

9

Разработване на приложение ModifyingDataExample (имплементиране)

ИМПЛЕМЕНТИРАНЕ НА ПРИЛОЖЕНИЕТО ModifyingDataExample

Стартирайте RAD Studio и изпълнете следните стъпки:

Стъпка 1: От главното меню изберете **File ► Open Project...** и изберете проектния файл "**ModifyingDataExample.dproj**" от архива ModifyingDataExampleDFM.zip.

Стъпка 2: В декларацията на класа TMainForm добавете в частта **private** 6 метода и една променлива с име Semafor от тип Integer (възможно е тези декларации вече да са добавени интерфейсната част на модула), както е показано по-долу:
(Ако са добавени декларациите, пропуснете тази стъпката.)

```
type
  TFormMain = class(TForm)
    TableEmployees: TClientDataSet;
    //.....
    procedure BitBtnAppendRecordClick(Sender: TObject);
    procedure BitBtnInsertRecordClick(Sender: TObject);
    procedure BitBtnSetFieldsClick(Sender: TObject);
    procedure BitBtnIncDecEmpSalaryClick(Sender: TObject);
    //.....
    procedure BitBtnMoreLessClick(Sender: TObject);
  private
    { Private declarations }
    Semafor: Integer;
    procedure UpdateStatusLine();
    procedure RegisterEvent(EventIndex: Integer);
    procedure DBControlsBtnFaceColor();
    procedure DBControlsWindowColor();
    procedure DBControlsReadOnlyTrue();
    procedure DBControlsReadOnlyFalse();
  public
    { Public declarations }
  end;
```

Стъпка 3: В частта **implementation** на модула MainUnit, добавете реализация на декларираните в интерфейсната част 6 метода, както е показано по-долу:

```
procedure TFormMain.UpdateStatusLine;
var
  S: AnsiString;
begin
  if (not TableEmployees.Active) then Exit;
  S := ' Record Count := ' + IntToStr(TableEmployees.RecordCount);
  if (StatusBar.Panels.Items[0].Text <> S) then
    StatusBar.Panels.Items[0].Text := S;

  case TableEmployees.State of

    dsInsert:
    begin
      S := ' Status := dsInsert';
      if (StatusBar.Panels.Items[1].Text <> S) then
        StatusBar.Panels.Items[1].Text := S;
      end;

    dsEdit:
    begin
      S := ' Status := dsEdit';
      if (StatusBar.Panels.Items[1].Text <> S) then
        StatusBar.Panels.Items[1].Text := S;
      end;

    dsBrowse:
    begin
      S := ' Status := dsBrowse';
      if (StatusBar.Panels.Items[1].Text <> S) then
        StatusBar.Panels.Items[1].Text := S;
      end

    else
    begin
      S := ' Status := Other State';
      if (StatusBar.Panels.Items[1].Text <> S) then
        StatusBar.Panels.Items[1].Text := S;
      end;
    end; //case

  S := ' Employee: ' +
    TableEmployeesTitle.AsString + ' ' +
    TableEmployeesFirstName.AsString + ' ' +
    TableEmployeesLastName.AsString + ' - Salary: ' +
    FloatToStrF(TableEmployeesSalary.AsFloat, ffCurrency, 15, 0);
  if (StatusBar.Panels.Items[2].Text <> S) then
    StatusBar.Panels.Items[2].Text := S;
end;
```

```

procedure TFormMain.RegisterEvent(EventIndex: Integer);
begin
    if (CheckListBoxEvents.Checked[EventIndex] = True) then
    begin
        Semafor := Semafor + 1; //Inc(Semafor);
        ListBoxTriggeredEventList.Items.Insert(0, IntToStr(Semafor) +
            ' - ' + CheckListBoxEvents.Items.Strings[EventIndex]);
    end;
    UpdateStatusLine();
end;
//-----

