Практическая работа 8.
MDI-приложения. Создание многооконного текстового редактора

Задание 1. Создайте многооконный текстовый редактор, который будет иметь следующие возможности: редактировать текст в нескольких окнах, изменять шрифт текста, выравнивать строки в абзацах, поиск и замена текста, выделение, копирование и вставка фрагментов текста, сохранение текста.

Решение:

Создайте на диске папку «Текстовый редактор».

Запустите текстовый редактор Word, при помощи которого создайте два текстовых документа с произвольным текстом и сохраните их по именами «*Документ1.rtf*» и «*Документ2.rtf*» в папке «Текстовый редактор».

Запустите интегрированную среду Delphi. Переименуйте имя формы, заменив значение свойства Form1.Name на MainForm. Задайте для свойства MainForm.Caption значение «Текстовый редактор». Чтобы окно редактора было родительским окном в MDI-приложении, задайте для свойства MainForm.FormStyle значение fsMDIForm.

Примечание: в MDI-приложениях два и более окон могут быть активны одновременно. Это удобное средство для одновременной работы с разными файлами.

Сохраните файл проекта и модуля в папке «Текстовый редактор» под именем main.pas, Edit.dpr. Для создания меню приложения поместите на форму компонент MainMenu из палитры Standard. Дважды щелкните мышью по компоненту на форме, чтобы раскрыть конструктор меню окна. Далее, пользуясь свойством Caption, задайте для меню следующую структуру:

Файл	**Правка**	**Формат**		**Окна**
Создать	Отменить	Шрифт		Каскад
Открыть	Вырезать	Выравнивание -	Влево	Горизонтально
Сохранить	Копировать		По центру	Вертикально
Выход	Вставить		Вправо	
	Найти			
	Заменить			

Чтобы во время работы приложения все дочерние окна отображались списком в меню Окна, выберите пункт меню Окна и переименуйте его в свойстве Name - WindowMenu. Затем выберите форму MainForm, в свойстве WindowMenu выберите из раскрывающегося списка пункт меню WindowMenu.

Спроектируйте окно дочернего документа, для этого добавьте в проект новую форму: File>New>Form. Сделайте окно новой формы меньше родительского (главного) и переименуйте свойство Name этого окна в ChildForm. Задайте для свойства ChildForm.FormStyle значение fsMDIChild. Поместите на это форму компонент RichEdit из палитры Win32. Чтобы компонент занимал все окно формы, и во время выполнения приложения его размеры изменялись при изменении размеров окна, измените свойство RichEdit.Align на alClient. Дважды щелкнув мышью в поле значений свойства RichEdit1.Lines в открывшемся окне удалите текст и нажмите OK.

Сохраните новый модуль под именем Child.pas в папке «Текстовый редактор». Нажмите сочетание клавиш Shift+F12 и выберите в окне главную форму приложения MainForm.

Поместите на главной форме компоненты палитры Dialogs: OpenDialog, SaveDialog, FontDialog и ActionList из палитры Standard. Эти компоненты понадобятся

при обработке событий приложения. Так как они не являются визуальными, то их можно разместить в любом месте формы.

Для создания дочернего окна документа в редакторе только при выборе меню **Файл → Создать** отключите создание окна во время запуска приложения, которое задается Delphi по умолчанию. Для этого выберите команду меню Project>Options. В окне на вкладке Forms выберите форму ChildForm и перенесите имя дочерней формы из списка Auto-create forms (формы, автоматически создающиеся при старте приложения) в список Available (доступные формы) и щелкните мышкой по кнопке ОК.

Для открытия нового дочернего окна приложения создайте процедуру обработчика выбора команды Файл\Создать. Для этого выберите на форме MainForm в меню Файл команду Создать. При этом в программном модуле Main будет сгенерирована процедура, которую нужно отредактировать:

```
procedure TMainForm.N9Click(Sender: TObject);
begin
TChildForm.Create(Self); {создать дочернее окно документа}
end;
```

Так как в этом окне создается другое окно, то к модулю Main необходимо подключить модуль Child. Для этого следует выбрать команду File > Use Unit…, в окне указать модуль Child и щелкнуть по кнопке ОК.

