especialmente útil na formatação de strings, veja alguns exemplos.

```
ShowMessage(Format('O número %d é a parte inteira do número %f.', [10, 10.5]));
ShowMessage(Format('Este texto%sfui formatado%susando o caractere %d.', [#13, #13, 13]));
ShowMessage(Format('O preço do livro %s é %m.', ['Como Programar em Delphi', 50.7]));
```

Um detalhe que deve ser observado é que as propriedades dos objetos não podem ser usadas como variáveis em funções. Veja a declaração do procedimento Delete no help.

```
procedure Delete(var S: string; Index, Count: Integer);
```

Digamos que você deseje apagar as 5 primeiras letras de um Edit, como a string do Delete é variável, não poderia usar o código abaixo.

```
Delete(Edit1.Text, 1, 5);
```

Para você poder fazer a operação desejada, teria que usar uma variável como variável auxiliar.

```
var
  S: string;
begin
  S := Edit1.Text;
  Delete(S, 1, 5);
  Edit1.Text := S;
end;
```

TIPOS ORDINAIS

Tipos ordinais são tipos que tem uma seqüência incremental, ou seja, você sempre pode dizer qual o próximo valor ou qual o valor anterior a um determinado valor desses tipos. São tipos ordinais o Char, os tipos inteiros, o Boolean e os tipos enumerados. Algumas rotinas para ordinais são mostradas abaixo.

Função	Descrição
Dec	Decrementa variável ordinal.
Inc	Incrementa variável ordinal.
Odd	Testa se um ordinal é ímpar.
Pred	Predecessor do ordinal.
Succ	Sucessor do ordinal.
Ord	Ordem de um valor na faixa de valores de um tipo ordinal.
Low	Valor mais baixo na faixa de valores.
High	Valor mais alto na faixa de valores.

Por exemplo, use o código abaixo no evento OnKeyPress de um Edit e veja o resultado.

```
Inc(Key);
```

BOOLEAN

Variáveis do tipo Boolean podem receber os valores lógicos True ou False, verdadeiro ou falso. Uma variável Boolean ocupa 1 byte de memória.

TDATETIME

O tipo TDateTime guarda data e hora em uma estrutura interna igual ao tipo Double, onde a parte inteira é o número de dias desde 31/12/1899 e a parte decimal guarda a hora, minuto, segundo e milissegundo. As datas podem ser somadas ou subtraídas normalmente.

Existem várias rotinas de manipulação de datas e horas, usadas com o tipo TDateTime, veja algumas abaixo.

Rotina	Descrição
Date	Retorna a data do sistema.
Now	Retorna a data e hora do sistema.
Time	Retorna a hora do sistema.
DayOfWeek	Retorna o dia da semana de uma data especificada.
DecodeDate	Decodifica um valor TDateTime em Words de dia, mês e ano.
DecodeTime	Decodifica um valor TDateTime em Words de hora, minuto, segundo e milissegundos.
EncodeDate	Retorna um TDateTime a partir de Words de dia, mês e ano.
EncodeTime	Retorna um TDateTime a partir de Words de hora, minuto, segundo e milissegundos.

No help de cada uma das funções acima você vai encontrar alguns exemplos, veja os colocados abaixo.

```
if DayOfWeek(Date) = 1 then
  ShowMessage('Hoje é Domingo, pé de cachimbo!')
else
  ShowMessage('Hoje não é Domingo, pé de cachimbo!');

var
  A, M, D: Word;
begin
  DecodeDate(Date, A, M, D);
  ShowMessage(Format('Dia %.2d do mês %.2d de %d.', [D, M, A]));
end;
```

VARIANT

Tipo genérico, que pode atribuir e receber valores de qualquer outro tipo. Evite usar variáveis do tipo Variant, pois o uso dessas variáveis pode prejudicar a performance do programa, além de diminuir a legibilidade do código fonte e a integridade do executável, veja o trecho de código abaixo e note como esse tipo de variável tem um comportamento estranho.

```

var
    V1, V2, V3: Variant;
begin
    V1 := True;
    V2 := 1234.5678;
    V3 := Date;
    ShowMessage(V1 + V2 + V3);
end;

```

CONVERSÕES DE TIPO

Freqüentemente você vai precisar converter um tipo de dado em outro, como um número em uma string. Para essas conversões você pode usar duas técnicas, o TypeCasting e as rotinas de conversão de tipos.

TYPECASTING

TypeCast é uma conversão direta de tipo, usando o identificador do tipo destino como se fosse uma função. Como o Delphi não faz nenhuma verificação se a conversão é válida, você deve tomar um certo cuidado ao usar um TypeCast para não criar programas instáveis.

```

var
    I: Integer;
    C: Char;
    B: Boolean;
begin
    I := Integer('A');
    C := Char(48);
    B := Boolean(0);
    Application.MessageBox(PChar('Linguagem de Programação' + #13 + 'Delphi 3'),
    'Linux Informática', MB_ICONEXCLAMATION);
end;

```

ROTINAS DE CONVERSÃO

As principais rotinas de conversão estão listadas na tabela abaixo. Caso você tente usar uma dessas rotinas em uma conversão inválida, pode ser gerada uma exceção.

