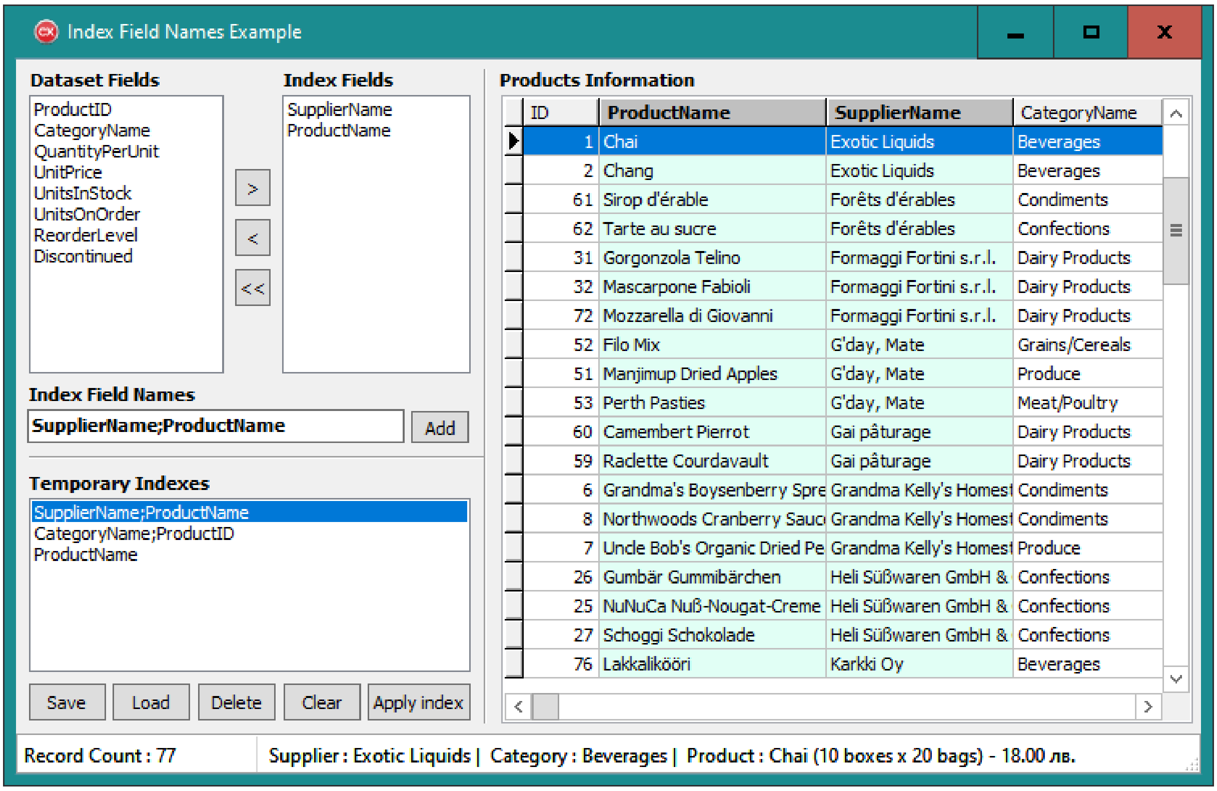
10

Разработване на приложение **IndexFieldNamesExample**

В настоящето упражнение ще бъде представено разработването на приложение за работа с бази от данни, което ще използва възможностите на наборите от данни за работа с индекси. Сесия на работа с приложението **IndexFieldNamesExample** е показана на Фиг. 1.



Dataset Fields

ProductID
CategoryName
QuantityPerUnit
UnitPrice
UnitsInStock
UnitsOnOrder
ReorderLevel
Discontinued

Index Fields

SupplierName
ProductName

Index Field Names

SupplierName;ProductName

Temporary Indexes

SupplierName;ProductName
CategoryName;ProductID
ProductName

Products Information

ID	ProductName	SupplierName	CategoryName
1	Chai	Exotic Liquids	Beverages
2	Chang	Exotic Liquids	Beverages
61	Sirop d'érable	Forêts d'érables	Condiments
62	Tarte au sucre	Forêts d'érables	Confections
31	Gorgonzola Telino	Formaggi Fortini s.r.l.	Dairy Products
32	Mascarpone Fabioli	Formaggi Fortini s.r.l.	Dairy Products
72	Mozzarella di Giovanni	Formaggi Fortini s.r.l.	Dairy Products
52	Filo Mix	G'day, Mate	Grains/Cereals
51	Manjimup Dried Apples	G'day, Mate	Produce
53	Perth Pasties	G'day, Mate	Meat/Poultry
60	Camembert Pierrot	Gai pâturage	Dairy Products
59	Raclette Courdavault	Gai pâturage	Dairy Products
6	Grandma's Boysenberry Spre	Grandma Kelly's Homest	Condiments
8	Northwoods Cranberry Sauc	Grandma Kelly's Homest	Condiments
7	Uncle Bob's Organic Dried Pe	Grandma Kelly's Homest	Produce
26	Gumbär Gummibärchen	Heli Süßwaren GmbH &	Confections
25	NuNuCa Nuß-Nougat-Creme	Heli Süßwaren GmbH &	Confections
27	Schoggi Schokolade	Heli Süßwaren GmbH &	Confections
76	Lakkalikööri	Karkki Oy	Beverages

Record Count : 77 | Supplier : Exotic Liquids | Category : Beverages | Product : Chai (10 boxes x 20 bags) - 18.00 лв.

Фигура 1: Сесия на работа с приложението **IndexFieldNamesExample**.

Приложението **IndexFieldNamesExample** ще използва данни, които са съхранени във външен файл, форматиран в xml формат - **ProductsEx.xml**.

СЪЗДАВАНЕ НА ПРИЛОЖЕНИЕТО IndexFieldNamesExample

Стартирайте RAD Studio и изпълнете следните стъпки:

Стъпка 1: От главното меню изберете **File ► New ► Windows VCL Application – Delphi (или C++ Builder)** в зависимост от езика, който искате да използвате.

Стъпка 2: От главното меню изберете **File ► Save All...** Задайте име на модула, който ще съдържа кода с имплементация на главната форма, което да бъде: **MainUnit.pas** (Delphi) или **MainUnit.cpp** (C++ Builder), проекта запишете като: **IndexFieldNamesExample.dproj** (Delphi) или **IndexFieldNamesExample.cbproj** (за C++ Builder).

Стъпка 3: **Главната форма** на приложението трябва да има настройки и компоненти, каквито са показани по-долу:

