

ТЕХНОЛОГИЯ СОЗДАНИЯ МНОГООКОННЫХ ПРИЛОЖЕНИЙ MDI СТИЛЯ В СРЕДЕ ПРОГРАММИРОВАНИЯ DELPHI

TECHNOLOGY FOR CREATING MULTI-WINDOW MDI-STYLE APPLICATIONS IN THE DELPHI PROGRAMMING ENVIRONMENT

**D. Zdor
E. Savelyeva
D. Lomonosov**

Summary. When creating an application, the initial stage of work is the development of an interface. Among all applications built as multiwindows, one can single out a special style — MDI. Despite the rather rare distribution in the practice of programming multi-window MDI-style applications, there is a need for them, so the technology for creating such applications deserves attention. The purpose of the work is to present the meaningful design elements of a multi-window MDI style application in the Delphi programming environment in a holistic way. In the article, an example presents in a holistic way the technology for designing a multi-window MDI style application. At each stage of the technology implementation, the content of its elements is determined. Comments presented in the program code explain the software implementation of technology elements. The content elements of the technology described in the paper can be used as invariant, which allows them to be used as a template when creating multi-window MDI applications.

Keywords: window, form, multi-window application, form style.

Здор Дмитрий Валерьевич

*К.п.н., доцент, ФГБОУ ВО «Приморская
государственная сельскохозяйственная академия»
dmitriy.dv@inbox.ru*

Савельева Екатерина Владимировна

*К.т.н., доцент, ФГБОУ ВО «Приморская
государственная сельскохозяйственная академия»
savva.6969@mail.ru*

Ломоносов Дмитрий Александрович

*К.т.н., доцент, ФГБОУ ВО «Приморская
государственная сельскохозяйственная академия»
lomonosov_dim@mail.ru*

Аннотация. При создании приложения первоначальным этапом работы является разработка интерфейса. Среди всех приложений, построенных как многооконные, можно выделить особый стиль — MDI. Несмотря на достаточно редкое распространение в практике программирования многооконных приложений MDI стиля, в них есть необходимость, поэтому технология создания таких приложений заслуживает внимания. Цель работы состоит в представлении содержательных элементов проектирования многооконного приложения MDI стиля в среде программирования Delphi в целостном виде. В статье на примере представлена в целостном виде технология проектирования многооконного приложения MDI стиля. На каждом этапе реализации технологии определено содержание ее элементов. Комментарии, представленные в программном коде, поясняют программную реализацию элементов технологии. Описанные в работе содержательные элементы технологии могут использоваться как инвариантные, что позволяет использовать их как шаблон при создании многооконных MDI приложений.

Ключевые слова: окно, форма, многооконное приложение, стиль формы.

При создании приложения первоначальным этапом работы является разработка интерфейса. В условиях преобладания в настоящее время оконного вида интерфейса в современных графических операционных системах актуальным является вопрос о проектировании приложений, имеющих в своей структуре более одного окна. Такие приложения именуются многооконными. Объектом изучения в настоящей работе являются инвариантные элементы технологии создания многооконных приложений в среде программирования Delphi.

Среди всех приложений, построенных как многооконные, можно выделить особый стиль — MDI (Multi Document Interface) многооконные приложения. Основная особенность их состоит в том, что главное окно приложения является родительским (рамочным), а дру-

гие окна создаются как дочерние, они находятся внутри главного окна и не выходят за его пределы. Все элементы управления (меню, панели инструментов и т.д.) для работы с дочерним окном и его содержимым находятся в главном (родительском) окне.

Следует отметить, что еще одной особенностью многооконных приложений MDI стиля является то обстоятельство, что родительские окна создаются не в момент запуска приложения на исполнения, а по мере необходимости, по специальной команде пользователя уже в ходе исполнения приложения. Это в значительной степени отличает приложения MDI стиля от многооконных приложений SDI стиля, в которых все окна относительно самостоятельные, могут иметь свои меню, панели инструментов, группы управляющих элементов, и все они создаются на момент старта приложения, но могут быть

