

# Объектно–ориентированное программирование



Lazarus -  
свободный аналог  
Borland Delphi

## Обработка событий

Юдина Ольга Владимировна  
МОУ средняя школа №6, Тюменская  
область, г.Когалым

# Содержание:

- Классификация языков программирования.
- Методы программирования.
- Рекомендации по оформлению кода программ.
- Управление объектом.
- Практикум по решению задач на обработку события **OnClick** и работе с компонентами **Edit** и **Label**.

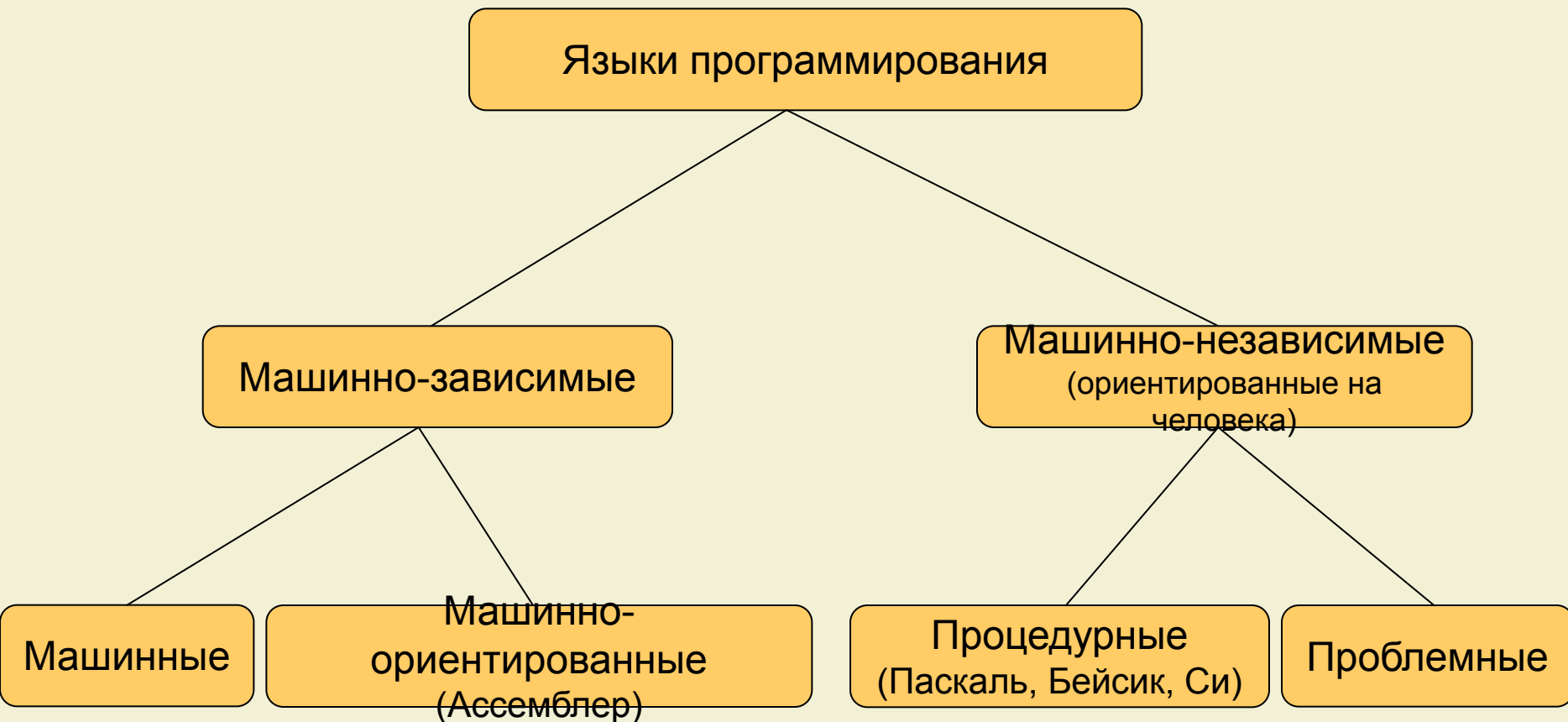
*С богом я говорю по-испански,  
С банкиром я говорю по-флорентийски,  
О любви я говорю по-французски,  
О коммерции я говорю по-английски,  
Но во время битвы я ругаю лошадь по-  
немецки.*