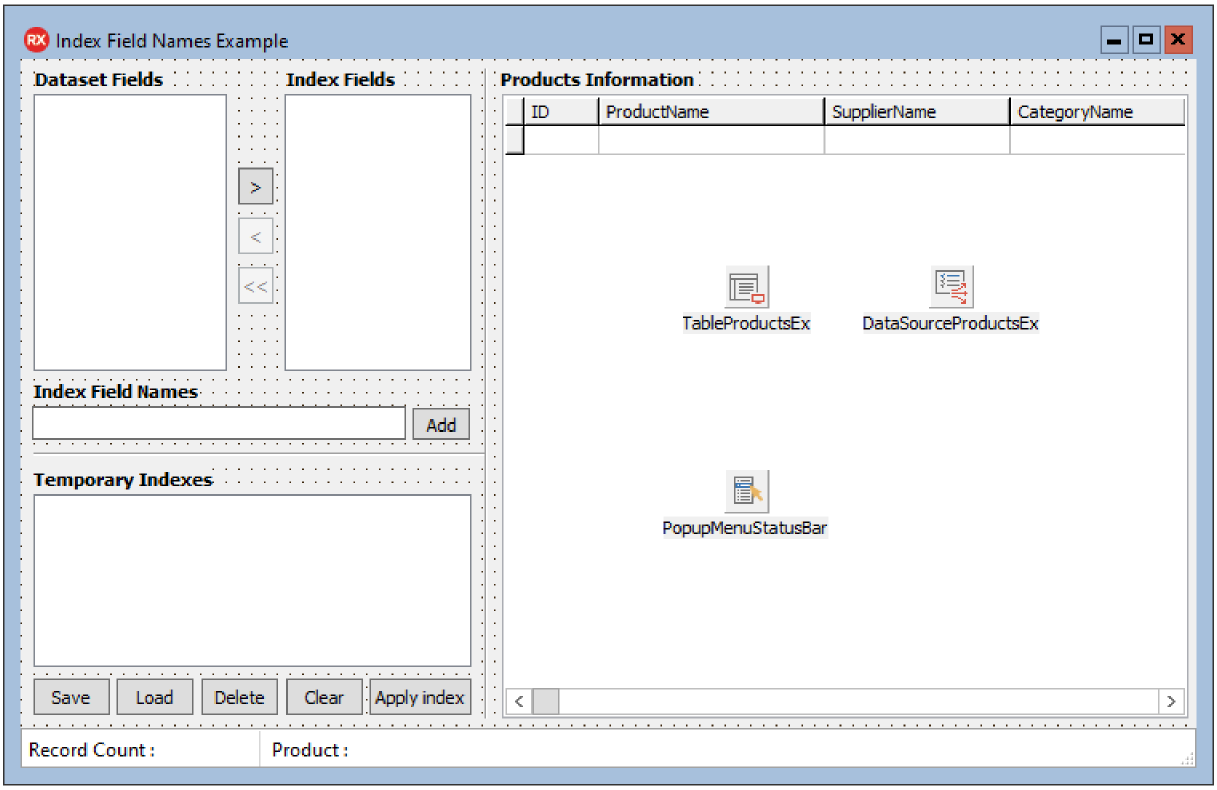
```
object FormMain: TFormMain
  Left = 0
  Top = 0
  Caption = 'Index Field Names Example'
  ClientHeight = 443
  ClientWidth = 735
  Color = clBtnFace
  Constraints.MinHeight = 460
  Constraints.MinWidth = 640
  Font.Charset = DEFAULT_CHARSET
  Font.Color = clWindowText
  Font.Height = -11
  Font.Name = 'Tahoma'
  Font.Style = []
  OldCreateOrder = False
  Position = poScreenCenter
  OnCreate = FormCreate
  OnDestroy = FormDestroy
  DesignSize = (
    735
    443)
  PixelsPerInch = 96
  TextHeight = 13

  object BevelLeft: TBevel
    Left = 290
    Top = 6
    Width = 5
    Height = 406
    Anchors = [akLeft, akTop, akBottom]
    Shape = bsLeftLine
    ExplicitHeight = 596
  end
```

```
object Bevel2: TBevel
  Left = 8
  Top = 246
  Width = 282
  Height = 5
  Shape = bsTopLine
end

object LabelIndexFieldNames: TLabel
  Left = 8
  Top = 201
  Width = 103
  Height = 13
  Caption = 'Index Field Names'
  Font.Charset = DEFAULT_CHARSET
  Font.Color = clWindowText
  Font.Height = -11
  Font.Name = 'Tahoma'
  Font.Style = [fsBold]
  ParentFont = False
end

object LabelDataSetFields: TLabel
  Left = 9
  Top = 6
  Width = 80
  Height = 13
  Caption = 'Dataset Fields'
  Font.Charset = DEFAULT_CHARSET
  Font.Color = clWindowText
  Font.Height = -11
  Font.Name = 'Tahoma'
  Font.Style = [fsBold]
  ParentFont = False
end

object LabelIndexFields: TLabel
  Left = 166
  Top = 6
  Width = 68
  Height = 13
  Caption = 'Index Fields'
  Font.Charset = DEFAULT_CHARSET
  Font.Color = clWindowText
  Font.Height = -11
  Font.Name = 'Tahoma'
  Font.Style = [fsBold]
  ParentFont = False
end

object Label2: TLabel
  Left = 300
  Top = 6
```

```
    Width = 121
    Height = 13
    Caption = 'Products Information'
    Font.Charset = DEFAULT_CHARSET
    Font.Color = clWindowText
    Font.Height = -11
    Font.Name = 'Tahoma'
    Font.Style = [fsBold]
    ParentFont = False
  end

  object Label3: TLabel
    Left = 8
    Top = 256
    Width = 112
    Height = 13
    Caption = 'Temporary Indexes'
    Font.Charset = DEFAULT_CHARSET
    Font.Color = clWindowText
    Font.Height = -11
    Font.Name = 'Tahoma'
    Font.Style = [fsBold]
    ParentFont = False
  end

  object StatusBar1: TStatusBar
    Left = 0
    Top = 418
    Width = 735
    Height = 25
    Panels = <
      item
        Text = ' Record Count : '
        Width = 150
      end
      item
        Text = ' Product : '
        Width = 50
      end>
    PopupMenu = PopupMenuStatusBar
    ExplicitTop = 526
    ExplicitWidth = 973
  end

  object DBGrid: TDBGrid
    Left = 301
    Top = 22
    Width = 429
    Height = 390
    Anchors = [akLeft, akTop, akRight, akBottom]
    DataSource = DataSourceProductsEx
    DrawingStyle = gdsClassic
    Options =
     [dgTitles, dgIndicator, dgColLines, dgRowLines, dgTabs, dgRowSelect,
      dgAlwaysShowSelection, dgConfirmDelete, dgCancelOnExit, dgTitleClick]
    TabOrder = 1
    TitleFont.Charset = DEFAULT_CHARSET
    TitleFont.Color = clWindowText
    TitleFont.Height = -11
    TitleFont.Name = 'Tahoma'
    TitleFont.Style = []
    OnTitleClick = DBGridTitleClick
    Columns = <
      item
        Expanded = False
        FieldName = 'ProductID'
        Title.Caption = ' ID'
        Width = 46
        Visible = True
      end
      item
        Expanded = False
        FieldName = 'ProductName'
        Title.Caption = ' ProductName'
        Width = 140
        Visible = True
      end
      item
        Expanded = False
        FieldName = 'SupplierName'
        Title.Caption = ' SupplierName'
        Width = 115
        Visible = True
      end
      item
        Expanded = False
        FieldName = 'CategoryName'
        Title.Caption = ' CategoryName'
        Width = 110
        Visible = True
      end
      item
        Expanded = False
        FieldName = 'QuantityPerUnit'
        Title.Caption = ' QuantityPerUnit'
        Width = 120
        Visible = True
      end
      item
        Expanded = False
        FieldName = 'UnitPrice'
        Title.Caption = ' UnitPrice'
        Width = 70
        Visible = True
      end
      item
        Expanded = False
        FieldName = 'UnitsInStock'
        Title.Caption = ' UnitsInStock'
        Width = 90
        Visible = True
      end
      item
        Expanded = False
        FieldName = 'UnitsOnOrder'
        Title.Caption = ' UnitsOnOrder'
        Width = 94
        Visible = True
      end
      item
        Expanded = False
        FieldName = 'ReorderLevel'
        Title.Caption = ' ReorderLevel'
        Width = 90
        Visible = True
      end
      item
        Expanded = False
        FieldName = 'Discontinued'
        Title.Caption = ' Discontinued'
        Width = 90
        Visible = True
      end>
  end

  object ListBoxDataSetFields: TListBox
    Left = 8
    Top = 22
    Width = 121
    Height = 173
    IntegralHeight = True
    ItemHeight = 13
    TabOrder = 2
    OnDblClick = ListBoxDataSetFieldsDblClick
    OnDragDrop = ListBoxDataSetFieldsDragDrop
    OnDragOver = ListBoxDataSetFieldsDragOver
    OnEnter = ListBoxDataSetFieldsEnter
    OnMouseMove = ListBoxDataSetFieldsMouseMove
  end

  object ListBoxIndexFields: TListBox
    Left = 165
    Top = 22
    Width = 117
    Height = 173
    IntegralHeight = True
    ItemHeight = 13
    TabOrder = 3
```

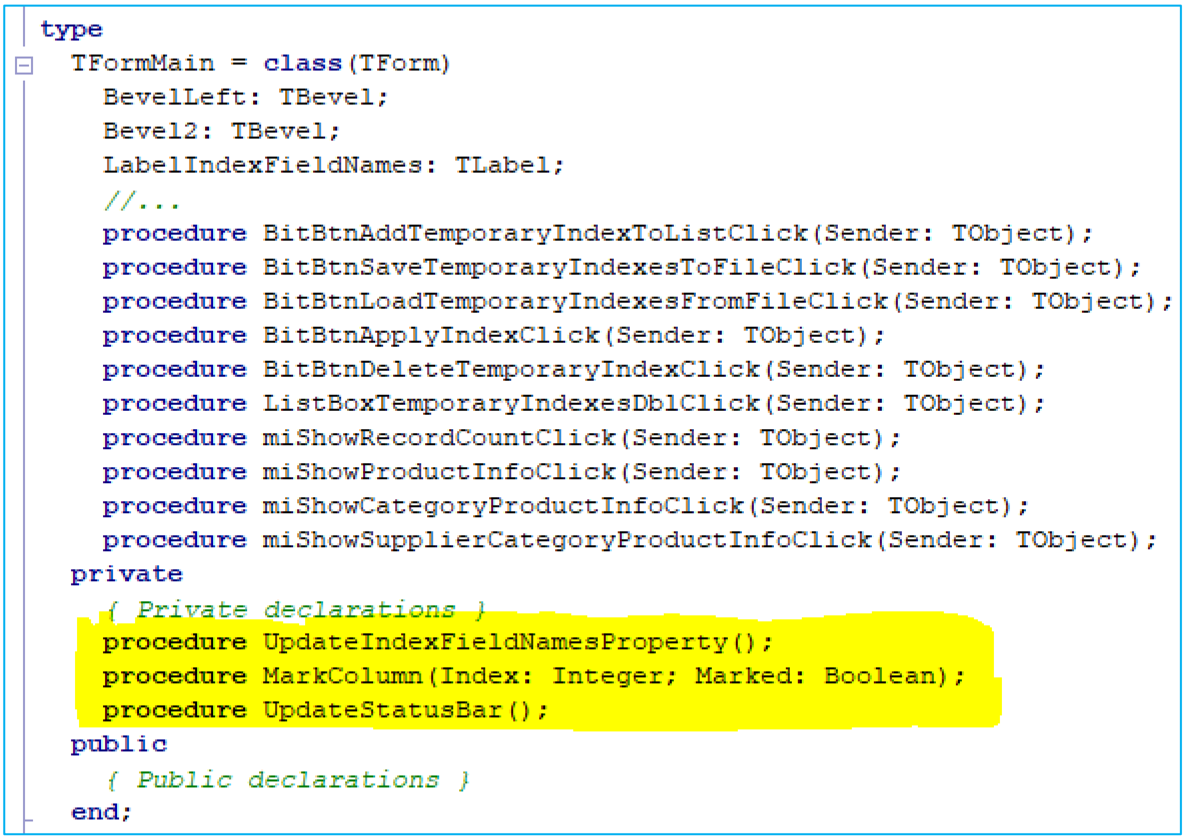
```
    OnDblClick = ListBoxIndexFieldsDblClick
    OnDragDrop = ListBoxIndexFieldsDragDrop
    OnDragOver = ListBoxIndexFieldsDragOver
    OnEnter = ListBoxIndexFieldsEnter
    OnMouseMove = ListBoxIndexFieldsMouseMove
  end

  object BitBtnAddField: TBitBtn
    Left = 135
    Top = 67
    Width = 24
    Height = 25
    Caption = '>'
    TabOrder = 4
    OnClick = BitBtnAddFieldClick
  end

  object BitBtnRemoveField: TBitBtn
    Left = 135
    Top = 98
    Width = 24
    Height = 25
    Caption = '<'
    Enabled = False
    TabOrder = 5
    OnClick = BitBtnRemoveFieldClick
  end

  object EditIndexFieldNames: TEdit
    Left = 7
    Top = 217
    Width = 234
    Height = 21
    Font.Charset = DEFAULT_CHARSET
    Font.Color = clWindowText
    Font.Height = -11
    Font.Name = 'Tahoma'
    Font.Style = [fsBold]
    ParentFont = False
    ReadOnly = True
    TabOrder = 6
  end

  object ListBoxTemporaryIndexes: TListBox
    Left = 8
    Top = 272
    Width = 274
    Height = 108
    Anchors = [akLeft, akTop, akBottom]
    IntegralHeight = True
    ItemHeight = 13
    TabOrder = 7
    OnDblClick = ListBoxTemporaryIndexesDblClick
    ExplicitHeight = 160
  end

  object BitBtnRemoveAllField: TBitBtn
    Left = 135
    Top = 129
    Width = 24
    Height = 25
    Caption = '<<'
    Enabled = False
    TabOrder = 8
    OnClick = BitBtnRemoveAllFieldClick
  end

  object BitBtnSaveTemporaryIndexesToFile: TBitBtn
    Left = 7
    Top = 386
    Width = 50
    Height = 25
    Anchors = [akLeft, akBottom]
    Caption = 'Save'
    TabOrder = 9
    OnClick = BitBtnSaveTemporaryIndexesToFileClick
    ExplicitTop = 438
  end

  object BitBtnLoadTemporaryIndexesFromFile: TBitBtn
    Left = 59
    Top = 386
    Width = 50
    Height = 25
    Anchors = [akLeft, akBottom]
    Caption = 'Load'
    TabOrder = 10
    OnClick = BitBtnLoadTemporaryIndexesFromFileClick
    ExplicitTop = 438
  end

  object BitBtnClearListBoxTemporaryIndexes: TBitBtn
    Left = 165
    Top = 386
    Width = 50
    Height = 25
    Anchors = [akLeft, akBottom]
    Caption = 'Clear'
    TabOrder = 11
    OnClick = BitBtnClearListBoxTemporaryIndexesClick
    ExplicitTop = 438
  end

  object BitBtnAddTemporaryIndexToList: TBitBtn
    Left = 244
    Top = 217
    Width = 38
    Height = 22
    Caption = 'Add'
    TabOrder = 12
    OnClick = BitBtnAddTemporaryIndexToListClick
  end

  object BitBtnApplyIndex: TBitBtn
    Left = 217
    Top = 386
    Width = 66
    Height = 25
    Anchors = [akLeft, akBottom]
    Caption = 'Apply index'
    TabOrder = 13
    OnClick = BitBtnApplyIndexClick
    ExplicitTop = 438
  end

  object BitBtnDeleteTemporaryIndex: TBitBtn
    Left = 112
    Top = 386
    Width = 50
    Height = 25
    Anchors = [akLeft, akBottom]
    Caption = 'Delete'
    TabOrder = 14
    OnClick = BitBtnDeleteTemporaryIndexClick
    ExplicitTop = 438
  end

  object TableProductsEx: TClientDataSet
    Aggregates = <>
    FileName = 'ProductsEx.xml'
    FieldDefs = <>
    IndexDefs = <>
    Params = <>
    StoreDefs = True
    AfterScroll = TableProductsExAfterScroll
    Left = 440
    Top = 128

    object TableProductsExProductID: TAutoIncField
      FieldName = 'ProductID'
      ReadOnly = True
    end

    object TableProductsExProductName: TWideStringField
      FieldName = 'ProductName'
      Size = 40
    end
    object TableProductsExSupplierName: TWideStringField
      FieldName = 'SupplierName'
      Size = 40
    end

    object TableProductsExCategoryName: TWideStringField
      FieldName = 'CategoryName'
      Size = 15
    end

    object TableProductsExQuantityPerUnit: TWideStringField
      FieldName = 'QuantityPerUnit'
    end

    object TableProductsExUnitPrice: TBCDField
      FieldName = 'UnitPrice'
      Precision = 19
    end

    object TableProductsExUnitsInStock: TSmallintField
      FieldName = 'UnitsInStock'
    end

    object TableProductsExUnitsOnOrder: TSmallintField
      FieldName = 'UnitsOnOrder'
    end

    object TableProductsExReorderLevel: TSmallintField
      FieldName = 'ReorderLevel'
    end

    object TableProductsExDiscontinued: TBooleanField
      FieldName = 'Discontinued'
    end
  end

  object DataSourceProductsEx: TDataSource
    DataSet = TableProductsEx
    Left = 568
    Top = 128
  end

  object PopupMenuStatusBar: TPopupMenu
    Left = 440
    Top = 256
    object miShowRecordCount: TMenuItem
      Caption = 'Show Record Count'
      Checked = True
      OnClick = miShowRecordCountClick
    end
    object N1: TMenuItem
      Caption = '-'
    end
    object miShowProductInfo: TMenuItem
      Caption = 'Show Product Info'
      GroupIndex = 100
      RadioItem = True
      OnClick = miShowProductInfoClick
    end

    object miShowCategoryProductInfo: TMenuItem
      Caption = 'Show Category & Product Info'
      GroupIndex = 100
      RadioItem = True
      OnClick = miShowCategoryProductInfoClick
    end

    object miShowSupplierCategoryProductInfo: TMenuItem
      Caption = 'Show Supplier & Category & Product Info'
      Checked = True
      GroupIndex = 100
      RadioItem = True
      OnClick = miShowSupplierCategoryProductInfoClick
    end
  end
end
```