Rotina	Descrição
Chr	Byte em Char.
StrToInt	String em Integer.
IntToStr	Integer em String.
StrToIntDef	String em Integer, com um valor default caso haja erro.
IntToHex	Número em String Hexadecimal.
Round	Arredonda um número real em um Integer.
Trunc	Trunca um número real em um Integer.
StrToFloat	String em Real.
FloatToStr	Real em string.
FormatFloat	Número real em string usando uma string de formato.
DateToStr	TDateTime em string de data, de acordo com as opções do Panel de Controle.
StrToDate	String de data em TDateTime.
TimeToStr	TDateTime em String de Hora.
StrToTime	String de hora em TDateTime.
DateTimeToStr	TDateTime em string de data e hora.
StrToDateTime	String de data e hora em TDateTime.
FormatDateTime	TDateTime em string usando uma string de formato.
VarCast	Qualquer tipo em outro usando argumentos do tipo Variant.
VarAsType	Variante em qualquer tipo.
Val	String em número, real ou inteiro.
Str	Número, real ou inteiro, em String.

Veja alguns exemplos de como usar essas rotinas. Conversão de dados é uma operação muito comum na programação em Object Pascal, seria interessante dar uma olhada no help de cada uma das funções acima.

```
var
  I: Integer;
  D: Double;
  S1, S2: string;
begin
  D := 10.5;
  I := Trunc(D);
  S1 := FloatToStr(D);
  S2 := IntToStr(I);
  ShowMessage(S1 + #13 + S2);
end;

var
  A, B, Soma: Double;
begin
  A := StrToFloat(EditA.Text);
  B := StrToFloat(EditB.Text);
  Soma := A + B;
  ShowMessage(Format('%f + %f = %f', [A, B, Soma]));
end;
```

EXPRESSÕES

Uma expressão é qualquer combinação de operadores, variáveis, constantes, valores literais e chamadas de funções que resultem em um valor de determinado tipo. Uma expressão é usada sempre que precisamos de um valor que possa ser obtido por uma expressão.

```
A + 12 * C
Date - 4
StrToInt(Edit1.Text + Edit2.Text)
StrToDate(Edit2.Text) - StrToDate(Edit1.Text)
12 * A / 100
A < B
```

OPERADORES

Os operadores são usados em expressões e a ordem em que as expressões são executadas depende da precedência desses operadores. Veja abaixo a lista de operadores em ordem descendente de precedência.

Operador	Descrição
Operadores Unários	
@	Endereço
not	Não booleano ou bit voltado para não
Operadores Multiplicativos e de direção de Bit	
*	Multiplicação ou interseção de conjuntos
/	Divisão de Real
div	Divisão de Inteiro
mod	Resto de divisão de Inteiros
as	TypeCast seguro quanto ao tipo (RTTI)
and	E booleano ou bit voltado para e
shl	Deslocamento de bits à esquerda
shr	Deslocamento de bits à direita
Operadores Aditivos	
+	Adição ou união de conjuntos
-	Subtração ou diferença de conjuntos
or	Ou booleano ou bit voltado para ou
xor	Ou exclusivo booleano ou bit voltado para ou exclusivo

Operadores Relacionais	
=	Igual
<>	Diferente
<	Menor
>	Maior
<=	Menor ou igual
>=	Maior ou igual
in	Pertinência a conjuntos
is	Compatibilidade de tipos (RTTI)

Para forçar uma expressão de menor precedência a ser executada antes, você pode usar os parênteses, como mostrado abaixo.

```
( 5 - 2 ) * 3 ;  
( A > B ) and ( A < C )
```

Para fazer potenciação, use a função Power, abaixo temos que A é igual a A elevado a 4.

```
A := Power( A, 4 );
```

EXERCÍCIOS

- 01) Faça um Form, com um Edit e botões para passar o texto do Edit para maiúsculas e minúsculas.
- 02) Faça um programa que sugira que o usuário pare de trabalhar quando o relógio do sistema não estiver em horário comercial.
- 03) Sendo I um número inteiro, R um real e S uma string com um texto numérico como encontrar o resultado da soma dos 3 valores, e de que tipo será esse resultado? Faça isso no Delphi.