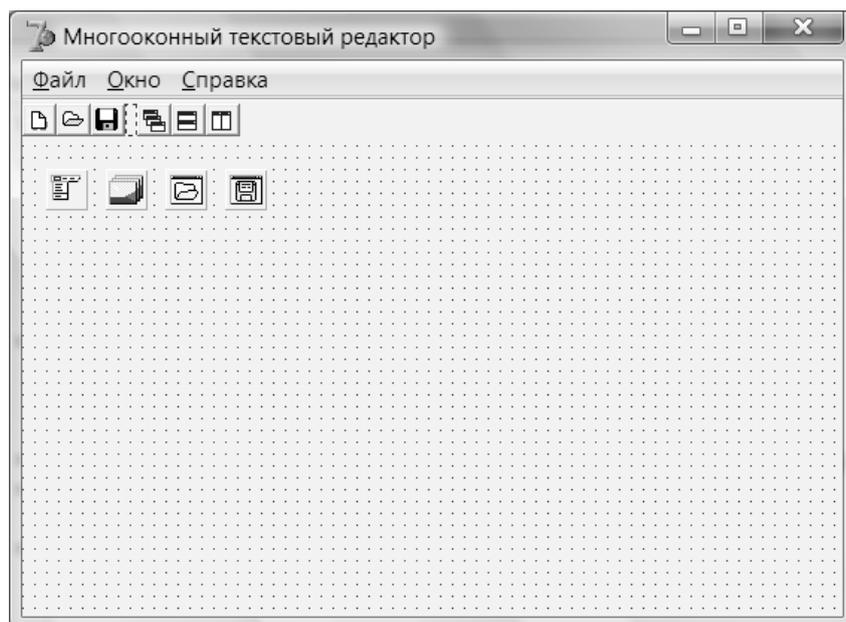


Рис. 1. Интерфейс родительской формы

скрыты от пользователя до вызова им специальной командой [5, с. 87].

Несмотря на достаточно редкое распространение в практике программирования многооконых приложений MDI стиля, в них есть необходимость, поэтому технология создания таких приложений заслуживает внимания, и данный аспект в вопросе создания многооконых приложений в Delphi выступает предметом изучения в данной работе.

Delphi относится к системам быстрой разработки приложений, которые характеризуются автоматизацией процесса проектирования, достигнутой за счет возможности использования готовых компонентов и автоматического составления большей части кода приложения. Это позволяет в простых случаях составлять приложения шаблонно, без особых знаний о деталях работы системы [8, с. 33]. Таким образом, технологизация процесса создания приложения в Delphi имеет высокий уровень.

Различным аспектам программирования в среде Delphi посвящены многие работы. Язык объектно-ориентированного программирования Object Pascal (в более поздних версиях среды именуется как язык Delphi), являющийся инструментальной основой среды программирования Delphi, рассмотрен в работах Тейксейра [1], А.Я. Архангельского [2].

Готовым компонентам среды Delphi как элементам автоматизации проектирования графического интерфейса оконого приложения большое внимание уде-

лено в работах В.Э. Гофмана [3], П.Г. Дарахвелидзе [4], В.М. Пестрикова [7].

Технология создания оконого приложения рассмотрена в работах Н.Б. Культина [6], А.Г. Федорова [9]. Учитывая, что прообразом окона на этапе проектирования приложения является форма, особый интерес представляют работы В.Э. Гофмана [3], П.Г. Дарахвелидзе [4], М.Е. Фленова [10]. Авторами рассмотрен объект-форма как инструмент создания окона приложения, освещены вопросы создания, визуализации, удаления оконов, взаимодействия между оконами. Приведены методы работы с оконами, в том числе с дочерними, с демонстрацией фрагментов программных кодов по управлению оконами.

Анализ литературных источников показал, что вопрос создания многооконых приложений в среде программирования Delphi имеет достаточное освещение в научной литературе. Вместе с тем обнаруживается недостаточность представления вопроса о технологизации проектирования многооконых приложений MDI стиля с использованием форм. Полученные результаты позволили определить цель, которая состоит в представлении содержательных элементов проектирования многооконого приложения MDI стиля в среде программирования Delphi в целостном виде. Рассмотрение вопроса проведем на примере создания многодокументного текстового редактора с демонстрацией отдельных технологических приемов.