След като бъдат добавени и настроени всички компоненти към формата, тя трябва да изглежда подобно на тази показана на Фиг. 2.



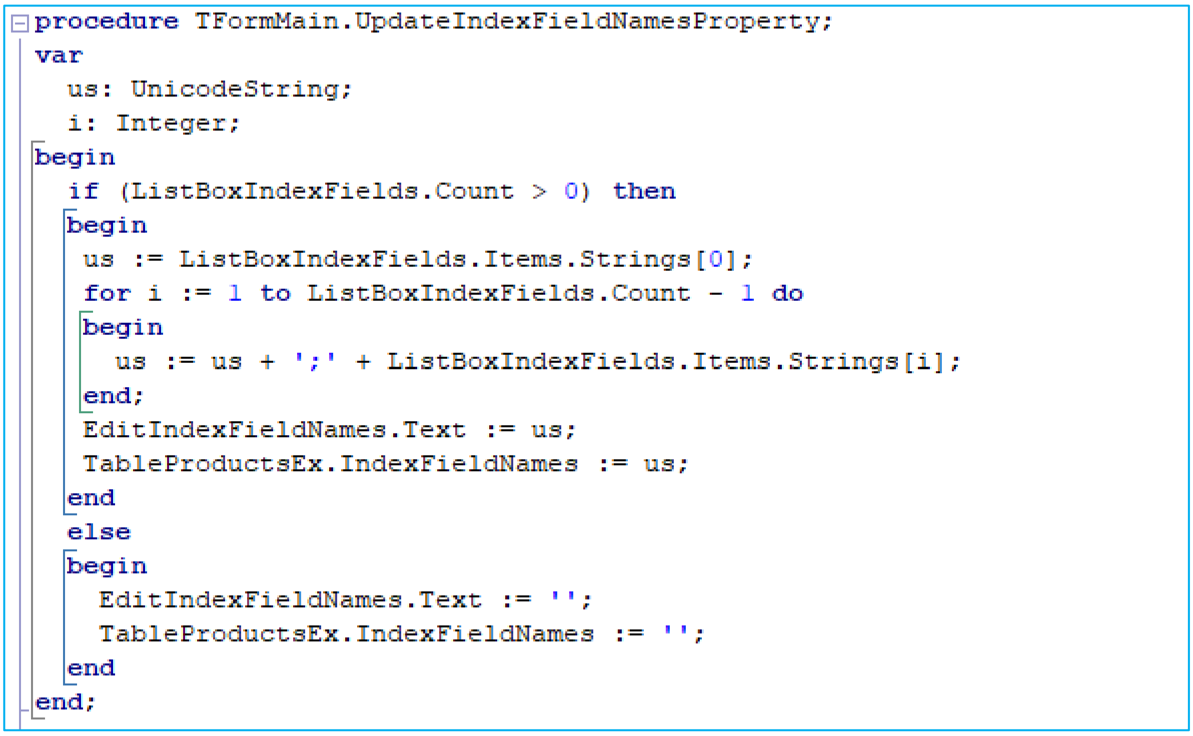
Фигура 2: Приложението **IndexFieldNamesExample** при проектиране.

<u>Стъпка 4</u>: В декларацията на класа TMainForm добавете в частта **private** 3 метода, както е показано по-долу:

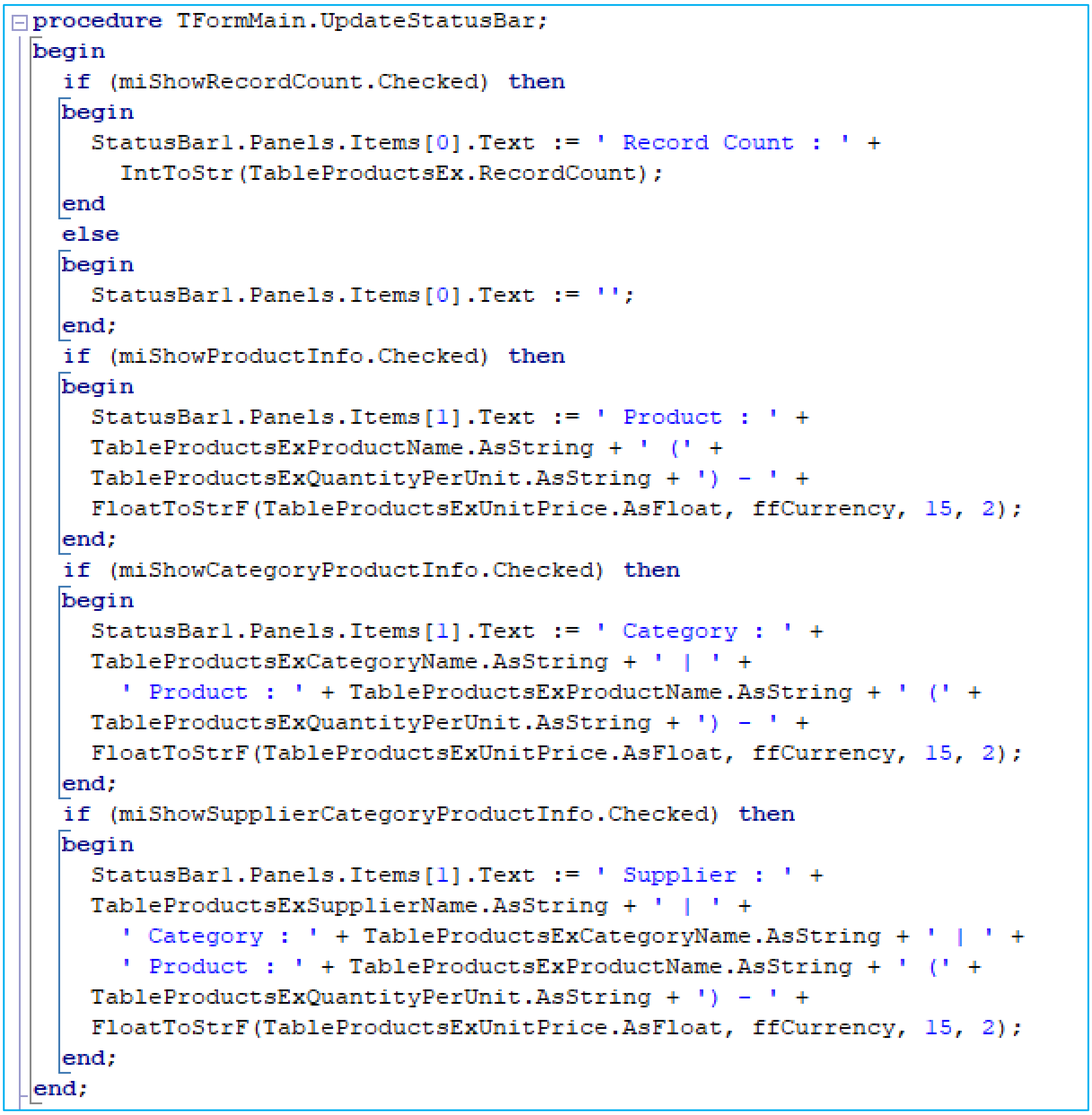


```
type
  TFormMain = class(TForm)
    BevelLeft: TBevel;
    Bevel2: TBevel;
    LabelIndexFieldNames: TLabel;
    //...
    procedure BitBtnAddTemporaryIndexToListClick(Sender: TObject);
    procedure BitBtnSaveTemporaryIndexesToFileClick(Sender: TObject);
    procedure BitBtnLoadTemporaryIndexesFromFileClick(Sender: TObject);
    procedure BitBtnApplyIndexClick(Sender: TObject);
    procedure BitBtnDeleteTemporaryIndexClick(Sender: TObject);
    procedure ListBoxTemporaryIndexesDblClick(Sender: TObject);
    procedure miShowRecordCountClick(Sender: TObject);
    procedure miShowProductInfoClick(Sender: TObject);
    procedure miShowCategoryProductInfoClick(Sender: TObject);
    procedure miShowSupplierCategoryProductInfoClick(Sender: TObject);
  private
    { Private declarations }
    procedure UpdateIndexFieldNamesProperty();
    procedure MarkColumn(Index: Integer; Marked: Boolean);
    procedure UpdateStatusBar();
  public
    { Public declarations }
  end;
```