```

```

//-----
procedure TFormMain.DBControlsBtnFaceColor;
begin
    DBEditPosition.Color := clBtnFace;
    DBEditAddress.Color := clBtnFace;
    DBEditCountry.Color := clBtnFace;
    DBEditRegion.Color := clBtnFace;
    DBEditCity.Color := clBtnFace;
    DBEditPostalCode.Color := clBtnFace;
    DBEditBirthDate.Color := clBtnFace;
    DBEditHomePhone.Color := clBtnFace;
    DBEditHireDate.Color := clBtnFace;
    DBEditSalary.Color := clBtnFace;
    DBEditExtension.Color := clBtnFace;
    DBMemoNotes.Color := clBtnFace;
    DBEditTitle.Color := clBtnFace;
    DBEditFirstName.Color := clBtnFace;
    DBEditLastName.Color := clBtnFace;
end;
//-----
procedure TFormMain.DBControlsWindowColor;
begin
    DBEditPosition.Color := clWindow;
    DBEditAddress.Color := clWindow;
    DBEditCountry.Color := clWindow;
    DBEditRegion.Color := clWindow;
    DBEditCity.Color := clWindow;
    DBEditPostalCode.Color := clWindow;
    DBEditBirthDate.Color := clWindow;
    DBEditHomePhone.Color := clWindow;
    DBEditHireDate.Color := clWindow;
    DBEditSalary.Color := clWindow;
    DBEditExtension.Color := clWindow;
    DBMemoNotes.Color := clWindow;
    DBEditTitle.Color := clWindow;
    DBEditFirstName.Color := clWindow;
    DBEditLastName.Color := clWindow;
end;

```

```

procedure TFormMain.DBControlsReadOnlyTrue;
begin
    DBEditPosition.ReadOnly := True;
    DBEditAddress.ReadOnly := True;
    DBEditCountry.ReadOnly := True;
    DBEditRegion.ReadOnly := True;
    DBEditCity.ReadOnly := True;
    DBEditPostalCode.ReadOnly := True;
    DBEditBirthDate.ReadOnly := True;
    DBEditHomePhone.ReadOnly := True;
    DBEditHireDate.ReadOnly := True;
    DBEditSalary.ReadOnly := True;
    DBEditExtension.ReadOnly := True;
    DBMemoNotes.ReadOnly := True;
    DBImagePhoto.ReadOnly := True;
    DBEditTitle.ReadOnly := True;
    DBEditFirstName.ReadOnly := True;
    DBEditLastName.ReadOnly := True;
end;
//-----
procedure TFormMain.DBControlsReadOnlyFalse;
begin
    DBEditPosition.ReadOnly := False;
    DBEditAddress.ReadOnly := False;
    DBEditCountry.ReadOnly := False;
    DBEditRegion.ReadOnly := False;
    DBEditCity.ReadOnly := False;
    DBEditPostalCode.ReadOnly := False;
    DBEditBirthDate.ReadOnly := False;
    DBEditHomePhone.ReadOnly := False;
    DBEditHireDate.ReadOnly := False;
    DBEditSalary.ReadOnly := False;
    DBEditExtension.ReadOnly := False;
    DBMemoNotes.ReadOnly := False;
    DBImagePhoto.ReadOnly := False;
    DBEditTitle.ReadOnly := False;
    DBEditFirstName.ReadOnly := False;
    DBEditLastName.ReadOnly := False;
end;

```

Стъпка 4: Създайте следните манипулатори (обработчици) на събития (ако вече не са създадени) и въведете сорс-кода в редактора за код (**Code Editor**):

```

procedure TFormMain.FormCreate(Sender: TObject);
begin
    if (TableEmployees.Active) then TableEmployees.Close();
    TableEmployees.FileName :=
        ExtractFilePath(Application.ExeName) + 'EmployeesEx.xml';
    if (not TableEmployees.Active) then TableEmployees.Open();
    OpenPictureDialog.InitialDir := ExtractFilePath(Application.ExeName);
    TableEmployees.LogChanges := False;
    TableEmployees.MergeChangeLog();
end;