Veja o Conteúdo Exclusivo do Curso Delphi Avançado



1 CD - 130 Minutos

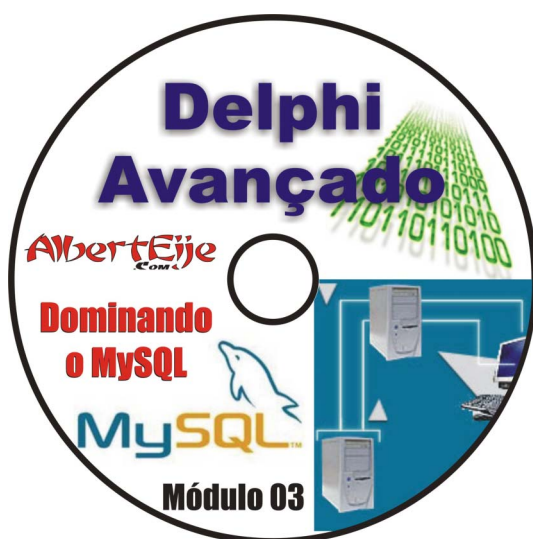
- * Compreender o modelo Entidade-Relacionamento e Lógico-Relacional;
- * Saber definir e identificar: Entidades, Relacionamentos, Atributos e tudo mais que tenha a ver com o Modelo Entidade-Relacionamento;
- * Conhecer os tipos de relacionamentos: Condicionais e Incondicionais;
- * Saber os graus de relacionamento: Um-Para-Um, Um-Para-Muitos, Muitos-Para-Muitos;
- * Compreender uma realidade e a partir dessa compreensão criar um modelo de dados e logo depois o banco de dados em si;
- * Dominar o conceito de chaves, índices e integridade referencial;
- * Saber normalizar um modelo através das Formas Normais (1FN a 5FN);
- * Usar programas para a criação do modelo de dados;
- * Utilização do DBDesigner.



1 CD - 159 Minutos

- * Definição e História da Linguagem SQL;
- * Entendendo as partes da Linguagem: DDL, DML, DCL e Transactions Control;
- * Instrução Select (alias, order by, like, asc, desc, group by, having, where, not, in, exists, sum, max, min, avg, count, between, distinct, etc);
- * Instrução Insert Into;
- * Instrução Update;
- * Instrução Delete;
- * Consultas Encadeadas (SubQueries);
- * Junções (inner join, left join, right join e full outer join);
- * Criação, alteração e deleção de tabelas e índices;
- * Controle de Transações;
- * Diversos exemplos feitos no programa SQL Explorer;
- * Criação de um sistema no Delphi implementando os comandos SQL juntamente com o componente Query. Cadastro de Clientes, Cadastro de

Usuarios e Formulário Mestre Detalhe com duas tabelas (CVenda e DVenda).



2 CDs - 340 Minutos

- * Introdução: Definição, História, Características, Licenças e Versões;
- * Instalação da versão 3.23 no Windows 98, tipos de arquivos, estrutura de diretórios do MySQL, uso do Console (prompt), etc;
- * Utilização da ferramenta gráfica MySQLFront 2.3, criação de bancos de dados, tabelas, explicação dos tipos de campos, importação de dados do paradox, atributos especiais de campos do MySQL, uso do Manual de Referência, uso das funções do MySQL;
- * Instalação da Versão 4 do MySQL no Windows XP, uso do MySQLFront 3.2, importação de arquivos do Excel, Exportação de dados para o formato HTML, etc;
- * Instalação do MySQL no Linux Slacware, uso do console no Linux, uso da ferramenta gráfica PHPMyAdmin;
- * Instalação e configuração da versão 5 no Windows XP;
- * Uso da ferramenta gráfica DBManager, tipos de tabelas, foreign keys, constraints, views, backup e restauração de dados, controle de transações;

- * Uso da ferramenta gráfica MySQL Administrator, controle de usuários, backup e restauração de dados, replicação;
- * Uso da ferramenta gráfica MySQL Query Browser;
- * Uso da ferramenta gráfica MySQL Migration Toolkit, migração de um banco de dados do MS-Access;
- * DBDesigner - Sincronização de dados e engenharia reversa;
- * Apresentação dos sites oficiais do MySQL - Internacional e Nacional;
- * Stored Procedures e Triggers.

Curso Delphi Avançado - 15 Módulos (19 CDs)



2 CDs - 254 Minutos

- * Introdução: Definição, História, Características, Licenças e Especificações Firebird;
- * Comparativo das arquiteturas Classic x SuperServer;
- * Descrição dos tipos de dados suportados pelo Firebird;
- * Dialetos;
- * Tipos de Transações;
- * Instalação passo a passo da versão 1.5 do Firebird no Windows;
- * Explicação da estrutura de diretórios;
- * Serviços executados no Windows e opções no Painel de Controle;
- * Utilização das seguintes ferramentas gráficas: IBExpert, IBEasy+, FlameRobin, IBAccess, SQLHammer, DBak Win, IB First AID, Interbase Grant Manager;
- * Criação de tabelas, triggers, generatos, domains, views, queries;
- * Controle de acesso para usuários;
- * Backup do BD;

- * Constraints: Check, PK, FK, Unique;
- * Corrigindo BDs corrompidos;
- * Criação de um projeto de BD;
- * Utilização do ISQL (console);
- * Criação de triggers para controle de estoque;
- * Exceptions;
- * Criação de Stored Procedures;
- * UDFs - Conceito; Criação de uma UDF no Delphi e uso no Firebird;
- * Corrupção de Dados: Conhecendo, evitando, corrigindo;
- * Utilização do Gfix (console);
- * Comparativo: MySQL x Firebird;
- * Apresentação dos sites importantes internacionais e nacionais;
- * Instalação do Firebird Cliente numa máquina virtual Windows 98;
- * Conexão com o banco de dados através da rede;
- * Apresentação da documentação.