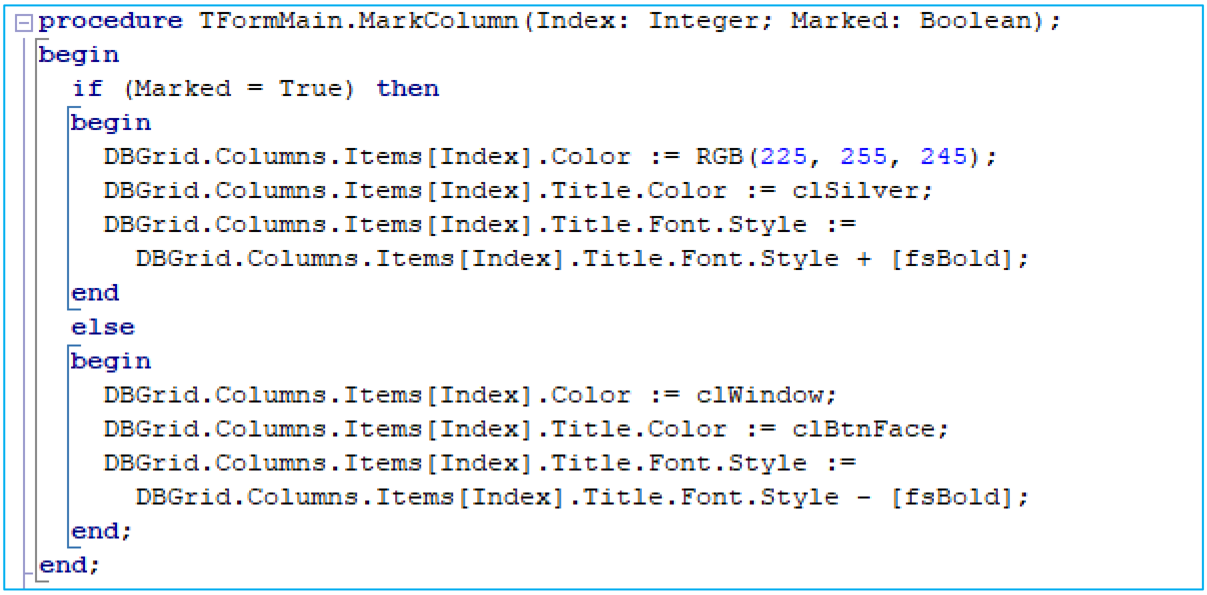
<u>Стъпка 5</u>: Във файла MainUnit, добавете реализация на декларираните в интерфейсната част 3 метода, както е показано по-долу:



```
procedure TFormMain.UpdateIndexFieldNamesProperty;
var
  us: UnicodeString;
  i: Integer;
begin
  if (ListBoxIndexFields.Count > 0) then
  begin
   us := ListBoxIndexFields.Items.Strings[0];
   for i := 1 to ListBoxIndexFields.Count - 1 do
   begin
     us := us + ';' + ListBoxIndexFields.Items.Strings[i];
   end;
   EditIndexFieldNames.Text := us;
   TableProductsEx.IndexFieldNames := us;
  end
  else
  begin
    EditIndexFieldNames.Text := '';
    TableProductsEx.IndexFieldNames := '';
  end
end;
```

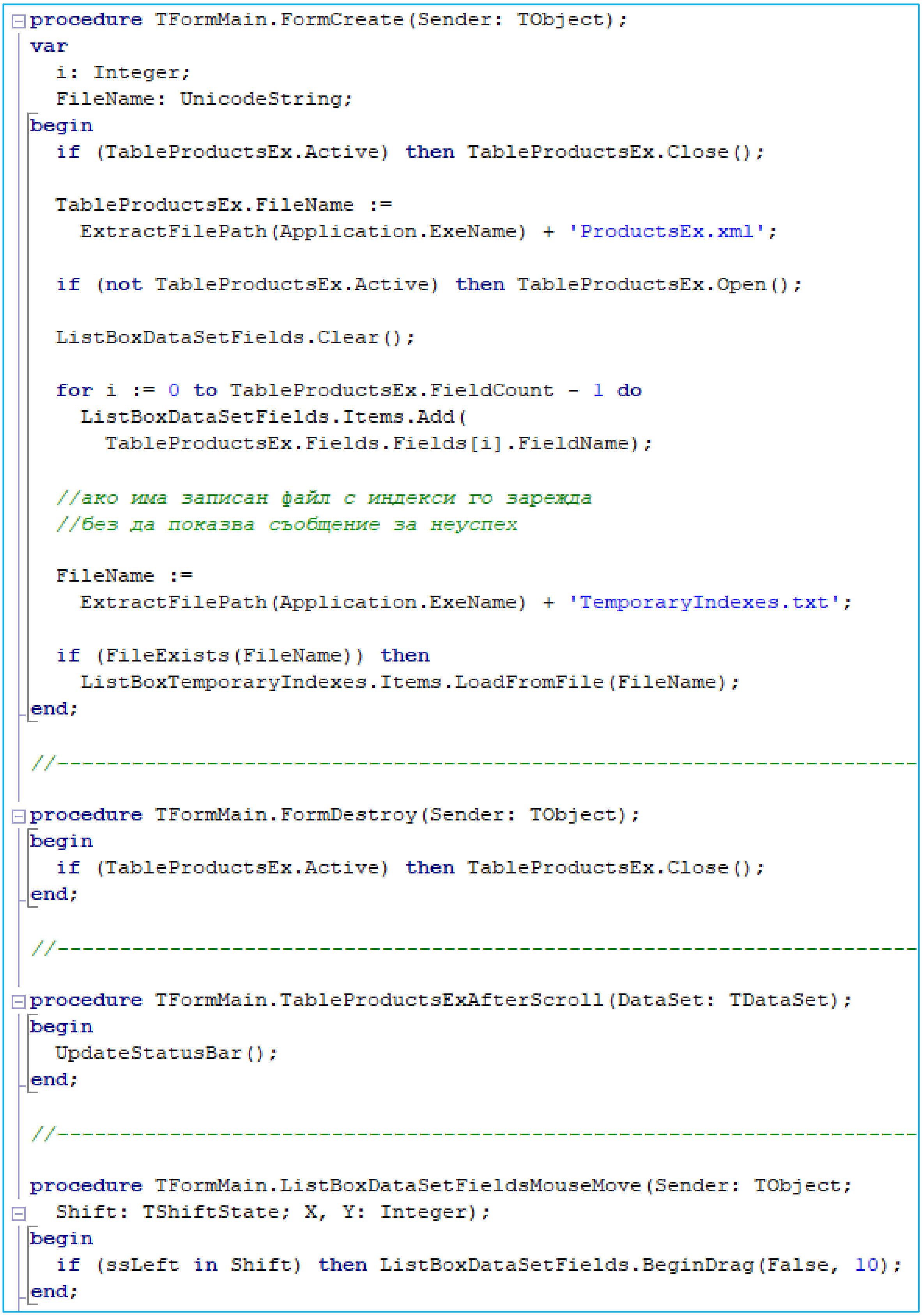


```
procedure TFormMain.UpdateStatusBar;
begin
  if (miShowRecordCount.Checked) then
  begin
    StatusBar1.Panels.Items[0].Text := ' Record Count : ' +
      IntToStr(TableProductsEx.RecordCount);
  end
  else
  begin
    StatusBar1.Panels.Items[0].Text := '';
  end;
  if (miShowProductInfo.Checked) then
  begin
    StatusBar1.Panels.Items[1].Text := ' Product : ' +
    TableProductsExProductName.AsString + ' (' +
    TableProductsExQuantityPerUnit.AsString + ') - ' +
    FloatToStrF(TableProductsExUnitPrice.AsFloat, ffCurrency, 15, 2);
  end;
  if (miShowCategoryProductInfo.Checked) then
  begin
    StatusBar1.Panels.Items[1].Text := ' Category : ' +
    TableProductsExCategoryName.AsString + ' | ' +
      ' Product : ' + TableProductsExProductName.AsString + ' (' +
    TableProductsExQuantityPerUnit.AsString + ') - ' +
    FloatToStrF(TableProductsExUnitPrice.AsFloat, ffCurrency, 15, 2);
  end;
  if (miShowSupplierCategoryProductInfo.Checked) then
  begin
    StatusBar1.Panels.Items[1].Text := ' Supplier : ' +
    TableProductsExSupplierName.AsString + ' | ' +
      ' Category : ' + TableProductsExCategoryName.AsString + ' | ' +
      ' Product : ' + TableProductsExProductName.AsString + ' (' +
    TableProductsExQuantityPerUnit.AsString + ') - ' +
    FloatToStrF(TableProductsExUnitPrice.AsFloat, ffCurrency, 15, 2);
  end;
end;
```