```

```

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

//-----

procedure TFormMain.miLoadPhotoFromFileClick(Sender: TObject);
begin
    if (OpenPictureDialog.Execute()) then
    begin
        if ((TableEmployees.State <> dsInsert) or
            (TableEmployees.State <> dsEdit)) then TableEmployees.Edit();
        DBImagePhoto.Picture.LoadFromFile(OpenPictureDialog.FileName);
        if (OpenPictureDialog.InitialDir <>
            ExtractFilePath(OpenPictureDialog.FileName)) then
        begin
            OpenPictureDialog.InitialDir :=
                ExtractFilePath(OpenPictureDialog.FileName);
        end;
    end;
end;

//-----

procedure TFormMain.BitBtnAppendClick(Sender: TObject);
begin
    TableEmployees.Append();
end;

//-----

procedure TFormMain.CheckBoxDataSourceAutoEditClick(Sender: TObject);
begin
    DataSourceEmployees.AutoEdit := CheckBoxDataSourceAutoEdit.Checked;
    if (CheckBoxDataSourceAutoEdit.Checked = True) then
    begin
        CheckBoxDataSourceAutoEdit.Color := clLime; //RGB(240,255,240);
    end
    else
    begin
        CheckBoxDataSourceAutoEdit.Color := clBtnFace;
    end
end;

//-----

```

```
procedure TFormMain.CheckBoxEditInformationInFormClick(Sender: TObject);
begin
    if ((TableEmployees.State = dsInsert) or
        (TableEmployees.State = dsEdit)) then TableEmployees.Cancel();
    CheckBoxDataSourceAutoEdit.Enabled :=
        not CheckBoxEditInformationInForm.Checked;
    if (CheckBoxEditInformationInForm.Checked = True) then
    begin
        DBControlsBtnFaceColor();
        DBControlsReadOnlyTrue();
        BitBtnInsert.Enabled := False;
        BitBtnAppend.Enabled := False;
        BitBtnEdit.Enabled := False;
        BitBtnPost.Enabled := False;
        BitBtnCancel.Enabled := False;
        CheckBoxEditInformationInForm.Color := clAqua; //RGB(243,232,255);
    end
    else // CheckBoxEditEmployeeInformationInForm.Checked = False
    begin
        DBControlsWindowColor();
        DBControlsReadOnlyFalse();
        BitBtnInsert.Enabled := True;
        BitBtnAppend.Enabled := True;
        BitBtnEdit.Enabled := True;
        BitBtnPost.Enabled := False;
        BitBtnCancel.Enabled := False;
        CheckBoxEditInformationInForm.Color := clBtnFace;
    end;
    BitBtnAppendRecord.Enabled := CheckBoxEditInformationInForm.Checked;
    BitBtnInsertRecord.Enabled := CheckBoxEditInformationInForm.Checked;
    BitBtnSetFields.Enabled := CheckBoxEditInformationInForm.Checked;
end;

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

procedure TFormMain.TableEmployeesAfterCancel(DataSet: TDataSet);
begin
    RegisterEvent(0); // 0 - On After Cancel
end;

//-----

procedure TFormMain.TableEmployeesAfterClose(DataSet: TDataSet);
begin
    RegisterEvent(1); // 1 - On After Close
end;
```

```
procedure TFormMain.TableEmployeesAfterDelete(DataSet: TDataSet);
begin
    RegisterEvent(2); // 2 - On After Delete
end;

//-----

procedure TFormMain.TableEmployeesAfterEdit(DataSet: TDataSet);
begin
    RegisterEvent(3); // 3 - On After Edit
end;

//-----

procedure TFormMain.TableEmployeesAfterInsert(DataSet: TDataSet);
begin
    RegisterEvent(4); // 4 - On After Insert
    DBEditPosition.SetFocus();
end;

//-----

procedure TFormMain.TableEmployeesAfterOpen(DataSet: TDataSet);
begin
    RegisterEvent(5); // 5 - On After Open
end;

//-----

procedure TFormMain.TableEmployeesAfterPost(DataSet: TDataSet);
begin
    RegisterEvent(6); // 6 - On After Post
end;

//-----

procedure TFormMain.TableEmployeesAfterRefresh(DataSet: TDataSet);
begin
    RegisterEvent(7); // 7 - On After Refresh
end;

//-----

procedure TFormMain.TableEmployeesAfterScroll(DataSet: TDataSet);
begin
    RegisterEvent(8); // 8 - On After Scroll
end;
```



```

procedure TFormMain.TableEmployeesBeforeCancel(DataSet: TDataSet);
begin
    RegisterEvent(9); // 9 - On Before Cancel
end;

//-----

procedure TFormMain.TableEmployeesBeforeClose(DataSet: TDataSet);
begin
    RegisterEvent(10); // 10 - On Before Close
end;

//-----

procedure TFormMain.TableEmployeesBeforeDelete(DataSet: TDataSet);
begin
    RegisterEvent(11); // 11 - On Before Delete
    if (MessageDlg('Delete This Record?',
        TMsgDlgType.mtConfirmation, mbYesNo, 0) <> mrYes) then Abort();
end;

//-----

procedure TFormMain.TableEmployeesBeforeEdit(DataSet: TDataSet);
begin
    RegisterEvent(12); // 12 - On Before Edit
end;

//-----

procedure TFormMain.TableEmployeesBeforeInsert(DataSet: TDataSet);
begin
    RegisterEvent(13); // 13 - On Before Insert
end;

//-----

procedure TFormMain.TableEmployeesBeforeOpen(DataSet: TDataSet);
begin
    RegisterEvent(14); // 14 - On Before Open
end;

//-----

procedure TFormMain.TableEmployeesBeforePost(DataSet: TDataSet);
begin
    RegisterEvent(15); // 15 - On Before Post
end;