1 CD - 116 Minutos

- * Definição e estrutura das empresas;
- * Níveis de decisão das empresas;
- * O Analista de Sistemas;
- * Engenharia de Software;
- * Ciclo de vida de um sistema;
- * Coleta de Informações: Entrevistas e pesquisas;
- * Entrevistas: planejamento, preparação, comportamento, linguagem, fatos x opiniões, desejos dos usuários, observações gerais;
- * Contratos e lei: análise das principais cláusulas de um contrato (identificação das partes, documentação, prazo de entrega, códigos fontes, confiabilidade);
- * Lei do Software - comentários sobre os principais artigos;
- * O que é melhor: autônomo ou empresa;
- * Estudo de caso: Sistema para Hotel - apresentação de um estudo de caso mostrando desde o início um problema enfrentado por um hotel e a

solução adotada por um analista de sistemas. Neste estudo são mostrados e comentados a proposta e o contrato utilizado pelo analista;

- * Apresentação de recibos: simples e RPA;
- * Vantagens do aluguel de software;
- * A carreira do consultor independente;
- * Comentários acerca do Guia do Empreendedor e do manual A Pequena Empresa e o Novo Código Civil;
- * Prosperidade - Mapas Mentais - explanação a apresentação do programa freemind.

Curso Delphi Avançado - 15 Módulos (19 CDs)



1 CD - 208 Minutos

- * Compreensão da realidade do SisCom;
- * Criação das tabelas (Produto - Cliente - Fornecedor - Funcionario - Banco - Departamento - Cartao - Unidade - Plano_Conta - Tipo_Pgto - CFOP - Pagamento - Recebimento - C_NFE - D_NFE - C_Venda - D_Venda - C_Requisicao - D_Requisicao - C_Cotacao - D_Cotacao - C_Pedido - D_Pedido - Configuracao - Nivel_Acesso);
- * Comentário detalhado sobre cada tabela e seus campos;
- * Apresentação do DER do SisCom e comentários sobre cada relacionamento;
- * Criação das tabelas e dos relacionamentos no DBDesigner;
- * Sincronização do DBDesigner com o MySQL;
- * Migrando o BD do MySQL para o Firebird;
- * Conectando o DBDesigner com o Firebird através de ODBC;
- * Conectando o DBDesigner com o Firebird diretamente;
- * Fazendo Engenharia Reversa de tabelas do Firebird no DBDesigner;

- * Visualizando a estrutura de arquivos do DBDesigner para fazê-lo conectar-se ao Firebird;
- * Visualizando o Código Fonte do DBDesigner dentro do Delphi;
- * Criação passo a passo do BD no Firebird utilizando o IBExpert;
- * Padronização de campos e tabelas;
- * Erros encontrados por falta de padronização.



2 CDs - 416 Minutos

- * Apresentação da Suite de Componentes ZeosDBO;
- * Vantagens de usar o ZeosDBO;
- * Instalação do ZeosDBO;
- * Explicação sobre cada componente do ZeosDBO;
- * Comparação dos componentes do ZeosDBO com os seus equivalentes no BDE;
- * Diferença entre sistemas MDI e SDI e explicação sobre qual o melhor;
- * Início do desenvolvimento do sistema SisCom;
- * Criação do Formulário Principal: Menu, Barra de Ferramentas, Barra de Status. Todos os hints do sistema serão enviados para a barra de status, possibilitando uma ajuda online através de dicas rápidas para os usuários;
- * Carregando um logotipo para a tela principal do sistema;
- * Uso de Application.MessageBox;
- * Componentes de acesso ao banco em DataModule;
- * Criação dos cadastros utilizando ZTable (equivalente a TTable). Para

quem vem do paradox ficará muito fácil desenvolver dessa forma;

- * Criação dos cadastros utilizando ZQuery (Equivalente a TQuery). Utilização de Edits Simples no lugar dos DBEdits, com procedimentos específicos para o controle de edição e gravação dos campos;
- * Utilização de comandos dos tipos TTable (Append, Edit, Post, Delete) e cadastros usando apenas SQL puro (Insert Into, Update Set, Delete From);
- * O sistema roda AUTOMATICAMENTE em Firebird e MySQL. O usuário escolhe qual banco usar. Não existe alteração nenhuma no código;
- * Utilização dos Generators (firebird) para autonumerar campos no sistema;
- * Implementação de uma rotina onde o usuário procura por qualquer informação em qualquer campo do sistema, bastando para isso clicar no título do DBGrid;
- * Criação do nível de acesso para controle de acessos dos usuários aos módulos do sistema;
- * Verificação e análise de várias mensagens de erro;
- * Implementação de um controle de auditoria, onde o sistema armazena qual o usuário que inseriu ou alterou registros, juntamente com a data e hora.

