



**Томский государственный университет**



**Факультет информатики**

**А.В. Скворцов, Т.Н. Поддубная**

# **Создание модальных форм**

**(методические указания к  
лабораторной работе № 2)**

**Томск – 1999**

Указания РАССМОТРЕНЫ и УТВЕРЖДЕНЫ методической комиссией факультета информатики.

Протокол № \_\_\_\_\_ от « \_\_\_\_\_ » \_\_\_\_\_ 1999 г.

Председатель методической комиссии факультета информатики,  
профессор \_\_\_\_\_ Поддубный В.В.

*Утверждено.* Зав. кафедрой прикладной информатики, доцент  
\_\_\_\_\_ Сущенко С.П. « \_\_\_\_\_ » \_\_\_\_\_ 1999 г.

Методические указания посвящены программной среде Borland Delphi компании Inprise Corporation – одному из мощнейших современных средств разработки приложений для Windows.

Указания разработаны для студентов межфакультетской специализации факультета информатики Томского государственного университета. Содержат подробное описание лабораторной работы: цель, задачи, описание работы и используемых компонентов Delphi. В конце работы приведены тексты программ.

*Рецензент* – канд. техн. наук, доцент **Ю.Л. Костюк.**

© Скворцов А.В., Поддубная Т.Н., 1999

© Оформление и верстка: Скворцов А.В., 1999

Основным строительным блоком в Delphi является форма. Любая программа имеет как минимум одну связанную с ней форму, которая называется «главной». Главная форма появляется на экране в момент старта программы. Однако программа может иметь сколько угодно форм, каждая из которых решает какую-то локальную задачу и появляется на экране по мере надобности.

Современные многооконные приложения чаще всего строятся в стиле SDI (Single Document Interface), который не накладывает ограничений на положение и размеры вспомогательных форм. В стиле SDI реализована сама среда Delphi.

В данной лабораторной работе вы познакомитесь с одним из способов использования вспомогательной формы в качестве модального диалогового окна. Задача для программирования взята та же самая, что и в первой лабораторной работе. Но для ввода двух целых чисел должна использоваться вспомогательная форма, а обработка введенных чисел и вывод результатов должен проводиться в модуле, связанном с главной формой.

## Цель работы

Создание приложения с несколькими формами.

## Задачи работы

1. Ознакомление с понятиями модальной и немодальной формы.
2. Создание программы-калькулятора для вычисления суммы и произведения двух чисел с использованием диалогового окна (модальной формы).

## Описание работы

Как и предыдущей лабораторной работе, вначале необходимо создать отдельную папку для сохранения всех файлов данной работы, например, Lab2. Затем, после запуска Delphi и создания нового проекта необходимо сохранить в созданную папку главный модуль под именем MainUnit.pas (вместо предлагаемого Delphi имени Unit1.pas), а проект под именем Lab2.dpr (вместо Project1.dpr). После этого можно начинать заполнять главную форму необходимыми компонентами в соответствии с рис. 6. В табл. 2 кратко приведен список свойств различных компонентов главной формы, требующих установок.

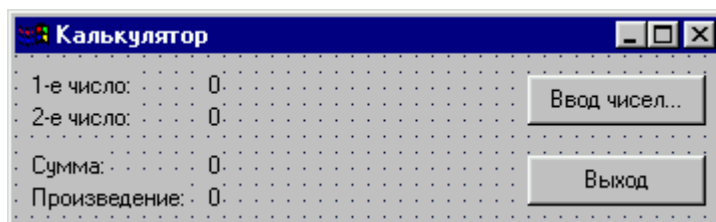


Рис. 6. Главная форма

**Таблица 2. Значения основных устанавливаемых свойств компонентов главной формы**

Свойство	Значение	Комментарий
<b>MainForm: TMainForm</b>		
Name	MainForm	Имя переменной для ссылки на форму
Caption	Калькулятор	Заголовок формы
BorderStyle	bsDialog	Запрещается растяжение формы
<b>Label1: TLabel</b>		
Caption	1-е число:	Комментарий к 1-му введенному числу
<b>Label2: TLabel</b>		
Caption	2-е число:	Комментарий ко 2-му введенному числу
<b>Label3: TLabel</b>		
Caption	Сумма:	Комментарий к выводимому значению
<b>Label4: TLabel</b>		
Caption	Произведение:	Комментарий к выводимому значению
<b>LabelSum: TLabel</b>		
Caption	0	Значение суммы по умолчанию
<b>LabelProduct: TLabel</b>		
Caption	0	Значение произведения по умолчанию
<b>ButtonInput: TButton</b>		
Caption	Ввод чисел...	Текст на кнопке
<b>ButtonExit: TButton</b>		
Caption	Выход	Текст на кнопке
Cancel	True	Указывается, что нажатие клавиши Esc в любом месте формы трактовалось как нажатие на эту кнопку

Теперь создадим вторую (вспомогательную) форму для ввода двух чисел. Для этого необходимо выбрать в меню команду File|New Form. В результате будет создана новая форма с соответствующим программным модулем. Модуль необходимо сохранить (командой File|Save) в рабочем каталоге Lab2 в файл Input.pas.