Карл Великий

*«Язык формирует наш способ мышления  
и определяет, о чем мы можем  
мыслить».*

*Визуальный язык программирования  
формирует наше воображение и  
определяет, что мы можем себе  
представить.*

**Б.Л.Ворф**





## **Процедурно-ориентированные языки**

Относятся к классу машинно-независимых языков.

В этих языках описывается процесс обработки информации терминами языка.

## **Машинно-ориентированные языки**

Для написания программ на таких языках используется определенный набор зарезервированных команд, выполняющих определенные операции.

Это так называемые мнемокоды



## **Машинный язык**

Система команд компьютера. То есть программы, написанные на таком языке, не требуют компиляции

## **Проблемно-ориентированные языки**

Специализированные языки, направленные на выполнение узкоспециализированных задач.



Машинно-независимые  
(ориентированные на человека)

Объектно –

Процедурные

Паскаль

Бейсик

Си

ориентированно

Borland Pascal

Visual Basic

Delphi

Lazarus

Проблемные

SQL

Perl

Кобол

Пролог

АДА





# Методы программирования

- Непосредственное (машинное) Ассемблер
- Процедурное (структурированное) Паскаль Бейсик Си
- Модульное программирование (сохранение созданных процедурных функций по определенным правилам)
- Объектно-ориентированное программирование (программа представлена в виде совокупности объектов, каждый из которых является реализацией определенного класса (вида) Delphi, Visual Basic, Borland Pascal, Visual Java,

**свободное СПО - Lazarus**



# Элементы языка Object Pascal

Рекомендации к оформлению  
кода программы



- Pascal ( в 1970 г. Н. Виртом)
- Turbo Pascal (Borland)
- Borland Pascal
- Object Pascal
- Delphi (с версии 7 Object Pascal )
- **Lazarus** – это свободный аналог Borland Delphi. Существуют версии для Windows и Linux



# Комментарии

{ Это многострочный комментарий }

(\* Это многострочный комментарий,  
допускает вложения\*)

// Это однострочный комментарий



# Рекомендации к оформлению кода программы

- Комментируйте «даже если и так все понятно».
- Не удаляйте (закомментируйте сначала).
- Записывайте исходный программный код ,  
используя отступы – «лесенку».
- Имена переменных должны быть интуитивно  
ПОНЯТНЫ.

# Объектно-ориентированное программирование



```
Unit1  
  
unit Unit1;  
  
interface  
  
uses  
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
  Dialogs, StdCtrls, ExtCtrls;  
  
type  
  TForm1 = class(TForm)  
    Button1: TButton;  
    Label1: TLabel;  
  private  
    { Private declarations }  
  public  
    { Public declarations }  
  end;  
  
var  
  Form1: TForm1;  
  Tik: Integer;  
  
implementation  
  
{ $R *.dfm }  
  
end.
```

моя программа

**Привет,МИР!!!**

**Моя кнопка**



```
Unit1
unit Unit1;      // Название модуля

interface       // Интерфейсная часть модуля

// Подключаемые модуля (основные модуля подключаются автоматически)
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls;

// Объявление класса TForm1 - наследник класса TForm
type
  TForm1 = class(TForm)
    // Список компонент помещённых на форму
    Button1: TButton;
    Button2: TButton;
    Edit1: TEdit;
    Label1: TLabel;
    // События компонент
    procedure Button2Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

// Раздел объявления глобальных переменных
var
  Form1: TForm1;      // Объект Form1, класса TForm1 (объявляется автоматически)

implementation     // Раздел описания
```