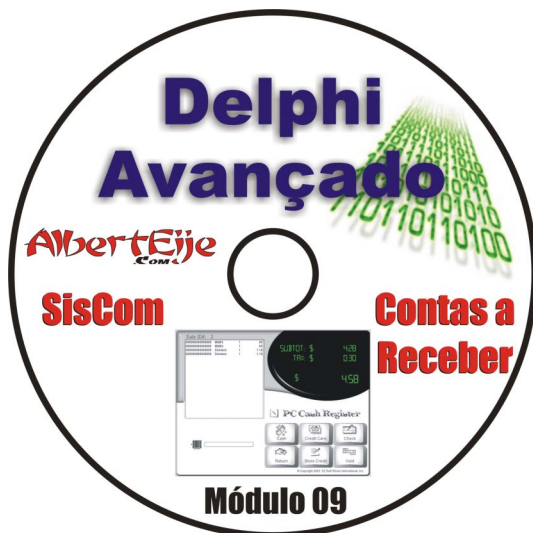
Curso Delphi Avançado - 15 Módulos (19 CDs)



1 CD - 236 Minutos

- * Introdução a tecnologia dbExpress;
- * Vantagens e desvantagens do dbExpress;
- * Comparativo dbExpress x BDE;
- * Apresentação de todos os componentes da paleta dbExpress;
- * Utilização do SQLQuery + DataSetProvider + ClientDataSet para criar o Contas a Pagar;
- * Desenvolvimento do Contas a Pagar passo a passo;
- * Acessando Firebird e MySQL ao mesmo tempo através do SQLConnection;
- * Utilizando um método de procura ao clicar no título do DBGrid, tendo por base um período passado por 2 MaskEdits;
- * Pegando o código AutoIncrement no MySQL para o ClientDataSet;
- * Pegando o código do Generator no Firebird para o ClientDataSet (Stored Procedures);
- * Explicação do uso de ActionLists para otimizar o trabalho;

* O SisCom funciona automaticamente com MySQL e Firebird. Os módulos de cadastro estão utilizando a tecnologia Zeos. O módulo de Contas a Pagar está usando a tecnologia dbExpress + ClientDataSet.



1 CD - 140 Minutos

- * Desenvolvimento do Contas a Receber do SisCom utilizando a tecnologia dbExpress + ClientDataSet;
- * Implementação do conceito de reconciliação de erros no SisCom;
- * Tradução do Formulário de Reconciliação de Erros para o Português;
- * Utilização apenas do SQLConnection para realizar consultas SQL;
- * Explicação detalhada do uso do SQLDataSet (Table, Query, StoredProc);
- * Utilização de parâmetros para realizar consultas SQL;
- * Manipulação dos dados do ClientDataSet antes de serem enviados para o Banco de Dados - ainda na memória do micro cliente (utilização da propriedade Delta do ClientDataSet);
- * Criação de consultas Mestre/Detalhe. (Dados mostrados numa mesma DBGrid e posteriormente em duas DBGrids);
- * Criação de bancos de dados locais com o ClientDataSet (utilização de arquivos XML);

- * Criação de Campos Agregados;
- * Utilização do componente SimpleDataSet;
- * Utilização do componente SQLMonitor para mapear o que está ocorrendo entre a aplicação e o banco de dados (tudo que está ocorrendo em baixo nível é mostrado num Memo);
- * Implementando o Controle de Transações com o dbExpress;
- * Explicação sobre o nível de isolamento (TransIsolationLevel) das transações do dbExpress.

Após assistir às vídeo aulas do curso você estará apto para desenvolver qualquer sistema comercial. Este conteúdo você não encontra em nenhum curso presencial ou online. Conteúdo único e exclusivo.

Curso Delphi Avançado - 15 Módulos (19 CDs)



2 CDs - 447 Minutos

- * Apresentação de cada componente da paleta ADO - principais propriedades;
- * Criando uma conexão no SisCom para Firebird e MySQL (alternando entre um e outro de acordo com a escolha do usuário);
- * Instalação dos Drivers ODBC para MySQL e Firebird e criação de DSN para acesso ao banco de dados do SisCom;
- * Criação do Módulo Requisição de Compras;
- * Utilização do ADOConnection, ADOQuery, ADOTable e ADOStoredProc;
- * Análise de erros ao construir o módulo;
- * Construção do formulário para importar itens para a requisição;
- * Trabalhando com passagem de parâmetros;
- * Localizando e eliminando erros do projeto do BD;
- * Criação do Módulo Cotação de Compras;
- * Utilização do repositório de formulários do Delphi;
- * Utilização de um componente permite abrir um menu em um botão;

- * Construção do formulário para importar os itens das requisições que vão compor a cotação;
- * Construção do Módulo Confirmação da Cotação (neste formulário o usuário insere os preços e condições enviadas pelos fornecedores);
- * Construção do Módulo Pedido de Compras (neste formulário o usuário visualiza as cotações e pode fazer pedidos aos fornecedores baseado nas informações prestadas pelos mesmos. É permitido realizar vários pedidos através de uma cotação);
- * Construção do formulário para importar os itens das cotações.



1 CD - 165 Minutos

- * Construção do Sistema de Vendas utilizando os componentes da paleta ADO (ADOConnection, ADOQuery, ADOTable, ADOStoredProc);
- * Chamadas a Stored Procedures armazenadas no BD com passagem e retorno de parâmetros;
- * Importação dos dados da tabela oficial CFOP de um BD diferente;
- * Análise de alguns erros que ocorrem com ADO;
- * Usando ADO com ClientDataSet;
- * Acessando o BD sem o uso do componente ADOConnection;
- * Construindo consultas Mestre-Detalhe com ADO (3 tabelas no exemplo);
- * ADO e Controle de Transações;
- * Enviando informações para o BD sem Result Set com o componente ADOCommand.