**Таблица 3.** Значения основных устанавливаемых свойств компонентов второй формы.

Свойство	Значение	Комментарий
<b>InputForm: TForm</b>		
Name	InputForm	Имя переменной для ссылки на форму
Caption	Ввод чисел	Заголовок формы
BorderStyle	bsDialog	Запрещается растяжение формы
<b>Label1: TLabel</b>		
Caption	1-е число:	Приглашение для ввода 1-го числа
<b>Label2: TLabel</b>		
Caption	2-е число:	Приглашение для ввода 2-го числа
<b>Edit1, Edit2: TEdit</b>		
Text	0	Вводимые значения по умолчанию
<b>ButtonOk: TButton</b>		
Caption	Ввод чисел...	Текст на кнопке
Default	True	Указывается, что нажатие клавиши Enter в любом месте формы, кроме кнопок, трактуется как нажатие на эту кнопку
<b>ButtonCancel: TButton</b>		
Caption	Выход	Текст на кнопке
Cancel	True	Указывается, что нажатие клавиши Esc в любом месте формы трактуется как нажатие на эту кнопку
ModalResult	mrCancel	Указывается, что при нажатии на эту кнопку форма закрывается с возвратом указанного значения в одноименном свойстве формы ModalResult

Теперь можно заполнить созданную вспомогательную форму необходимыми компонентами в соответствии с рис. 7. В табл. 3 кратко приведен список свойств различных компонентов формы, требующих установок.

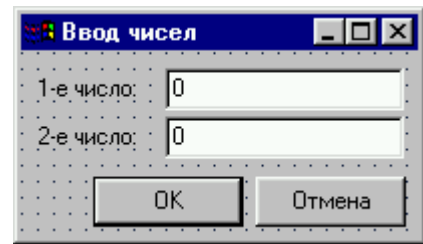


Рис. 7. Вспомогательная форма

После того как обе формы спроектированы, необходимо написать соответствующие обработчики событий для связи форм друг с другом.

В Delphi вспомогательные окна можно выводить на экран в двух режимах с помощью методов формы Show и ShowModal соответственно.

3. Метод Show открывает «немодальное» окно, которое появляется на экране и работает одновременно с вызвавшим его окном. Управление программой передается оператору, стоящему за оператором вызова метода Show. Этот режим обычно не используется для ведения диалога с пользователем. Примером такого рода окон являются окна Редактора кода и Инспектора объектов в Delphi.

4. Метод ShowModal открывает «модальное диалоговое» окно, которое полностью берет на себя дальнейшее управление программой. Модальные окна требуют от пользователя принятия какого-то решения. Для того

#### Замечание

Вывод диалоговых окон (модальных форм) на экран удобен, если они появляются по центру экрана. Для этого необходимо установить значение свойства Position формы равным psScreenCenter.

чтобы пользователь мог сообщить о принятом решении, в модальное окно вставляются необходимые интерфейсные элементы. Чаще всего это стандартные кнопки (с надписями «ОК», «Отмена», «Да», «Нет»), которые автоматически выполняют действия закрытия модального окна и помещения значения результата диалога (информации о нажатой кнопке) в свойство формы ModalResult. Свойство ModalResult может иметь значения mrNone, mrOk, mrCancel, mrYes, mrNo и другие. Вызывающая программа получает это значение после выполнения метода ShowModal и может анализировать его для определения нужного направления своей дальнейшей работы.

Нам необходимо создать обработчик событий в ответ на нажатие кнопки ButtonInput («Ввод чисел...») главной формы, который бы вызывал вспомогательную форму в модальном режиме (листинг 3). Для этого прежде всего следует указать, что вспомогательный модуль Input.pas будет

использоваться в главном модуле MainUnit.pas. Необходимо открыть модуль MainUnit.pas, выбрать команду меню **File|Use Unit...** и указать в диалоге модуль Input.pas. Это приведет к тому, что в главном модуле в секции **implementation** появится строка «**uses** Input;».

Главным в создаваемом обработчике является метод ShowModal формы InputForm, который останавливает выполнение основной программы и передает управление вспомогательной форме. В зависимости от результата выполнения метода ShowModal (нужно выполнить проверку на равенство результата значению mrOk) необходимо выполнить действия по вычислению суммы и произведения чисел и вывести полученные значения в соответствующих метках главной формы.

Во вспомогательной форме необходимо написать обработчик событий для кнопки ButtonOk («ОК»), который должен проверить правильность введения чисел и, если все верно, то закрыть форму с результатом mrOk. Для второй кнопки ButtonCancel («Отмена») обработчик писать не нужно, т.к. у этой кнопки уже выставлено свойство ModalResult, говорящее о том, что форма должна быть закрыта при нажатии на эту кнопку.