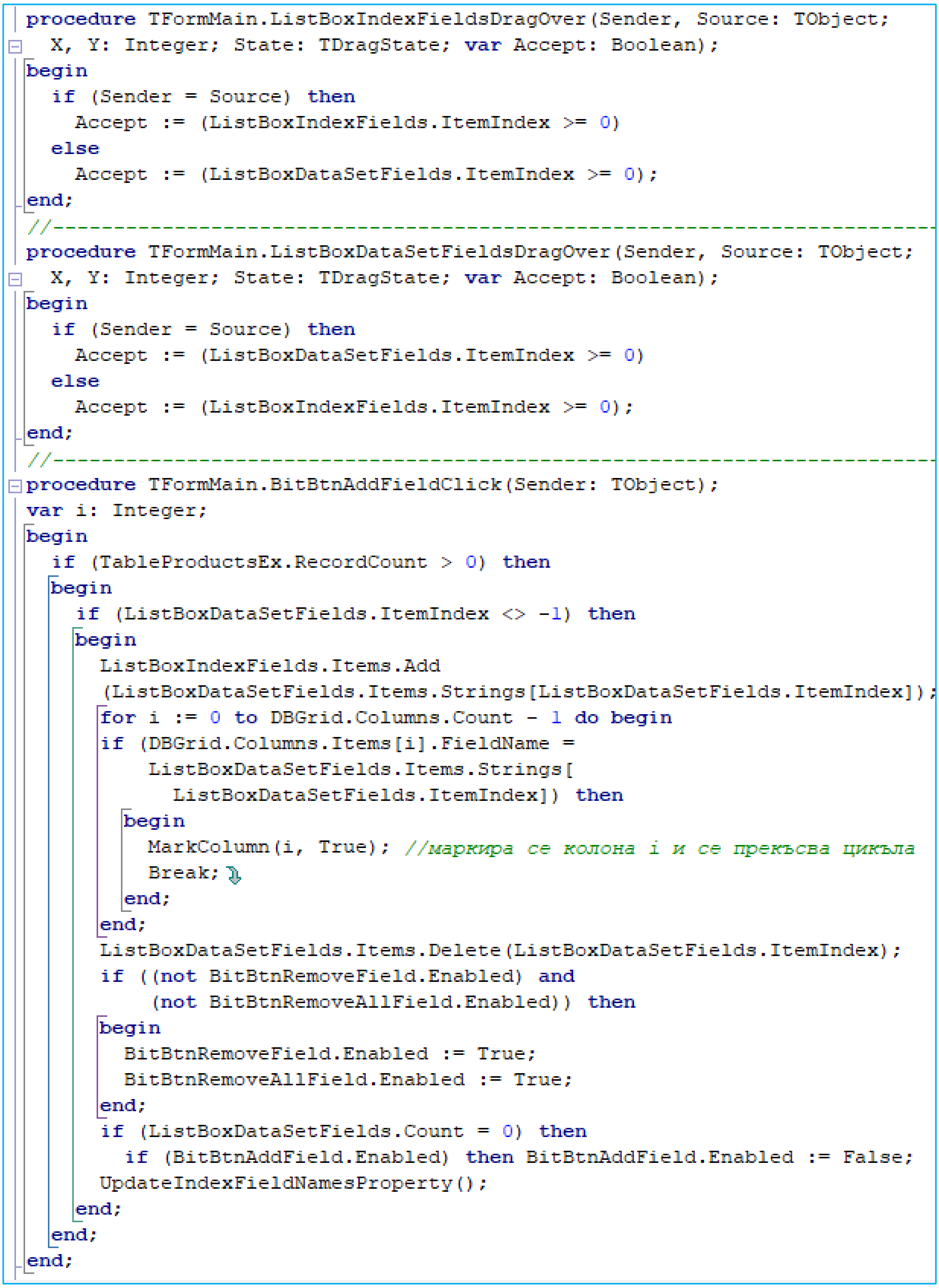


```
procedure TFormMain.MarkColumn(Index: Integer; Marked: Boolean);
begin
  if (Marked = True) then
  begin
    DBGrid.Columns.Items[Index].Color := RGB(225, 255, 245);
    DBGrid.Columns.Items[Index].Title.Color := clSilver;
    DBGrid.Columns.Items[Index].Title.Font.Style :=
      DBGrid.Columns.Items[Index].Title.Font.Style + [fsBold];
  end
  else
  begin
    DBGrid.Columns.Items[Index].Color := clWindow;
    DBGrid.Columns.Items[Index].Title.Color := clBtnFace;
    DBGrid.Columns.Items[Index].Title.Font.Style :=
      DBGrid.Columns.Items[Index].Title.Font.Style - [fsBold];
  end;
end;
```

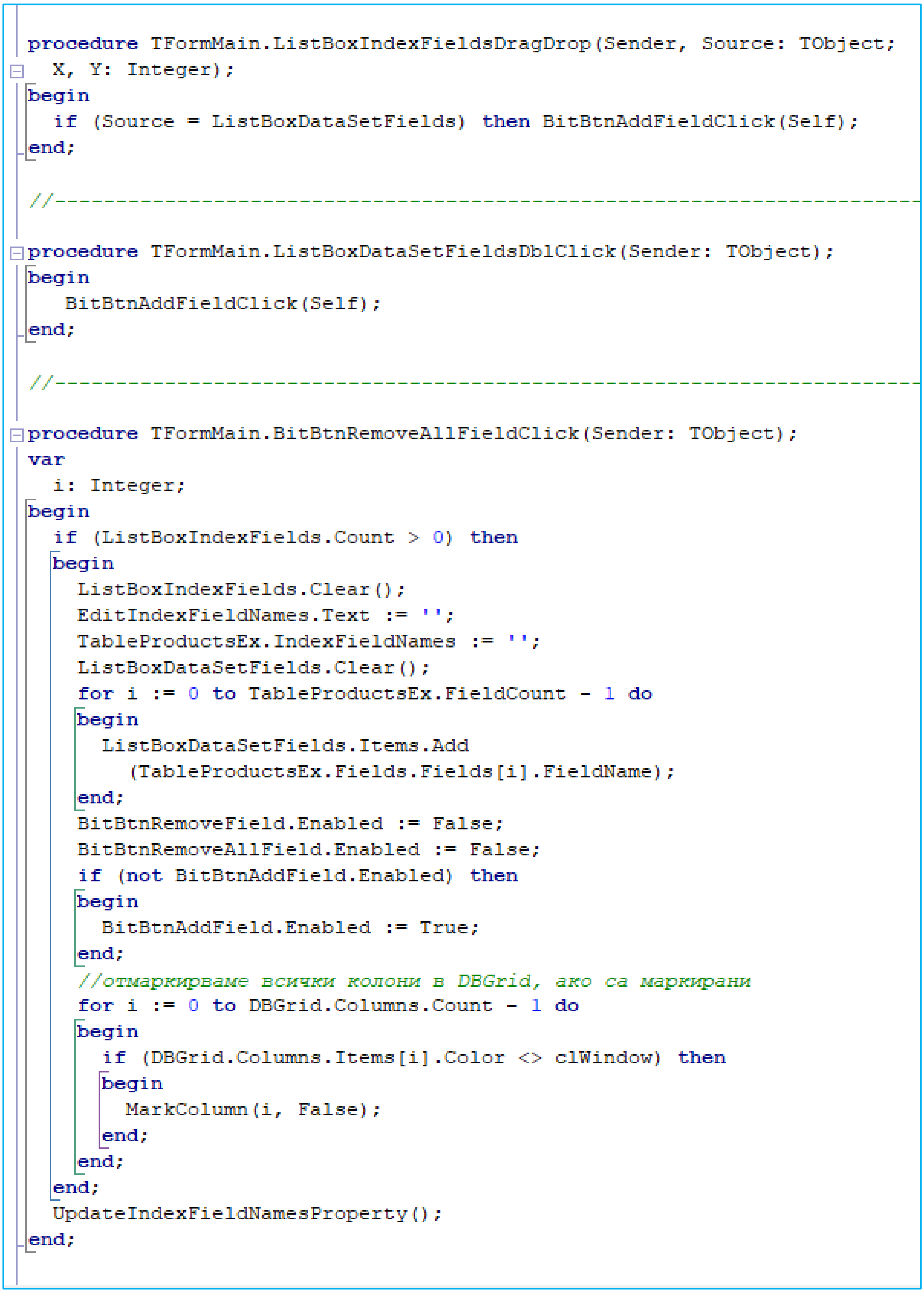
Стъпка 6: Създайте (ако все още не са) следните манипулатори (обработчици) на събития и въведете сорс-кода в редактора за код (**Code Editor**):