1 CD - 188 Minutos

- * Descrição detalhada dos 13 Componentes da paleta IBX (Interbase Express);
- * Comentários sobre suas propriedades;
- * Construção do Formulário de Entrada de Notas Fiscais (IBDataBase, IBTransaction, IBDataSet, IBQuery, IBUpdateSQL);
- * Passagem de parâmetros;
- * Atualização do Estoque via código feito no Delphi;
- * Atualização do Estoque via Triggers criadas no Firebird (AfterInsert, AfterUpdate, AfterDelete);
- * Construção do Formulário para Ajuste de Preços;
- * Código para ajustar preços na tabela de produtos utilizando uma taxa (aumentando e diminuindo valores).

Curso Delphi Avançado - 15 Módulos (19 CDs)



1 CD - 198 Minutos

- * Instalação do Quick Report no Delphi 7;
- * Descrição detalhada dos componentes da Suite Quick Report;
- * Desenvolvimento dos relatórios de cadastro do SisCom utilizando o Quick Report;
- * Criação de relatórios Mestre-Detalhe com o Quick Report;
- * Exportando relatórios para TXT e HTML através do Quick Report;
- * Descrição detalhada dos componentes da Suite Rave Reports;
- * Desenvolvimento dos relatórios de Contas a Pagar e a Receber utilizando o Rave Reports;
- * Criação de relatórios Mestre-Detalhe com o Rave Reports;
- * Utilização de imagens e códigos de barras usando o Rave Reports;
- * Exportando relatórios para PDF através do Rave Reports;
- * Apresentação do Componente PrintFast (componente que envia o relatório diretamente para a impressora - excelente para impressoras matriciais e Clippeiros de plantão);
- * Instalação do PrintFast no Delphi 7;

* Desenvolvimento dos seguintes relatórios usando o PrintFast: Requisição de Compras, Mapa Comparativo de Preços e Pedido de Compras;

* Apresentação dos Geradores de Relatórios: FreeReport e Fortes Reports.



1 CD - 168 Minutos

- * Análise de 10 Manuais em formato PDF;
- * Criação do Manual do SisCom em formato PDF;
- * Criação de um arquivo HLP (ajuda) utilizando ferramentas gratuitas;
- * Chamando a ajuda de dentro do SisCom ao pressionar a tecla F1 (ajuda sensível ao contexto);
- * Desenvolvimento de um manual HTML para o SisCom;
- * Convertendo arquivos HTML para CHM (HTML Help);
- * Convertendo arquivos CHM para 42 outros formatos (PDF, HLP, RTF, TXT, XLS, DOC e muitos outros);
- * Utilização de uma ferramenta que cria o Help e já salva em diversos formatos diferentes (HLP, CHM, PDF e outros). Ferramenta muito fácil de utilizar;
- * Chamando a ajuda CHM de dentro de uma aplicação Delphi ao pressionar F1 (ajuda sensível ao contexto);
- * Utilização de uma ferramenta que cria o arquivo de ajuda baseado nas

imagens capturadas das telas do SisCom. O usuário clica em qualquer parte das janelas capturadas e a ajuda referente àquela parte da janela é apresentada.



1 CD - 175 Minutos

- * Modificando o código do SisCom para que o mesmo funcione em rede;
- * Testando o funcionamento do SisCom em rede (máquina virtual);
- * Identificação das bibliotecas necessárias para a instalação numa máquina cliente;
- * Trabalhando com arquivos Ini (configuração do acesso aos dados através desses arquivos);
- * Desenvolvimento de um instalador personalizado no Delphi (o instalador se encarrega de copiar todos os arquivos necessários para o funcionamento do SisCom e instala o MySQL e o Firebird);
- * Utilização de ferramentas para criação de instaladores: Setup2Go, CreateInstall, InnoSetup;
- * Compactação de executáveis;
- * Criando uma biblioteca de funções e integrando a mesma ao SisCom;
- * Aperfeiçoamento da janela de Login;
- * Desenvolvimento de uma Splash Screen para o SisCom utilizando uma

imagem feita no CorelDraw e exportada para o SisCom. Utilização de uma barra de progresso.

Curso Delphi Avançado - Módulos Plus



2 CDs - 224 Minutos

- * Introdução a Orientação a Objetos
- * Histórico do Paradigma de Orientação a Objetos
- * Explicações sobre: Abstração; Objeto; Encapsulamento; Mensagem; Classe; Herança; Polimorfismo; Classificação; Associação; Agregação; Generalização; Especialização
- * Introdução a UML - Unified Modeling Language
- * Histórico da notação UML
- * Fases do desenvolvimento de um sistema em UML: Análise de requisitos; Análise; Design (projeto); Programação; Testes
- * Visões: Visão de Componentes; Visão Lógica; Visão de Use-Case; Visão de Organização; Visão de Concorrência
- * Modelos de Elementos: Classes; Objetos; Estado; Pacote; Componente; Relacionamentos (associação, generalização, dependência e refinamento)
- * Diagramas: Use-Case; Classes; Objetos; Estado; Seqüência;

Colaboração; Atividade; Componente; Execução

* Estudos de caso: conta corrente e aplicações financeiras; locadora; sistema de matrícula; posto de gasolina; biblioteca.