Рис. 8. Главная форма

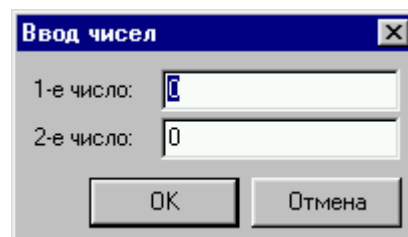


Рис. 9. Вспомогательная форма

Теперь программу можно выполнять. На рис. 8-9 приведён внешний вид разработанного приложения.

В листингах 2-4 приведены тексты файла проекта и обоих модулей.

**Листинг 2.** Текст файла проекта Lab2.dpr

```
program Lab2;

uses
  Forms,
  Input in 'Input.pas' {InputForm},
  MainUnit in 'MainUnit.pas' {MainForm};

{$R *.RES}

begin
  Application.Initialize;
  Application.CreateForm(TMainForm, MainForm);
  Application.CreateForm(TInputForm, InputForm);
  Application.Run;
end.
```

**Листинг 3.** Текст главного модуля MainUnit.pas

```
unit MainUnit;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls;

type
  TMainForm = class(TForm)
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    LabelNumber1: TLabel;
    LabelNumber2: TLabel;
    LabelSum: TLabel;
    LabelProduct: TLabel;
    ButtonInput: TButton;
    ButtonExit: TButton;
    procedure ButtonExitClick(Sender: TObject);
    procedure ButtonInputClick(Sender: TObject);
  end;

var
  MainForm: TMainForm;

implementation

uses Input;

{$R *.DFM}

procedure TMainForm.ButtonExitClick(Sender: TObject);
begin // Выход из программы
  Close;
end;

procedure TMainForm.ButtonInputClick(Sender: TObject);
begin // Нажата кнопка "Ввод чисел..."
  if InputForm.ShowModal=mrOk then
    with InputForm do
```



```
begin
  LabelNumber1.Caption:=Edit1.Text;
  LabelNumber2.Caption:=Edit2.Text;
  try
    LabelSum.Caption:=FloatToStr(Number1+Number2);
  except
    LabelSum.Caption:='Ошибка сложения';
  end;
  try
    LabelProduct.Caption:=FloatToStr(Number1*Number2);
  except
    LabelProduct.Caption:='Ошибка умножения';
  end;
end;
end.
```

#### Листинг 4. Текст вспомогательного модуля Input.pas

```
unit Input;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls;

type
  TInputForm = class(TForm)
    Label1: TLabel;
    Label2: TLabel;
    Edit1: TEdit;
    Edit2: TEdit;
    ButtonOk: TButton;
    ButtonCancel: TButton;
    procedure ButtonOkClick(Sender: TObject);
  public
    Number1, Number2: double;
  end;

var
  InputForm: TInputForm;

implementation

{$R *.DFM}

procedure TInputForm.ButtonOkClick(Sender: TObject);
begin // Перед закрытием формы проверить правильность ввода чисел
  try
    Number1:=StrToFloat(Edit1.Text);
  except
    Edit1.SetFocus;
    ShowMessage('Ошибка в первом числе');
    exit;
  end;
  try
    Number2:=StrToFloat(Edit2.Text);
  except
    Edit2.SetFocus;
    ShowMessage('Ошибка во втором числе');
  end;
end;
```

```
        exit;  
    end;  
    ModalResult:=mrOk; // Числа введены правильно, поэтому форму можно закрывать  
end;  
  
end.
```

## Вопросы и задания для самостоятельной работы

1. Каким образом создаются дополнительные окна для приложений? Как они связываются друг с другом?
2. Какие методы используются для вызова дополнительных окон?
3. Каково функциональное назначение разделов **Interface** и **Implementation** в структуре модуля?
4. Для чего Delphi вставляет в текст программы директиву компилятора `{ $\$R$  *.DFM}`?
5. В каком месте программы задается конкретный экземпляр формы, с которым она работает?
6. Как реагирует Delphi в случае, когда в строке ввода оказывается не целое число?
7. Модифицируйте программу: если среди введенных чисел есть отрицательное, то перед выполнением вычислений оно заменяется нулем.
8. Придумайте и реализуйте интерфейс для игровой программы «Угадай число».
9. Каким образом можно изменить размер окна?
10. Введите нечисловые значения в строке ввода и посмотрите, как Delphi реагирует на ошибки. После этого отключите в настройках Delphi остановку выполнения программы при возникновении исключительных ситуаций и продолжите выполнение приложения. Обратите внимание на сообщение отладчика.
11. Дополните программу для вычисления разности, частного двух чисел, максимального и минимального значения, а также факториала наименьшего числа.
12. Измените программу так, чтобы при запуске приложения метки главной формы с названиями результатов и их значений были бы невидимы.
13. Какие горячие клавиши можно использовать для выбора нужной формы или модуля, переключения между текущей формой, редактором кода и инспектором объектов?

Учебное издание

Алексей Владимирович Скворцов  
Поддубная Тамара Николаевна

Создание модальных форм

(методические указания к  
лабораторной работе № 2)