Сохраните проект, проверьте работу процедуры, выбирая в меню приложения команду Файл\Создать. Создаваемое окно не закрывается, а минимизируется. Для того чтобы окно создаваемого документа имело название «Новый», создайте процедуру обработки события OnCreate для объекта ChildForm. Для этого откройте форму ChildForm и в окне Инсперктора объектов на странице События дважды щелкните мышкой в поле значений события OnCreate, затем добавьте оператор:

```
procedure TChildForm.FormCreate(Sender: TObject);
begin
Caption:='Новый'; // Заголовок окна нового файла
end;
```

Чтобы окно нормально закрывалось, перейдите в окно ChildForm (Shift+F12), затем выберите в окне Инспектора объектов на странице События OnClose и дважды щелкните мышкой по пустому полю. Отредактируйте заготовку процедуры закрытия окна:

```
procedure TChildForm.FormClose(Sender: TObject; var Action:
TCloseAction);
begin
Action:=caFree; // Закрыть форму и уничтожить из памяти
end;
```

Для создания процедуры закрытия окна редактора при выборе команды Файл\Выход выберите команду в главной форме и на странице События щелкните мышкой в поле значений события OnClick. После этого добавьте в процедуру оператор:

```
procedure TMainForm.N14Click(Sender: TObject);
begin
Close; // Закрыть окно приложения
end;
```

Создайте процедуру открытия файла с помощью компонента OpenDialog (см. лаб. раб. №14). Выберите в меню Файл команду Открыть на главной форме. Отредактируйте процедуру следующим образом:

```
procedure TMainForm.N10Click(Sender: TObject);
begin
If OpenDialog1.Execute then begin { если файл выбран}
ChildForm:=TChildForm.Create(Self); {создать дочернее окно}
ChildForm.RichEdit1.Lines.LoadFromFile(OpenDialog1.FileName)
; {загрузить выбранный файл}
ChildForm.Caption:=OpenDialog1.FileName;    {установить    в
заголовке окна название файла}
ChildForm.RichEdit1.Tag:=0; {признак неизмененного файла}
end;
end;
```

Сохраните изменения в проекте и запустите приложение. При открытии файла выбирайте документы с расширением .rtf или .txt.

Для создания процедуры изменения шрифта выберите на главной форме приложения команду Формат\Шрифт и отредактируйте заготовку процедуры обработки события следующим образом:

```
procedure TMainForm.N16Click(Sender: TObject);
Begin
if FontDialog1.Execute then
(ActiveMDIChild                                          as
TChildForm).RichEdit1.Font:=FontDialog1.Font;
{Установить выбранный шрифт}
end;
```

Создайте самостоятельно процедуру Сохранения файлов, используя компонент SaveDialog (см. пр.р №7).

Для того чтобы изменить порядок отображения файлов в окне приложения, необходимо создать процедуру обработки команд меню Окна. Откройте главную форму приложения и дважды щелкните по компоненту ActionList1. В открывшемся окне нажмите Ctrl+Ins и выберите пункт New Standatd Action из раскрывающегося меню стандартных действий выберите при нажатой клавише Shift выберите все стандартные действия: TWindowCascade, TWindowTitleHorizontal, TWindowTitleVertical и щелкните ОК. После этого в списке действий появится категория Window.

Укажите для каждой из команд меню Окна соответствующее стандартное действие. Выбрав в окне ObjectTreeView команду Каскад, в окне Инспектора объектов на странице События откройте список значений стандартных событий Action и выберите в нем событие WindowCascade1. Так как по умолчанию все стандартные действия имеют названия на английском языке, то после этой операции название соответствующей команды меню поменяется на стандартное. Чтобы заменить это название, на странице Properties выберите свойство Caption и замените его значение на &Каскад.

Аналогичным образом задайте стандартные действия для команд меню Окна\ Горизонтально и Вертикально. Сохраните изменения в проекте и проверьте работу данного пункта меню.

Аналогичным образом подключите стандартные действия для обработки пунктов меню:

Правка (Отменить (undo), Вырезать (cut), Копировать (copy), Вставить (paste), Найти (SearchFind), Заменить (SearchReplace).

Формат - Выравнивание (слева - RichEditAlignLeft , по центру - RichEditAlignCenter, справа - RichEditAlignRight).

При этом не забудьте о русификации свойства Caption.