Стандартно после создания нового приложения в среде на экране появляется форма. Свойству Name

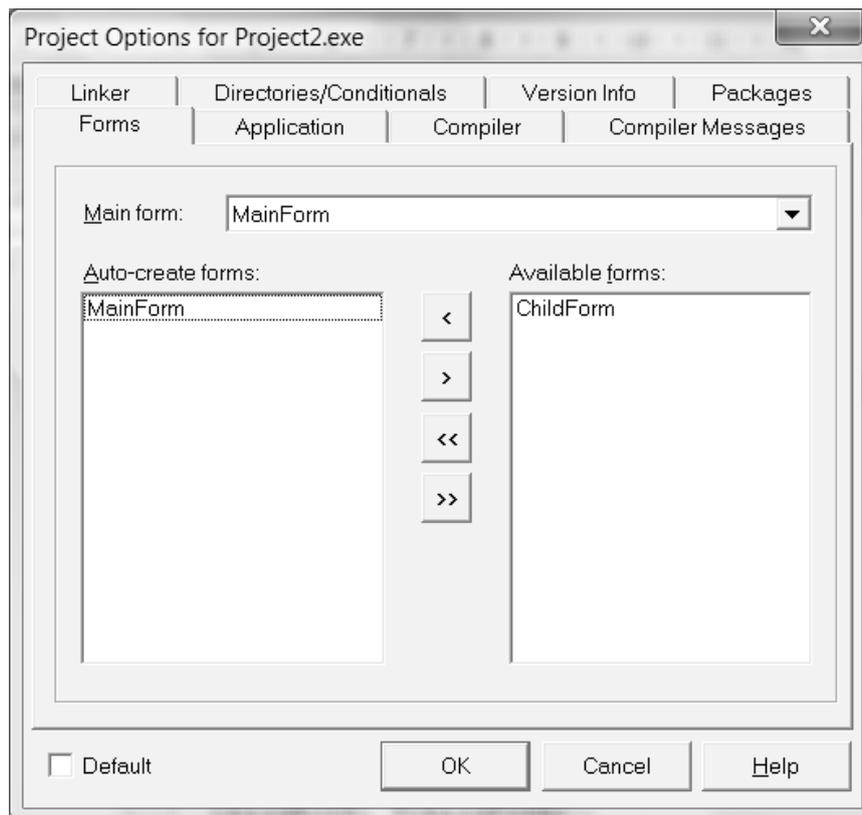


Рис. 2. Окно параметров проекта

формы зададим значение MainForm. Далее в инспекторе объектов необходимо установить свойство FormStyle для формы в fsMDIForm. Это означает, что данное окно будет родительским в MDI-приложении.

Следующим этапом является разработка интерфейса родительской формы. Для примера создадим графический интерфейс родительской формы, приведенный на рисунке 1, используя компоненты MainMenu1, ToolBar1, ImageList1, OpenFileDialog1, SaveDialog1.

В группу меню *Файл* включим команды: *Создать*, *Открыть...*, *Сохранить...*, *Заккрыть*, *Заккрыть всё*. Таким образом, данная группа содержит команды выполнения основных операций с файлами документов. В группу меню *Окно* включим команды: *Каскадом*, *По горизонтали*, *По вертикали*. Это группа меню содержит команды упорядочивания дочерних окон в рамках главного окна приложения. Пункт меню *Справка* пусть для примера содержит одну команду *О программе*, предполагающей вывод краткого информационного сообщения.

Следующим этапом является разработка дочернего окна документа. Для этого необходимо добавить в проект новую форму. Пусть ее будет ChildForm. Для того

чтобы данное окно было дочерним в MDI-приложении, необходимо установить для дочернего окна MDI-приложения свойство FormStyle в fsMDIChild. Подчеркнем, что в Delphi для создания рамочного окна используется стиль fsMDIForm, а для создания дочернего MDI-окна — стиль fsMDIChild.

Отметим также, что при разработке MDI приложений актуальны следующие свойства. Свойство MDIChildCount в рамочном MDI-окне определяет количество связанных с ним дочерних окон. Свойство MDIChildren [N: integer] в рамочном MDI окне позволяет получить прямой доступ к дочернему окну под номером N.