```



```
//-----  
procedure TFormMain.TableEmployeesBeforeRefresh(DataSet: TDataSet);  
begin  
    RegisterEvent(16); // 16 - On Before Refresh  
end;  
  
//-----  
procedure TFormMain.TableEmployeesBeforeScroll(DataSet: TDataSet);  
begin  
    RegisterEvent(17); // 17 - On Before Scroll  
end;  
  
//-----  
procedure TFormMain.TableEmployeesNewRecord(DataSet: TDataSet);  
begin  
    RegisterEvent(18); // 18 - On New Record  
    TableEmployeesHireDate.AsDateTime := Date();  
end;  
  
//-----  
//-----  
procedure TFormMain.BitBtnClearTriggeredEventListClick(Sender: TObject);  
begin  
    if (ListBoxTriggeredEventList.Count > 0) then  
    begin  
        ListBoxTriggeredEventList.Clear();  
    end  
end;  
  
//-----  
procedure TFormMain.BitBtnInvertSelectionClick(Sender: TObject);  
var  
    i: Integer;  
begin  
    for i := 0 to CheckListBoxEvents.Count - 1 do  
    begin  
        CheckListBoxEvents.Checked[i] := not CheckListBoxEvents.Checked[i];  
    end;  
end;  
  
//-----
```

```
//-----  
procedure TFormMain.BitBtnSelectAllClick(Sender: TObject);  
var  
    i: Integer;  
begin  
    for i := 0 to CheckListBoxEvents.Count - 1 do  
    begin  
        if (not CheckListBoxEvents.Checked[i]) then  
        begin  
            CheckListBoxEvents.Checked[i] := True;  
        end;  
    end;  
end;  
  
//-----  
  
procedure TFormMain.miDeletePhotoClick(Sender: TObject);  
begin  
    if (MessageDlg('Delete Photo?',  
        mtConfirmation, mbYesNo, 0) <> mrYes) then Exit; ↴  
    if ((TableEmployees.State <> dsInsert) or  
        (TableEmployees.State <> dsEdit)) then  
    begin  
        TableEmployees.Edit();  
    end;  
    DBImagePhoto.Picture.Bitmap := nil;  
end;  
  
//-----  
  
procedure TFormMain.DBImagePhotoDblClick(Sender: TObject);  
begin  
    miLoadPhotoFromFileClick(Self);  
end;  
  
//-----
```

```
procedure TFormMain.BitBtnAppendRecordClick(Sender: TObject);

var
    f: TAddEditEmployeeForm;

begin
    f := TAddEditEmployeeForm.Create(Self);
    f.Caption := 'Employee Append Record';
    f.OpenPictureDialog.InitialDir := OpenPictureDialog.InitialDir;

    if (f.ShowModal() = mrOk) then
    begin
        TableEmployees.AppendRecord([
            nil, //ID
            f.EditTitle.Text, //Title
            f.EditLastName.Text, //LastName
            f.EditFirstName.Text, //FirstName
            f.EditPosition.Text, //Position
            f.MaskEditBirthDate.Text, //BirthDate
            f.MaskEditHireDate.Text, //HireDate
            f.EditAddress.Text, //Address
            f.EditCity.Text, //City
            f.EditRegion.Text, //Region
            f.EditPostalCode.Text, //PostalCode
            f.EditCountry.Text, //Country
            f.EditHomePhone.Text, //HomePhone
            f.EditExtension.Text, //Extension
            f.ImagePhoto.Picture.Bitmap, //Photo
            f.MemoNotes.Text, //Notes
            f.EditSalary.Text, //Salary
        ]);

        TableEmployees.Post();
    end;

    if (OpenPictureDialog.InitialDir <>
        f.OpenPictureDialog.InitialDir) then
    begin
        OpenPictureDialog.InitialDir := f.OpenPictureDialog.InitialDir;
    end;

    f.Free;
end;
```

```
procedure TFormMain.BitBtnInsertRecordClick(Sender: TObject);

var
    f: TAddEditEmployeeForm;

begin
    f := TAddEditEmployeeForm.Create(Self);
    f.Caption := 'Employee Insert Record';
    f.OpenPictureDialog.InitialDir := OpenPictureDialog.InitialDir;

    if (f.ShowModal() = mrOk) then
    begin
        TableEmployees.InsertRecord([
            nil, //ID
            f.EditTitle.Text, //Title
            f.EditLastName.Text, //LastName
            f.EditFirstName.Text, //FirstName
            f.EditPosition.Text, //Position
            f.MaskEditBirthDate.Text, //BirthDate
            f.MaskEditHireDate.Text, //HireDate
            f.EditAddress.Text, //Address
            f.EditCity.Text, //City
            f.EditRegion.Text, //Region
            f.EditPostalCode.Text, //PostalCode
            f.EditCountry.Text, //Country
            f.EditHomePhone.Text, //HomePhone
            f.EditExtension.Text, //Extension
            f.ImagePhoto.Picture.Bitmap, //Photo
            f.MemoNotes.Text, //Notes
            f.EditSalary.Text, //Salary
        ]);

        TableEmployees.Post();
    end;

    if (OpenPictureDialog.InitialDir <>
        f.OpenPictureDialog.InitialDir) then
    begin
        OpenPictureDialog.InitialDir := f.OpenPictureDialog.InitialDir;
    end;

    f.Free;
end;

//-----
```

```

procedure TFormMain.BitBtnSetFieldsClick(Sender: TObject);
var
  f: TAddEditEmployeeForm;
begin
  f := TAddEditEmployeeForm.Create(Self);
  f.Caption := 'Employee Edit Record';
  f.OpenPictureDialog.InitialDir := OpenPictureDialog.InitialDir;
  f.EditTitle.Text := TableEmployeesTitle.AsString; //Title
  f.EditLastName.Text := TableEmployeesLastName.AsString; //LastName
  f.EditFirstName.Text := TableEmployeesFirstName.AsString; //FirstName
  f.EditPosition.Text := TableEmployeesPosition.AsString; //Position
  f.MaskEditBirthDate.Text := DBEditBirthDate.Text; //BirthDate
  f.MaskEditHireDate.Text := DBEditHireDate.Text; //HireDate
  f.EditAddress.Text := TableEmployeesAddress.AsString; //Address
  f.EditCity.Text := TableEmployeesCity.AsString; //City
  f.EditRegion.Text := TableEmployeesRegion.AsString; //Region
  f.EditPostalCode.Text := TableEmployeesPostalCode.AsString; //PostalCode
  f.EditCountry.Text := TableEmployeesCountry.AsString; //Country
  f.EditHomePhone.Text := TableEmployeesHomePhone.AsString; //HomePhone
  f.EditExtension.Text := TableEmployeesExtension.AsString; //Extension
  f.ImagePhoto.Picture.Bitmap.Assign(DBImagePhoto.Picture.Bitmap); //Photo
  f.MemoNotes.Text := TableEmployeesNotes.AsString; //Notes
  f.EditSalary.Text := TableEmployeesSalary.AsString; //Salary
  if (f.ShowModal() = mrOk) then
  begin
    if (TableEmployees.State <> dsEdit) then TableEmployees.Edit();
    TableEmployees.SetFields([nil, //ID
      f.EditTitle.Text, //Title
      f.EditLastName.Text, //LastName
      f.EditFirstName.Text, //FirstName
      f.EditPosition.Text, //Position
      f.MaskEditBirthDate.Text, //BirthDate
      f.MaskEditHireDate.Text, //HireDate
      f.EditAddress.Text, //Address
      f.EditCity.Text, //City
      f.EditRegion.Text, //Region
      f.EditPostalCode.Text, //PostalCode
      f.EditCountry.Text, //Country
      f.EditHomePhone.Text, //HomePhone
      f.EditExtension.Text, //Extension
      f.ImagePhoto.Picture.Bitmap, //Photo
      f.MemoNotes.Text, //Notes
      f.EditSalary.Text]); //Salary
    TableEmployees.Post();
  end;
  if (OpenPictureDialog.InitialDir <> f.OpenPictureDialog.InitialDir) then
    OpenPictureDialog.InitialDir := f.OpenPictureDialog.InitialDir;
  f.Free;
end;