* Cada estudo de caso acima é feito em uma ferramenta diferente.

Obs: Este Módulo é pré-requisito para o Módulo Delphi OOP



1 CD - 220 Minutos

- * Introdução à Linguagem Object Pascal (Delphi Language);
- * Variáveis - declaração e exemplos; Variáveis Locais e Globais;
- * Constantes - declaração e exemplos;
- * Tipos de Dados - Inteiros, Reais, Texto, Ordinais;
- * Rotinas de conversão de tipos; TypeCasting;
- * Operadores - unários, multiplicativos, direção de bits, aditivos, relacionais;
- * Estruturas de Repetição - While, For, Repeat; Quebras de Laço (continue, break, exit, halt);
- * Tipos Definidos pelo Usuários - SubRange, Enumerações, Ponteiros, Records, Arrays, Sets;
- * Procedimentos, Funções e Métodos;
- * Passagem de parâmetros - explicações sobre passagem por valor e por referência; Uso do With;
- * Desenvolvimento de diversas rotinas para praticar os conceitos acima;
- * Explicações sobre a estrutura da Unit: Áreas: Unit, Interface,

Implementation, Var, Const, Uses, Referência Circular;

* Sobrecarga de métodos: Overload - Exemplo de vários métodos de mesmo nome para mostrar o funcionamento;

* Blocos protegidos - explicações sobre exceções (try, except, finally, raise);

* Explicações sobre parâmetros formais e parâmetros reais;

* Classes e Objetos; Desenvolvimento de diversas classes no Delphi;

* Métodos construtores e destrutores; Métodos Getters e Setters;

* Instanciação das classes - Desenvolvimento de uma aplicação Console para analisar o funcionamento;

* Encapsulamento - Operadores de visibilidade (private, public, protected, published) - explicações e prática de cada um deles;

* Herança entre classes; * Utilização de Property (propriedades);

* Sobreposição de Métodos - Virtual/Override - Sobrescrever métodos em classes descendentes - Conceito de Polimorfismo;

* Desenvolvimento do SisCom OO - explicações sobre o objetivo do SisCom (Sistema Comercial);

* Exibição do DER e da UML do SisCom. Diferenças entre os dois diagramas;

* Apresentação do SisCom desenvolvido durante o Curso Delphi Avançado;

* Desenvolvimento passo a passo do Formulário Padrão de Cadastro. Através desse formulário, todos os outros forms de cadastro serão criados. Utilização do conceito de Herança Visual do Delphi;

* Integração Delphi / ModelMaker; * Engenharia reversa do SisCom para o ModelMaker;

* Alterações em modelo no ModelMaker e atualização em Tempo Real na aplicação no Delphi;

* Explicações para desenvolvimento dos diagramas UML do SisCom no ModelMaker;

Obs: O Módulo Plus Entendendo UML é pré-requisito para este módulo

Curso Delphi Avançado - Módulos Plus



2 CDs - 339 Minutos

- * Explicações sobre ECF (Emissor de Cupom Fiscal) e TEF (Transferência Eletrônica de Fundos);
- * Instalação e utilização de Emulador de Impressora Fiscal para utilização junto com o sistema de Frente de Caixa;
- * Análise dos manuais das Impressoras Bematech, Yanco, Daruma e Zanthus;
- * Criação passo a passo da Interface do Sistema de Frente de Caixa;
- * Implementação dos comandos para utilização com a impressora fiscal: Leitura X, Redução Z, Relatórios de Memória Fiscal; Abertura e Fechamento de Cupons; Envio de Formas de Pagamento; Envio e Cancelamento de Itens; Etc;
- * Criação de um formulário de configurações onde o próprio usuário define as formas de pagamentos que vai utilizar, em quais ocasiões vai solicitar senhas, qual impressora está utilizando e o caminho onde as fotos dos produtos estão armazenadas (as fotos dos produtos são armazenadas

fora do banco de dados);

- * Criação de um formulário para localização de produtos, caso o sistema não consiga ler o código ou o produto não tenha código de barras;
- * Criação de um formulário para armazenar os dados da venda;
- * Estudo e implementação da Solução TEF;



2 CDs - 275 Minutos

- * SQL Server - Apresentação - Características;
- * Instalação passo a passo do SQL Server e Ferramentas Avançadas - Instalação do .NET Framework;
- * Apresentação da ferramenta de configuração (SQL Server Configuration Manager);
- * Apresentação e configuração do SQL Browser - ferramenta para aceitar conexões remotas;
- * Explicação e configuração dos tipos de conexões (Shared Memory, Named Pipes, TCP/IP, VIA);
- * Criação de aliases para o Banco de Dados;
- * Apresentação da ferramenta de configuração (SQL Server Surface Area Configuration);
- * Configuração e uso da ferramenta para gerenciamento de bancos de dados feitos em SQL Server (Microsoft SQL Server Management Studio Express);