# Объектно-ориентированное программирование



```
// Обработка события от нажатия на кнопку Button1
```

```
procedure TForm1.Button1Click(Sender: TObject);
```

```
begin
```

```
// Вывод текста в компоненте Edit1, используя свойство Text
```

```
Edit1.Text:='Ура! Заработало!';
```

```
end;
```

```
// Обработка события от нажатия на кнопку Button2
```

```
procedure TForm1.Button2Click(Sender: TObject);
```

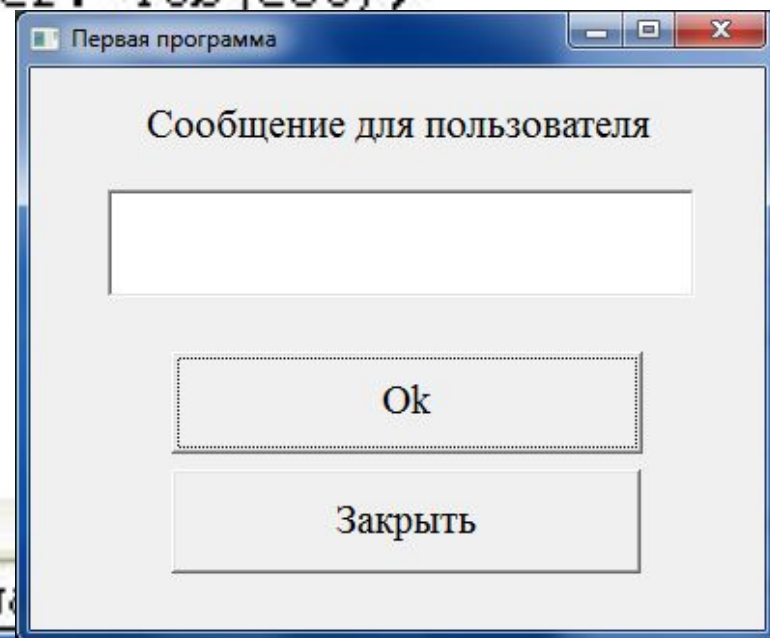
```
begin
```

```
// Закрытие программы
```

```
Close;
```

```
end;
```

```
end.
```







```
implementation
```

```
{ $R *.dfm }
```

```
procedure TForm1.ButtonMathClick(Sender: TObject);
```

```
Var Result, Num1, Num2: Integer;
```

```
begin
```

```
    // Запоминаем числа
```

```
    Num1 := StrToInt (EditNum1.text);
```

```
    Num2 := StrToInt (EditNum2.text);
```

```
    // Вычисляем сумму
```

```
    Result := Num1 + Num2;
```

```
    // Отображаем результат сложения
```

```
    EditResult.Text := IntToStr (Result);
```

```
end;
```

```
end.
```

Form1

Первое число

Второе число

Результат

21: 39

Read only

Code Diagram



# Объектно-ориентированное программирование

## УПРАВЛЕНИЕ ОБЪЕКТОМ

**Объекты** управляются через точечную пунктуацию

<Имя объекта>.<Свойство> = <Значение>

Груша1.Цена = 130

Квадрат1.Цвет = Красный

<Имя объекта>.<Метод> (<Параметры>)

Груша1.Купить      Груша1.Купить (130)

Квадрат1.Нарисовать



**<Имя компонента> . <Свойство> := <Значение свойства>;**

```
Edit1.Text := 'Привет, мир!';  
Label1.Color := ClRed;  
A := Edit1.Text;           // A - типа String
```