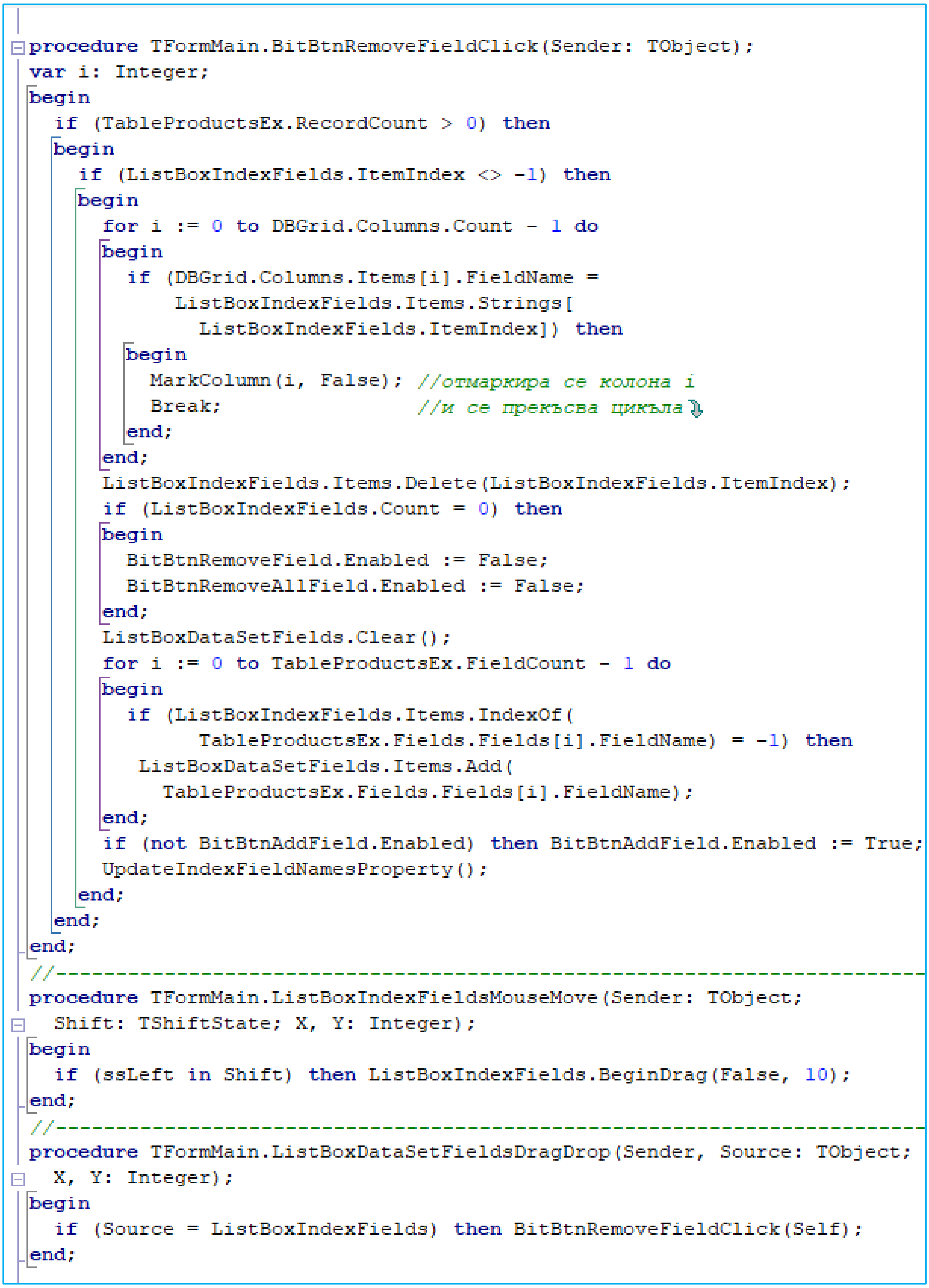
```
procedure TFormMain.FormCreate(Sender: TObject);
var
  i: Integer;
  FileName: UnicodeString;
begin
  if (TableProductsEx.Active) then TableProductsEx.Close();

  TableProductsEx.FileName :=
    ExtractFilePath(Application.ExeName) + 'ProductsEx.xml';

  if (not TableProductsEx.Active) then TableProductsEx.Open();

  ListBoxDataSetFields.Clear();

  for i := 0 to TableProductsEx.FieldCount - 1 do
    ListBoxDataSetFields.Items.Add(
      TableProductsEx.Fields.Fields[i].FieldName);

  //ако има записан файл с индекси го зарежда
  //без да показва съобщение за неуспех

  FileName :=
    ExtractFilePath(Application.ExeName) + 'TemporaryIndexes.txt';

  if (FileExists(FileName)) then
    ListBoxTemporaryIndexes.Items.LoadFromFile(FileName);
end;

//---------------------------------------------------------------------

procedure TFormMain.FormDestroy(Sender: TObject);
begin
  if (TableProductsEx.Active) then TableProductsEx.Close();
end;

//---------------------------------------------------------------------

procedure TFormMain.TableProductsExAfterScroll(DataSet: TDataSet);
begin
  UpdateStatusBar();
end;

//---------------------------------------------------------------------

procedure TFormMain.ListBoxDataSetFieldsMouseMove(Sender: TObject;
  Shift: TShiftState; X, Y: Integer);
begin
  if (ssLeft in Shift) then ListBoxDataSetFields.BeginDrag(False, 10);
end;
```



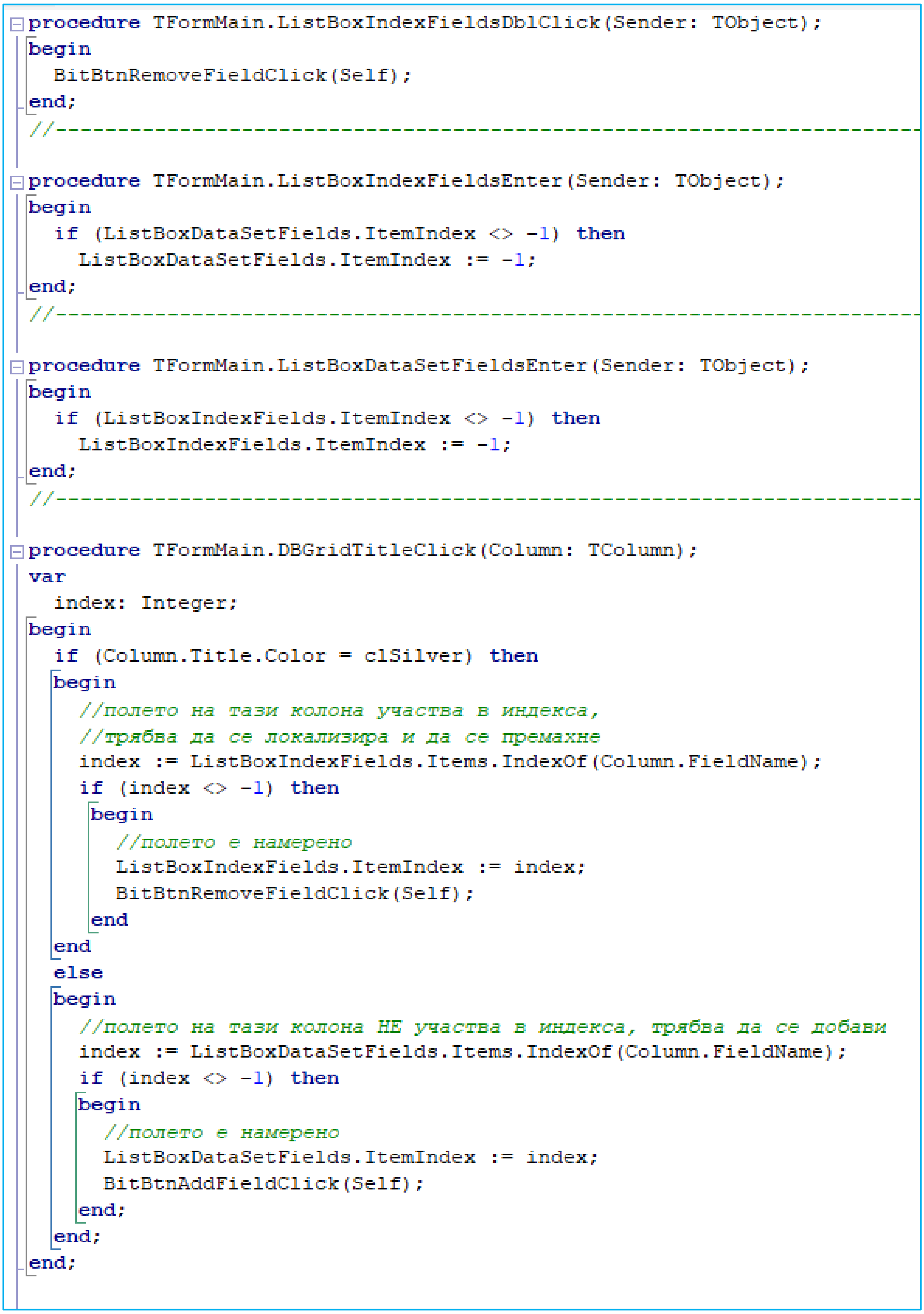
```
procedure TFormMain.ListBoxIndexFieldsDragOver(Sender, Source: TObject;
  X, Y: Integer; State: TDragState; var Accept: Boolean);
begin
  if (Sender = Source) then
    Accept := (ListBoxIndexFields.ItemIndex >= 0)
  else
    Accept := (ListBoxDataSetFields.ItemIndex >= 0);
end;
//------------------------------------------------------------------------
procedure TFormMain.ListBoxDataSetFieldsDragOver(Sender, Source: TObject;
  X, Y: Integer; State: TDragState; var Accept: Boolean);
begin
  if (Sender = Source) then
    Accept := (ListBoxDataSetFields.ItemIndex >= 0)
  else
    Accept := (ListBoxIndexFields.ItemIndex >= 0);
end;
//------------------------------------------------------------------------
procedure TFormMain.BitBtnAddFieldClick(Sender: TObject);
var i: Integer;
begin
  if (TableProductsEx.RecordCount > 0) then
  begin
    if (ListBoxDataSetFields.ItemIndex <> -1) then
    begin
      ListBoxIndexFields.Items.Add
      (ListBoxDataSetFields.Items.Strings[ListBoxDataSetFields.ItemIndex]);
      for i := 0 to DBGrid.Columns.Count - 1 do begin
      if (DBGrid.Columns.Items[i].FieldName =
          ListBoxDataSetFields.Items.Strings[
            ListBoxDataSetFields.ItemIndex]) then
        begin
          MarkColumn(i, True); //маркира се колона i и се прекъсва цикъла
          Break;
        end;
      end;
      ListBoxDataSetFields.Items.Delete(ListBoxDataSetFields.ItemIndex);
      if ((not BitBtnRemoveField.Enabled) and
          (not BitBtnRemoveAllField.Enabled)) then
      begin
        BitBtnRemoveField.Enabled := True;
        BitBtnRemoveAllField.Enabled := True;
      end;
      if (ListBoxDataSetFields.Count = 0) then
        if (BitBtnAddField.Enabled) then BitBtnAddField.Enabled := False;
      UpdateIndexFieldNamesProperty();
    end;
  end;
end;
```