```

```

procedure TFormMain.BitBtnIncDecEmpSalaryClick(Sender: TObject);
var
    Salary: Double;
begin
    Salary := 0.0;

    if ((TableEmployees.State <> dsInsert) or
        (TableEmployees.State <> dsEdit)) then TableEmployees.Edit();

    Salary := TableEmployeesSalary.AsFloat;

    if (ComboBoxIncDec.ItemIndex = 0) then
    begin
        Salary := Salary + Salary * 0.02; //Increase
    end;

    if (ComboBoxIncDec.ItemIndex = 1) then
    begin
        Salary := Salary - Salary * 0.02; //Decrease
    end;

    TableEmployeesSalary.AsFloat := Salary;
    //или TableEmployees.FieldByName('Salary').AsFloat := Salary;
    //или TableEmployees.FieldValues['Salary'] := Salary;

    TableEmployees.Post();
end;

//-----

procedure TFormMain.BitBtnMoreLessClick(Sender: TObject);
begin
    if (BitBtnMoreLess.Caption = 'Less <<') then
    begin
        Self.Constraints.MinWidth := 592;
        Self.Width := 592;
        BitBtnMoreLess.Caption := 'More >>';
    end
    else
    begin
        Self.Constraints.MinWidth := 765;
        Self.Width := 765;
        BitBtnMoreLess.Caption := 'Less <<';
    end;
end;