**<Имя компонента> . <Метод>;**

```
Edit1.Clear;  
Form1.Close;
```

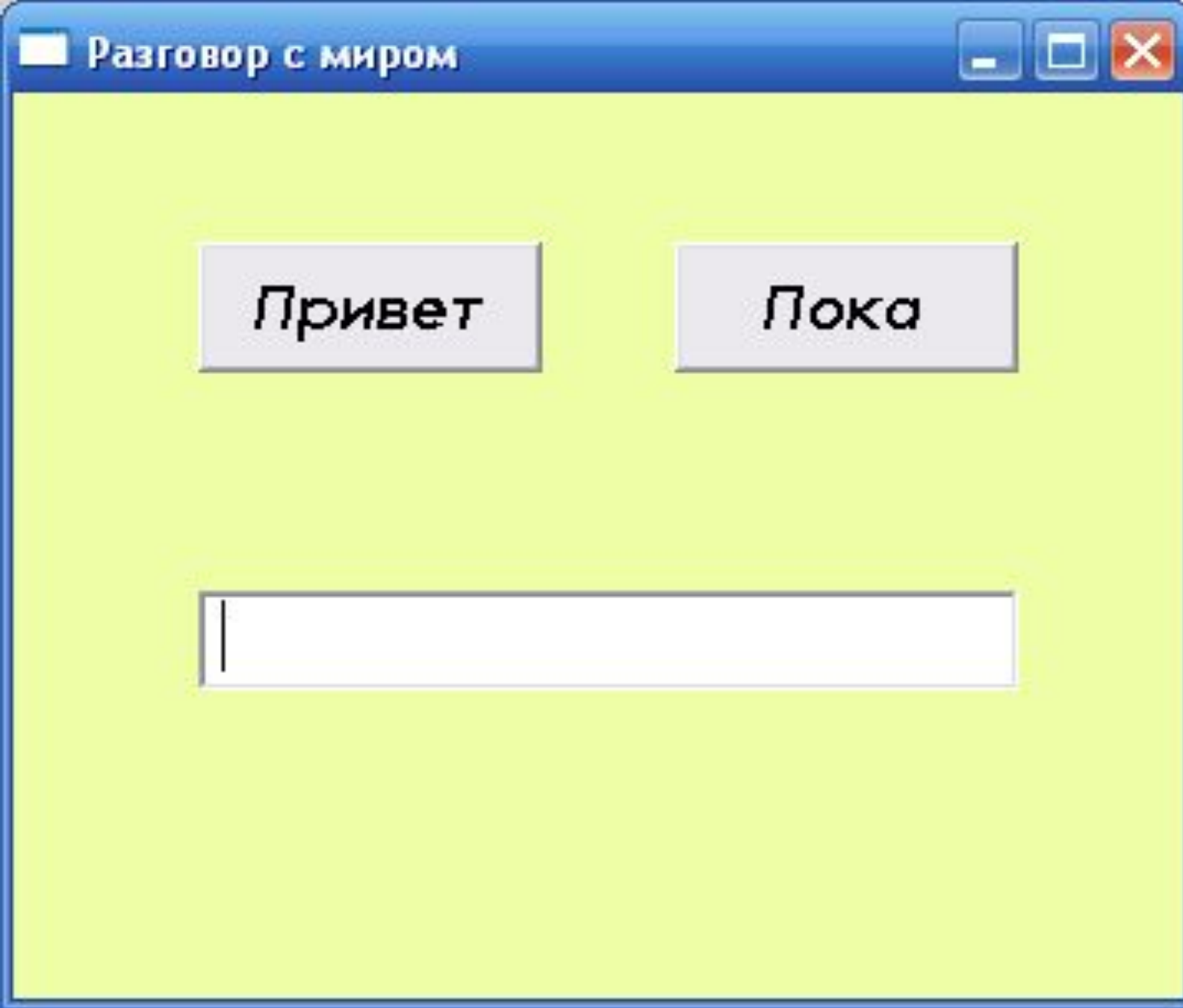
```
Edit1.Text := 'Привет, Мир!';  
Edit1.Left := 100;  
Edit1.Color := clRed;  
Edit1.Font.Color := ClGreen;
```

# Обработка события OnClick и работа компонентом Edit



## Задание 1

Создайте приложение **Разговор с миром** с полем **Edit** и двумя кнопками: **Привет** и **Пока**. При нажатии на кнопку **Привет** в поле ввода должна появиться надпись **Здравствуй, мир**, а при нажатии на кнопку **Пока** – **До свидания, мир**.