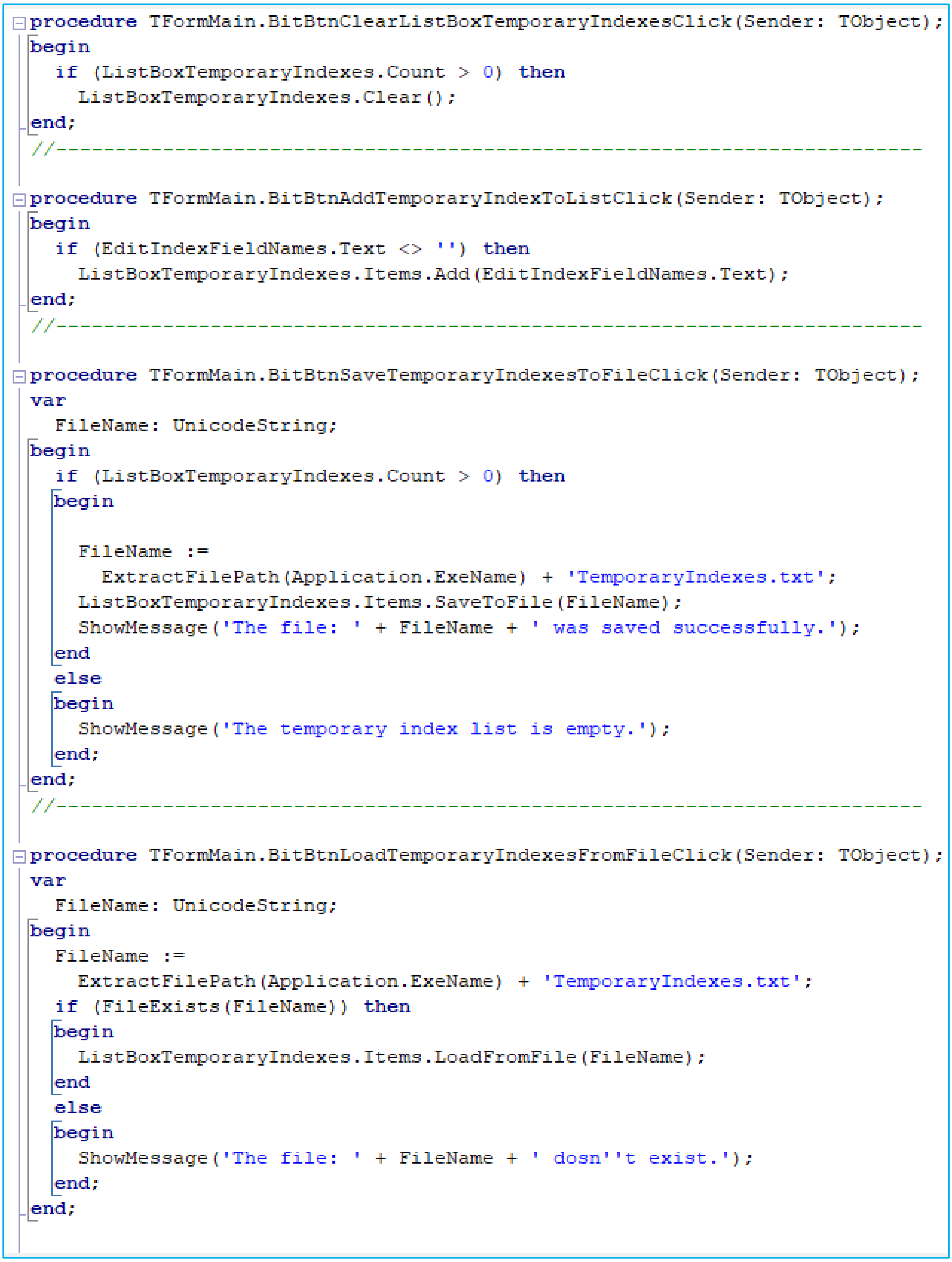
```
procedure TFormMain.ListBoxIndexFieldsDragDrop(Sender, Source: TObject;
  X, Y: Integer);
begin
  if (Source = ListBoxDataSetFields) then BitBtnAddFieldClick(Self);
end;

//----------------------------------------------------------------------

procedure TFormMain.ListBoxDataSetFieldsDblClick(Sender: TObject);
begin
   BitBtnAddFieldClick(Self);
end;

//----------------------------------------------------------------------

procedure TFormMain.BitBtnRemoveAllFieldClick(Sender: TObject);
var
  i: Integer;
begin
  if (ListBoxIndexFields.Count > 0) then
  begin
    ListBoxIndexFields.Clear();
    EditIndexFieldNames.Text := '';
    TableProductsEx.IndexFieldNames := '';
    ListBoxDataSetFields.Clear();
    for i := 0 to TableProductsEx.FieldCount - 1 do
    begin
      ListBoxDataSetFields.Items.Add
        (TableProductsEx.Fields.Fields[i].FieldName);
    end;
    BitBtnRemoveField.Enabled := False;
    BitBtnRemoveAllField.Enabled := False;
    if (not BitBtnAddField.Enabled) then
    begin
      BitBtnAddField.Enabled := True;
    end;
    //отмаркирваме всички колони в DBGrid, ако са маркирани
    for i := 0 to DBGrid.Columns.Count - 1 do
    begin
      if (DBGrid.Columns.Items[i].Color <> clWindow) then
      begin
        MarkColumn(i, False);
      end;
    end;
  end;
  UpdateIndexFieldNamesProperty();
end;
```



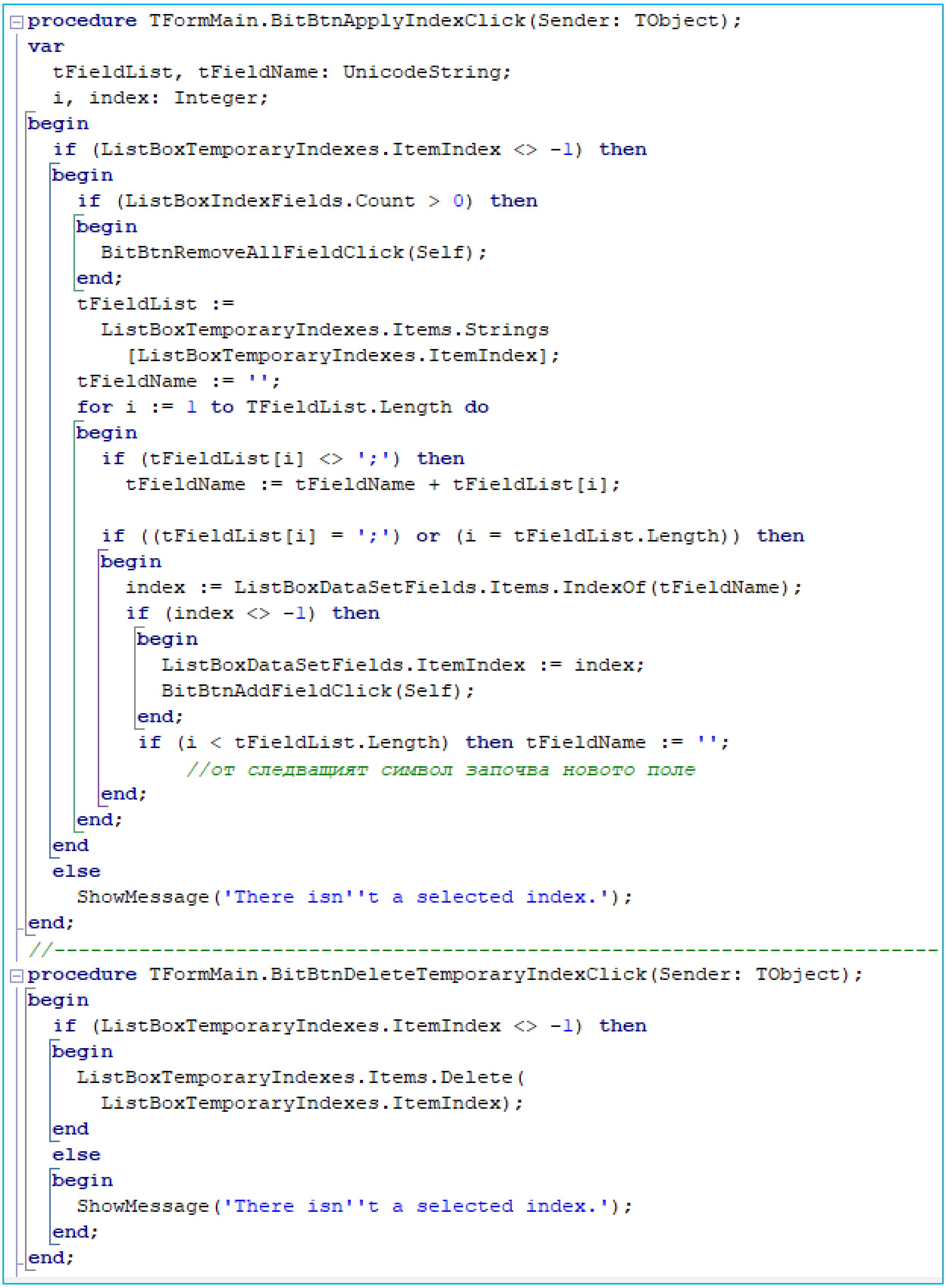
```
procedure TFormMain.BitBtnRemoveFieldClick(Sender: TObject);
var i: Integer;
begin
  if (TableProductsEx.RecordCount > 0) then
  begin
    if (ListBoxIndexFields.ItemIndex <> -1) then
    begin
      for i := 0 to DBGrid.Columns.Count - 1 do
      begin
        if (DBGrid.Columns.Items[i].FieldName =
            ListBoxIndexFields.Items.Strings[
              ListBoxIndexFields.ItemIndex]) then
        begin
          MarkColumn(i, False); //отмаркира се колона i
          Break;                //и се прекъсва цикъла
        end;
      end;
      ListBoxIndexFields.Items.Delete(ListBoxIndexFields.ItemIndex);
      if (ListBoxIndexFields.Count = 0) then
      begin
        BitBtnRemoveField.Enabled := False;
        BitBtnRemoveAllField.Enabled := False;
      end;
      ListBoxDataSetFields.Clear();
      for i := 0 to TableProductsEx.FieldCount - 1 do
      begin
        if (ListBoxIndexFields.Items.IndexOf(
              TableProductsEx.Fields.Fields[i].FieldName) = -1) then
         ListBoxDataSetFields.Items.Add(
           TableProductsEx.Fields.Fields[i].FieldName);
      end;
      if (not BitBtnAddField.Enabled) then BitBtnAddField.Enabled := True;
      UpdateIndexFieldNamesProperty();
    end;
  end;
end;
//--------------------------------------------------------------------------
procedure TFormMain.ListBoxIndexFieldsMouseMove(Sender: TObject;
  Shift: TShiftState; X, Y: Integer);
begin
  if (ssLeft in Shift) then ListBoxIndexFields.BeginDrag(False, 10);
end;
//--------------------------------------------------------------------------
procedure TFormMain.ListBoxDataSetFieldsDragDrop(Sender, Source: TObject;
  X, Y: Integer);
begin
  if (Source = ListBoxIndexFields) then BitBtnRemoveFieldClick(Self);
end;
```