В дочернем окне разместим его элементы, например поле текстового редактора с помощью компонента RichEdit1.

Важным технологическим приемом является то обстоятельство, что дочернее окно должно создаваться по команде. В связи с этим форму ChildForm необходимо исключить из разряда автоматически создаваемых форм. Для этого необходимо выбрать команду *Project—Options*, затем в диалоговом окне *Options* на вкладке *Forms* (рис. 2) переместить одинарной стрелкой пере-нести ChildForm в сторону *Available forms*.

Поскольку из первого окна (родительского) осуществляется вызов второго (дочернего), то необходимо подключить модуль, соответствующий второму окну (дочернему). Для этого выберите в меню *File* пункт *Use Unit...*, и укажите модуль, соответствующий дочернему окну.

Теперь необходимо разработать обработчики событий для каждой из команд меню, представленных в главном окне. Приведем примеры соответствующих обработчиков событий с пояснениями, записанными в комментариях к программному коду.

Обработчик события `OnClick` для пункта меню *Создать*.

```
procedure TMainForm.CreateClick(Sender: TObject);
begin
  n:= n+1; // определение порядкового номера дочер-
ного окна
  TChildForm.Create(Self); // создание дочернего окна
  // задание заголовка активному дочернему окну
  (ActiveMDIChild as TChildForm).
Caption:=(ActiveMDIChild as TChildForm).Caption+'
'+InttoStr(n);
end;
```

Обработчик события `OnClick` для пункта меню *Закрыть*.

```
procedure TMainForm.CloseClick(Sender: TObject);
begin
  ActiveMDIChild.Free; // закрытие активного дочернего
окна
end;
```

Обработчик события `OnClick` для пункта меню *Закрыть всё*.

```
procedure TMainForm.ClosallClick(Sender: TObject);
var i: integer;
begin
  for i:=0 to MdiChildCount-1 do // определение количе-
ства дочерних окон
    MDIChildren[i].Close; // закрытие дочернего окна с но-
мером
end;
```

Обработчик события `OnClick` для пункта меню *Открыть...* Этот обработчик предполагает вызов диалогового окна и прямое обращение к элементу дочернего окна `RichEdit1`.

```
procedure TMainForm.OpenClick(Sender: TObject);
begin
  if OpenFileDialog1.Execute then
```

```
begin
  fname:=OpenDialog1.FileName;
  (ActiveMDIChild as TChildForm).RichEdit1.Lines.
LoadFromFile(fname);
  (ActiveMDIChild as TChildForm).Caption:=fname;
end;
end;
```

Аналогичную структуру имеет обработчик события `OnClick` пункта меню *Сохранить...*

```
procedure TMainForm.SaveClick(Sender: TObject);
begin
  Savedialog1.FileName:=(ActiveMDIChild as TChildForm).
Caption;
  if Savedialog1.Execute then
    begin
      fname:=SaveDialog1.FileName;
      (ActiveMDIChild as TChildForm).RichEdit1.Lines.
SaveToFile(fname);
      (ActiveMDIChild as TChildForm).Caption:=fname;
    end;
end;
```

Приведем обработчики событий для упорядочивания дочерних окон в рамках главного (родительского) окна.

```
procedure TMainForm.CascadeClick(Sender: TObject);
begin
  cascade; // упорядочивание каскадом
end;
procedure TMainForm.HorisonalClick(Sender: TObject);
begin
  TileMode:=tbHorizontal; // упорядочивание по гори-
зонтали
  Tile;
end;
procedure TMainForm.VerticalClick(Sender: TObject);
begin
  TileMode:=tbVertical; // упорядочивание по вертикали
  Tile;
end;
```

Для того чтобы при запуске многооконного редактора создавалось первое дочернее окно документа, в обработчик события `OnActivate` для формы `MainForm` необходимо добавить вызов команды создания нового окна, то есть вызов обработчика `OnClick` для пункта меню *Создать*.

```
procedure TMainForm.FormActivate(Sender: TObject);
begin
  MainForm.Create.Click;
end;
```