## Фрагмент программы (обработчик события)

```
implementation
```

```
{ $R *.dfm }
```

```
procedure TForm1.Button1Click(Sender: TObject);
```

```
begin
```

```
    Edit1.Text:='Здравствуй МИР';
```

```
end;
```

```
procedure TForm1.Button2Click(Sender: TObject);
```

```
begin
```

```
    Edit1.Text:='До свидания МИР' ;
```

```
end;
```



# Обработка события OnClick и работа с компонентом Edit

## Задание 2

Создайте приложение с полем ввода **Edit** и двумя кнопками: **Имя** и **Очистить**. При нажатии на кнопку **Имя** в компоненте **Edit** должно отобразиться ваше имя. А при нажатии на кнопку **Очистить** поле ввода должно быть очищено.

Form1

Имя

ОЧИСТИТЬ







## Фрагмент программы (обработчик события)

implementation

{ \$R \*.dfm }

procedure TForm1.Button1Click(Sender: TObject);

begin

**Edit1.Text:='Чернов Василий'**

end;

procedure TForm1.Button2Click(Sender: TObject);

begin

**Edit1.Text:=""**

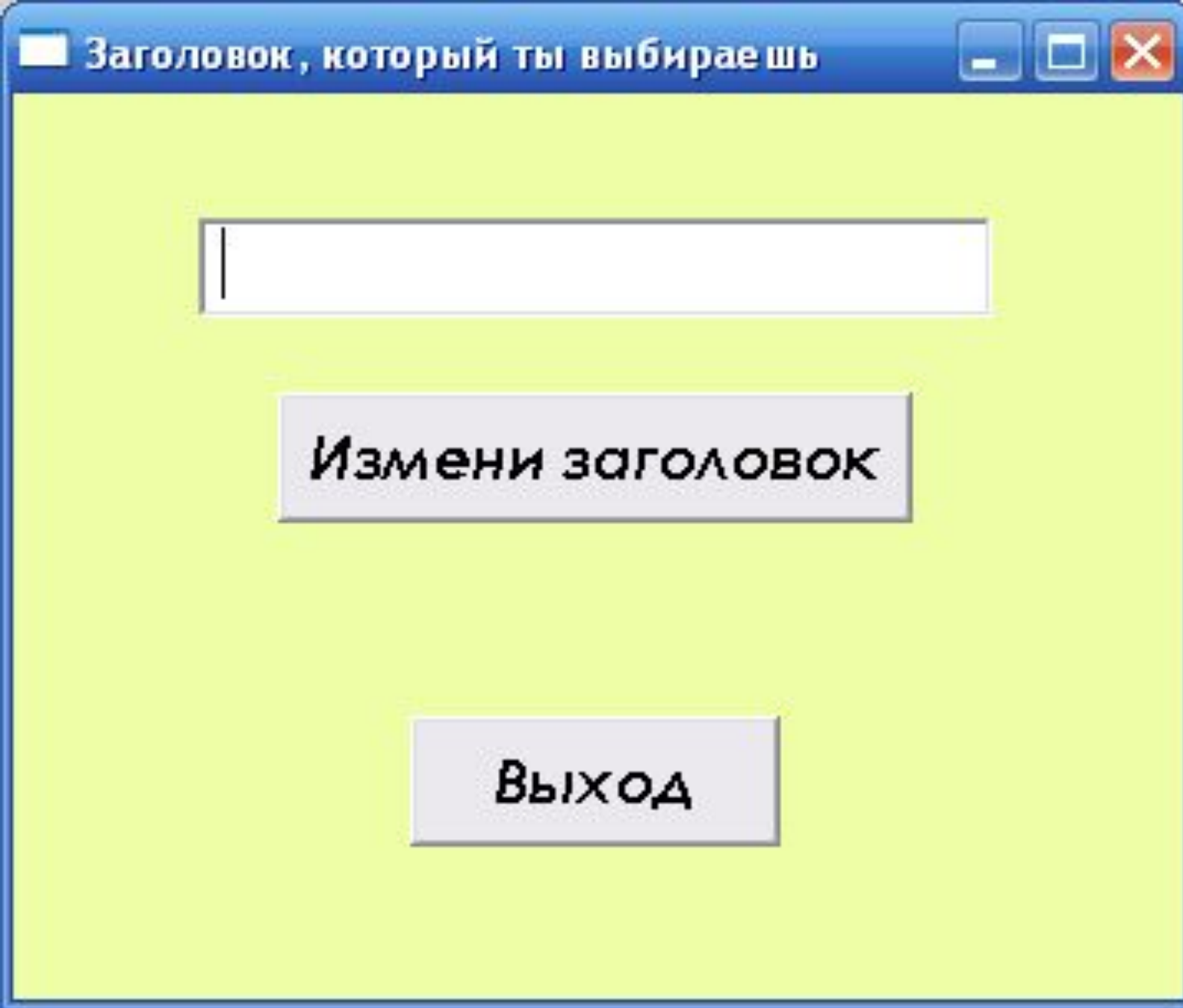
end;