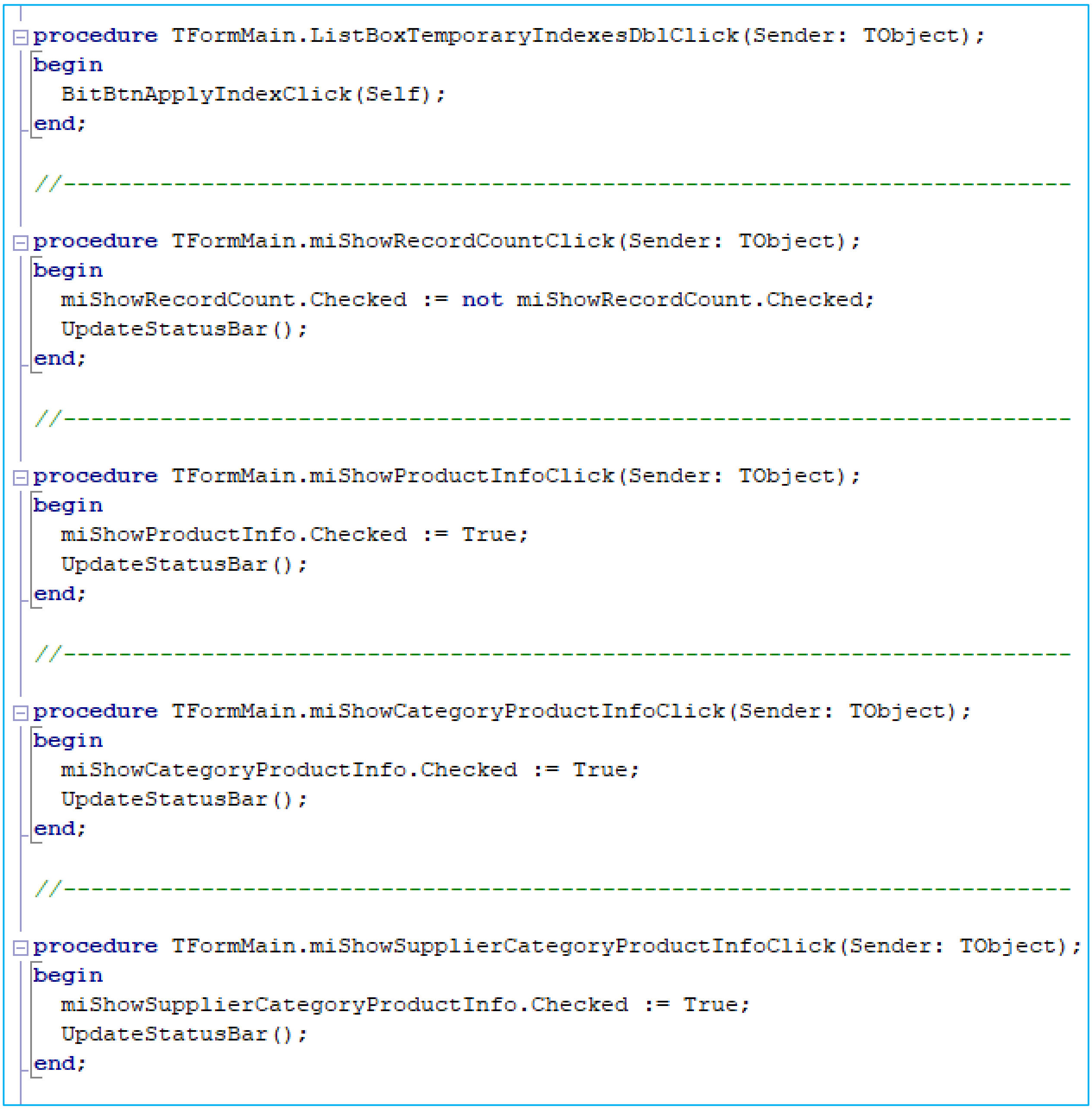
```
procedure TFormMain.ListBoxIndexFieldsDblClick(Sender: TObject);
begin
  BitBtnRemoveFieldClick(Self);
end;
//----------------------------------------------------------------------

procedure TFormMain.ListBoxIndexFieldsEnter(Sender: TObject);
begin
  if (ListBoxDataSetFields.ItemIndex <> -1) then
    ListBoxDataSetFields.ItemIndex := -1;
end;
//----------------------------------------------------------------------

procedure TFormMain.ListBoxDataSetFieldsEnter(Sender: TObject);
begin
  if (ListBoxIndexFields.ItemIndex <> -1) then
    ListBoxIndexFields.ItemIndex := -1;
end;
//----------------------------------------------------------------------

procedure TFormMain.DBGridTitleClick(Column: TColumn);
var
  index: Integer;
begin
  if (Column.Title.Color = clSilver) then
  begin
    //полето на тази колона участва в индекса,
    //трябва да се локализира и да се премахне
    index := ListBoxIndexFields.Items.IndexOf(Column.FieldName);
    if (index <> -1) then
    begin
      //полето е намерено
      ListBoxIndexFields.ItemIndex := index;
      BitBtnRemoveFieldClick(Self);
    end
  end
  else
  begin
    //полето на тази колона НЕ участва в индекса, трябва да се добави
    index := ListBoxDataSetFields.Items.IndexOf(Column.FieldName);
    if (index <> -1) then
    begin
      //полето е намерено
      ListBoxDataSetFields.ItemIndex := index;
      BitBtnAddFieldClick(Self);
    end;
  end;
end;
```



```
procedure TFormMain.BitBtnClearListBoxTemporaryIndexesClick(Sender: TObject);
begin
  if (ListBoxTemporaryIndexes.Count > 0) then
    ListBoxTemporaryIndexes.Clear();
end;
//-------------------------------------------------------------------------

procedure TFormMain.BitBtnAddTemporaryIndexToListClick(Sender: TObject);
begin
  if (EditIndexFieldNames.Text <> '') then
    ListBoxTemporaryIndexes.Items.Add(EditIndexFieldNames.Text);
end;
//-------------------------------------------------------------------------

procedure TFormMain.BitBtnSaveTemporaryIndexesToFileClick(Sender: TObject);
var
  FileName: UnicodeString;
begin
  if (ListBoxTemporaryIndexes.Count > 0) then
  begin

    FileName :=
      ExtractFilePath(Application.ExeName) + 'TemporaryIndexes.txt';
    ListBoxTemporaryIndexes.Items.SaveToFile(FileName);
    ShowMessage('The file: ' + FileName + ' was saved successfully.');
  end
  else
  begin
    ShowMessage('The temporary index list is empty.');
  end;
end;
//-------------------------------------------------------------------------

procedure TFormMain.BitBtnLoadTemporaryIndexesFromFileClick(Sender: TObject);
var
  FileName: UnicodeString;
begin
  FileName :=
    ExtractFilePath(Application.ExeName) + 'TemporaryIndexes.txt';
  if (FileExists(FileName)) then
  begin
    ListBoxTemporaryIndexes.Items.LoadFromFile(FileName);
  end
  else
  begin
    ShowMessage('The file: ' + FileName + ' dosn''t exist.');
  end;
end;
```



```
procedure TFormMain.BitBtnApplyIndexClick(Sender: TObject);
var
  tFieldList, tFieldName: UnicodeString;
  i, index: Integer;
begin
  if (ListBoxTemporaryIndexes.ItemIndex <> -1) then
  begin
    if (ListBoxIndexFields.Count > 0) then
    begin
      BitBtnRemoveAllFieldClick(Self);
    end;
    tFieldList :=
      ListBoxTemporaryIndexes.Items.Strings
        [ListBoxTemporaryIndexes.ItemIndex];
    tFieldName := '';
    for i := 1 to TFieldList.Length do
    begin
      if (tFieldList[i] <> ';') then
        tFieldName := tFieldName + tFieldList[i];

      if ((tFieldList[i] = ';') or (i = tFieldList.Length)) then
      begin
        index := ListBoxDataSetFields.Items.IndexOf(tFieldName);
        if (index <> -1) then
         begin
           ListBoxDataSetFields.ItemIndex := index;
           BitBtnAddFieldClick(Self);
         end;
         if (i < tFieldList.Length) then tFieldName := '';
             //от следващият символ започва новото поле
      end;
    end;
  end
  else
    ShowMessage('There isn''t a selected index.');
end;
//-------------------------------------------------------------------------
procedure TFormMain.BitBtnDeleteTemporaryIndexClick(Sender: TObject);
begin
  if (ListBoxTemporaryIndexes.ItemIndex <> -1) then
  begin
    ListBoxTemporaryIndexes.Items.Delete(
      ListBoxTemporaryIndexes.ItemIndex);
  end
  else
  begin
    ShowMessage('There isn''t a selected index.');
  end;
end;
```



```
procedure TFormMain.ListBoxTemporaryIndexesDblClick(Sender: TObject);
begin
  BitBtnApplyIndexClick(Self);
end;

//-------------------------------------------------------------------------

procedure TFormMain.miShowRecordCountClick(Sender: TObject);
begin
  miShowRecordCount.Checked := not miShowRecordCount.Checked;
  UpdateStatusBar();
end;

//-------------------------------------------------------------------------

procedure TFormMain.miShowProductInfoClick(Sender: TObject);
begin
  miShowProductInfo.Checked := True;
  UpdateStatusBar();
end;

//-------------------------------------------------------------------------

procedure TFormMain.miShowCategoryProductInfoClick(Sender: TObject);
begin
  miShowCategoryProductInfo.Checked := True;
  UpdateStatusBar();
end;

//-------------------------------------------------------------------------

procedure TFormMain.miShowSupplierCategoryProductInfoClick(Sender: TObject);
begin
  miShowSupplierCategoryProductInfo.Checked := True;
  UpdateStatusBar();
end;
```