Поскольку закрытие дочернего окна (кнопкой *Закрыть* в строке заголовка) в MDI приложении всего лишь минимизирует его в клиентской области родительского окна, то необходимо обеспечить процедуру OnClose, которая будет закрывать дочернее окно. Для этого необходимо форму ChildForm связать с событием OnClose, и написать обработчик этого события следующего содержания.

```
procedure TChildForm.FormClose(Sender: TObject; var
Action: TCloseAction);
begin
Action:= caFree;
end;
```

Обработчик события пункта меню *О программе* может содержать команду вывода произвольного текстового сообщения.

```
procedure TMainForm.AboutClick(Sender: TObject);
begin
ShowMessage('Многооконный текстовый редактор');
end;
```

Рассмотрение на примере технологии проектирования MDI многооконного приложения позволяет сформулировать основные ее содержательные элементы:

1) проектирование интерфейса главного (рамочного) окна; 2) добавление к проекту новой формы и разработка интерфейса дочернего окна; 3) удаление формы дочернего окна из автоматически создаваемых форм; 4) осуществление взаимосвязи модулей, соответствующих родительской и дочерней формам; 5) составление обработчиков событий с применением методов работы с дочерними окнами, в том числе с осуществлением прямого доступа к элементам дочерних окон.

Таким образом, на примере в работе представлена в целостном виде технология проектирования многооконного приложения MDI стиля. На каждом этапе реализации технологии определено содержание ее элементов. Комментарии, представленные в программном коде, поясняют программную реализацию элементов технологии. Работа не претендует на единственность и точность в технологическом исполнении. Обработчики событий могут быть составлены иначе с использованием других методов и процедур. Вместе с тем, описанные в работе содержательные элементы технологии могут использоваться как инвариантные, что позволяет использовать их как шаблон при создании многооконных MDI приложений. Полученные результаты могут быть использованы в дальнейшей разработке вопросов создания многооконных приложений в Delphi.

ЛИТЕРАТУРА

1. Borland Delphi 6: руководство разработчика: Пер. с англ. [Текст] / Стив Тейксейра, Ксавье Пачеко. — М.: Вильямс, 2002. — 1112 с.
2. Архангельский, А.Я. Delphi 7 [Текст] / А.Я. Архангельский. — М.: Изд-во БИНОМ, 2003. — 1023 с.
3. Гофман, В.Э. Delphi. Быстрый старт [Текст] / В.Э. Гофман, А.Д. Хомоненко. — СПб.: БХВ-Петербург, 2003. — 288 с.
4. Дарахвелидзе, П.Г. Delphi 2005 для Win32 [Текст] / П.Г. Дарахвелидзе, Е.П. Марков. — СПб.: БХВ-Петербург, 2005. — 1112 с.
5. Здор, Д.В. Создание многооконных приложений SDI стиля в среде программирования Delphi [Текст] / Д.В. Здор, Е.В. Савельева // Современная наука: актуальные проблемы теории и практики. Серия «Естественные и технические науки». — 2021. — № 3. — С. 86–91.
6. Культин, Н.Б. Основы программирования в Delphi XE [Текст] / Н.Б. Культин. — СПб.: БХВ-Петербург, 2011. — 416 с.
7. Пестриков, В.М. Delphi на примерах [Текст] / В.М. Пестриков, А.Н. Маслобоев. — СПб.: БХВ-Петербург, 2005. — 496 с.
8. Ремнев, А.А. Курс Delphi для начинающих. Полигон нестандартных задач [Текст] / А.А. Ремнев, С.В. Федотова. — М.: Солон-Пресс, 2015. — 360 с.
9. Федоров, А.Г. Создание Windows-приложений в среде Delphi [Текст] / А.Г. Федоров. — М.: ТОО «Компьютер Пресс», 1999. — 347 с.
10. Фленов, М.Е. Библия Delphi [Текст] / М.Е. Фленов. — 2-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2008. — 800 с.

© Здор Дмитрий Валерьевич (dmitriy.dv@inbox.ru), Савельева Екатерина Владимировна (savva.6969@mail.ru),

Ломоносов Дмитрий Александрович (lomonosov_dim@mail.ru).

Журнал «Современная наука: актуальные проблемы теории и практики»