- * Explicação sobre Collations; Detalhamento sobre os tipos de dados;
- * Criação e configuração de um banco de dados;
- * Criação das tabelas, índices, relacionamentos, constraints, etc do banco de dados utilizando ferramenta para modelagem;
- * Criação do campo Auto-Increment; Criação de Views atualizáveis;
- * Apresentação de ferramenta gráfica adicional para gerenciamento do banco de dados (SQL Manager 2005 for SQL Server);
- * Criação de Checks Constraints; Criação de Usuários/Roles para acesso ao banco;
- * Confeção de relatórios utilizando a ferramenta;
- * Exportando dados (Excel, Access, Word, RTF, HTML, PDF, TXT, CSV, DBF, XML);
- * Importando dados (Excel, Access, DBF, XML, TXT, CSV);
- * Explicação sobre Stored Procedures e Triggers;
- * Operações com o Banco (extração de dados, backup e restore);
- * Apresentação de ferramenta gráfica adicional para gerenciamento do banco de dados (SQL Lite);
- * Conhecendo as funções de sistema (agregação, matemáticas, data e hora, string, diversas...);
- * Desenvolvimento de uma aplicação do Delphi para acesso ao banco de dados (cadastros, vendas, etc). Utilização das tecnologias de acesso ADO e dbExpress;
- * Formulário mestre-detelhe e campos de lookup; Explicação e criação de Trigger para atualização do estoque;
- * Utilização de transações (início, cancelamento e confirmação);
- * Funcionamento da aplicação em rede através de máquina virtual W98;

Curso Delphi Avançado - Módulos Plus



1 CD - 129 Minutos

- * Introdução a Criptologia (Criptografia e Criptoanálise);
- * Utilizando na prática a Esteganografia (esconder arquivos em imagens);
- * Análise de vários componentes que implementam a Criptografia no Delphi;
- * Apresentação de vários métodos para Proteção de Programas (Nag-Screen, Período, Limitação de Registros, HardLocks, Identidade Única);
- * Implementando na prática várias técnicas para proteção de programas (Por Arquivo, Por Diretório, Por Registro, etc);
- * Desenvolvimento de um sistema que bloqueia o uso do sistema no usuário. O sistema grava as informações no registro do Windows. Pega informações da máquina do usuário (HD, BIOS) e combina tais informações com Constantes implementadas pelo programador (Phi e Pi). O desbloqueio do sistema se dá ao informar a Contra-Chave (Demonstrado passo a passo);
- * Apresentação de vários componentes para Delphi que implementam a

proteção de programas;

- * Dicas importantes para colocar em prática no momento da criação de uma proteção;
- * Apresentação de Engenharia Reversa (Descompiladores);
- * Utilização na prática de alguns descompiladores (DeDe, Source Rescuer, ResHacker);
- * Demonstração de técnica para evitar a Engenharia Reversa de programas feitos em Delphi;
- * Demonstração de como Crackear um programa feito em Delphi utilizando um Editor HexaDecimal (passo a passo);



1 CD - 229 Minutos

- * MSAccess - Apresentação;
- * Criação de Tabelas (explicação dos tipos de dados);
- * Especificações do Access - limitações do banco de dados;
- * Definindo relacionamentos e criando integridade referencial. Explicações sobre cardinalidade;
- * Criação de consultas (Visual e SQL);
- * Criação de formulários. Formulários prontos para impressão. Formulários vinculados a código em VBA. Subformulários;
- * Criação de uma consulta vinculada a dados de um formulário. Visualização dos dados de uma consulta em um subformulário;
- * Desenvolvimento de relatórios;
- * Desenvolvimento de uma página WEB que acessa e altera os dados da tabela;
- * Introdução a VBA (Visual Basic for Applications);
- * Acessando o banco de dados através do Delphi (tecnologia DAO);
- * Criação de um pequeno aplicativo no Delphi acessando o BD Access.

Utilização de Acion Lists;

- * Demonstração do Assistente de tabelas do Access que ajuda a normalizar o Banco de Dados;
- * Demonstração do Analisador de Desempenho do Access;
- * Definindo e alterando senhas no Access;
- * Acessando os relatórios desenvolvidos no Access através do Delphi: visualização em tela e envio para impressão sem que o usuário perceba que o relatório foi feito no Access;
- * Criação de índices primários e secundários;
- * Introdução ao conceito de boletos bancários;
- * Apresentação do Manual da Federação Brasileira dos Bancos;
- * Criação passo a passo de um registro de 240 posições definido no manual da Febraban;
- * Apresentação de uma solução pronta (componente open source) para a impressão de boletos bancários em Delphi, emissão do arquivo de remessa que é enviado pelas empresas aos bancos e leitura do arquivo de retorno que é enviado pelos bancos para as empresas;
- * Instalação e utilização do referido componente: impressão de boletos bancários em QuickReport (visualização personalizada e RaveReports, exportação do boleto para PDF, geração do arquivo de remessa);
- * Demonstrando a impressão de boletos bancários com dados vindos de uma tabela do banco de dados Access;
- * Orientações para migrar o componente para uma versão posterior do Delphi;