```

Стъпка 5: Активирайте модула AddEditEmployeeForm и добавете имплементация на следващите методи на класа TAddEditEmployeeForm:

```
//-----  
procedure TAddEditEmployeeForm.miDeletePhotoClick(Sender: TObject);  
begin  
    if (MessageDlg('Delete This Photo',  
        TMsgDlgType.mtConfirmation, mbYesNo, 0) <> mrYes) then Exit;  
    if (ImagePhoto.Picture.Bitmap <> nil) then  
        ImagePhoto.Picture.Bitmap := nil;  
end;  
  
//-----  
procedure TAddEditEmployeeForm.miLoadPhotoFromFileClick(Sender: TObject);  
begin  
    if (OpenPictureDialog.Execute()) then  
        ImagePhoto.Picture.LoadFromFile (OpenPictureDialog.FileName);  
end;  
  
//-----  
procedure TAddEditEmployeeForm.EditPositionChange(Sender: TObject);  
begin  
    BitBtnOk.Enabled :=  
        (EditPosition.Text <> '') and  
        (EditTitle.Text <> '') and  
        (EditFirstName.Text <> '') and  
        (EditLastName.Text <> '') and  
        (EditCountry.Text <> '') and  
        (EditCity.Text <> '') and  
        (EditRegion.Text <> '') and  
        (EditAddress.Text <> '') and  
        (EditPostalCode.Text <> '') and  
        (EditHomePhone.Text <> '') and  
        (MaskEditBirthDate.Text <> '') and  
        (MaskEditHireDate.Text <> '') and  
        (EditSalary.Text <> '') and  
        (EditExtension.Text <> '') and  
        (MemoNotes.Text <> '') and  
        (ImagePhoto.Picture.Bitmap <> nil);  
end;  
  
//-----  
procedure TAddEditEmployeeForm.ImagePhotoDblClick(Sender: TObject);  
begin  
    miLoadPhotoFromFileClick(Self);  
end;  
  
//-----
```