# Обработка события OnClick и работа с компонентом Edit



## Задание 3

Создайте приложение Заголовок, который ты выбираешь! В поле ввода Edit пользователь заносит текст, и при нажатии на кнопку этот текст должен отобразиться в заголовке формы, которая изначально имеет надпись Впиши своё название.





# Фрагмент программы (обработчик события)

implementation

{ \$R \*.dfm }

procedure TForm1.Button1Click(Sender: TObject);

begin

    Form1.Caption:=Edit1.Text;

end;

procedure TForm1.Button2Click(Sender: TObject);

begin

    Form1.Close ;

end;



## Задача 4

- Проанализируйте фрагмент программного кода приложения и определите **какие компоненты** использованы в приложении?
- **Сформулируйте задание** по созданию приложения с заданным программным кодом.



## Фрагмент программы (обработчик события) implementation

```
{ $R *.dfm }
```

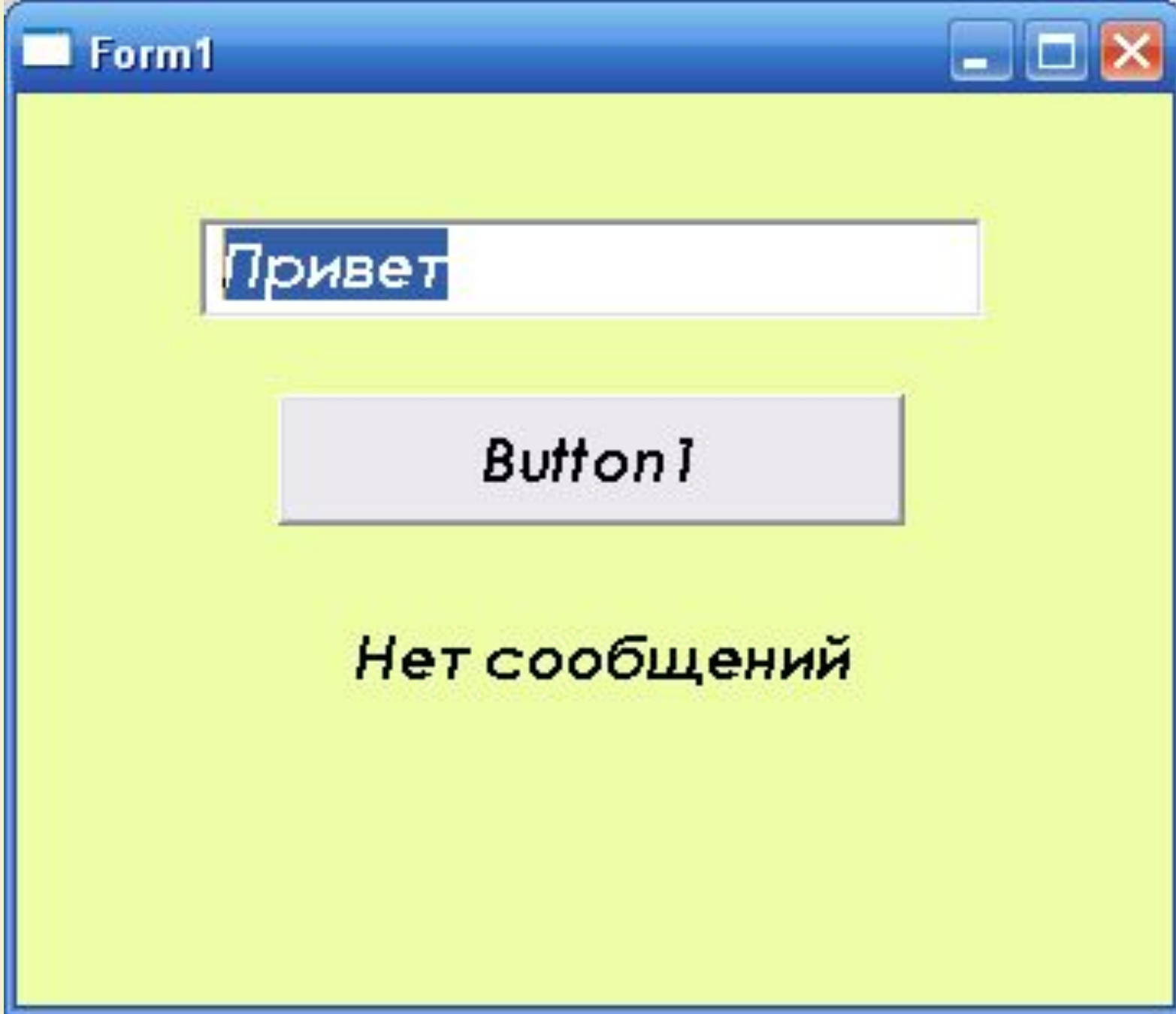
```
procedure TForm1.Button1Click(Sender:
```

```
TObject);
```

```
begin
```

```
    Label1.Caption:=edit1.Text;
```

```
end;
```





## Задача 4

Создайте приложение, где в поле ввода

**Edit** пользователь заносит текст. При

нажатии на кнопку этот текст должен

отобразиться в компоненте **Label** ,

который изначально имеет надпись **Нет**

**сообщений** .





## Задача 5

- Проанализируйте фрагмент программного кода приложения и определите **какие компоненты** использованы в приложении?
- **Сформулируйте задание** по созданию приложения с заданным программным кодом.



## Фрагмент программы (обработчик события)

```
implementation
{$R *.dfm}
procedure TForm1.Button1Click(Sender: TObject);
begin
    Edit1.Text:='Борщ'
end;
procedure TForm1.Button2Click(Sender: TObject);
begin
    Edit2.Text:='Пельмени'
end;
procedure TForm1.Button3Click(Sender: TObject);
begin
    Edit3.Text:='Сок'
end;
```



Form1

1 блюдо	Борщ
2 блюдо	Пельмени
3 блюдо	Сок



## Задача 5

Создайте приложение **Меню** с полем ввода **Edit** и тремя кнопками: **Первое**, **Второе** и **Компот**. При нажатии на кнопку **Первое** в поле ввода отображается первое блюдо, например , суп . При нажатии на кнопку **Второе** – второе блюдо, например , каша. При нажатии на кнопку **Компот** – в поле ввода появляется текст и компот!!!



## Задача 6

- Проанализируйте фрагмент программного кода приложения и определите **какие компоненты** использованы в приложении?
- **Сформулируйте задание** по созданию приложения с заданным программным кодом.



## Фрагмент программы (обработчик события)

implementation

{ \$R \*.dfm }

procedure TForm1.Button2Click(Sender: TObject);

begin

Form1.Caption:=Label1.Caption

end;

procedure TForm1.Button1Click(Sender: TObject);

begin

Form1.Caption:=Label2.Caption

end;



Заголовок 2



1

Заголовок 1

2

Заголовок 2



## Задача 6

Создайте приложение с двумя метками **Label** и двумя кнопками: **Заголовок1** и **Заголовок2**. При нажатии на кнопку **Заголовок1** заголовок формы меняется на содержимое первой метки. При нажатии на кнопку **Заголовок2** - на содержимое второй метки.





## Задача 7

- Проанализируйте фрагмент программного кода приложения и определите какие компоненты использованы в приложении?
- Сформулируйте задание по созданию приложения с заданным программным кодом.



## Фрагмент программы (обработчик события)

implementation

{ \$R \*.dfm }

procedure TForm1.Button1Click(Sender: TObject);

begin

    Label1.Caption:='КАЗНИТЬ, нельзя  
ПОМИЛОВАТЬ!!!'

end;

procedure TForm1.Button2Click(Sender: TObject);

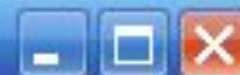
begin

    Label1.Caption:='Казнить нельзя,  
ПОМИЛОВАТЬ!!! '

end;



Царский приказ



Казнить

"Казнить нельзя, ПОМИЛОВАТЬ!!!"

Помиловать



## Задача 7

Создайте приложение Царский приказ с меткой **Label** и двумя кнопками:

**Казнить, Помиловать** . При нажатии на кнопку **Казнить** текст метки становится ***КАЗНИТЬ , нельзя помиловать!***, а при нажатии на кнопку **Помиловать** – ***Казнить нельзя, ПОМИЛОВАТЬ!***



# Обработка события OnClick и работа с компонентом Label

## Задание 8

Создайте приложение с меткой **Label** и двумя кнопками: **Красный**, **Синий**. При нажатии на кнопку **Красный** в метку должен отобразиться текст **красный** и цвет метки должен измениться на **красный**. При нажатии на кнопку **Синий** – текст **синий** и цвет метки, соответственно, **синий**.



Поме няй цвет



Синий

Красный

Красный



# Фрагмент программы (обработчик события)

implementation

{ \$R \*.dfm }

```
procedure TForm1.Button1Click(Sender: TObject);
```

```
begin
```

```
    Label1.Caption:='Синий';
```

```
    Label1.Color:=clBlue
```

```
end;
```

```
procedure TForm1.Button2Click(Sender: TObject);
```

```
begin
```

```
    Label1.Caption:='Красный';
```

```
    Label1.Color:=clRed
```

```
end;
```



# Обработка события OnClick

## Задание 9

Создайте приложение **Русско – английская** поддержка с двумя кнопками. Вначале на первой кнопке должно быть написано **English** , а на второй кнопке – **Русский**. При нажатии на первую кнопку названия кнопок меняются на **English** и **Russian** , а при нажатии на вторую кнопку – на **Английский** и **Русский** для первой и второй кнопки соответственно.





## Фрагмент программы (обработчик события)

implementation

{\$R \*.dfm}

procedure TForm1.Button1Click(Sender: TObject);

begin

    Button2.Caption:='Russian';

    Button1.Caption:='English';

end;

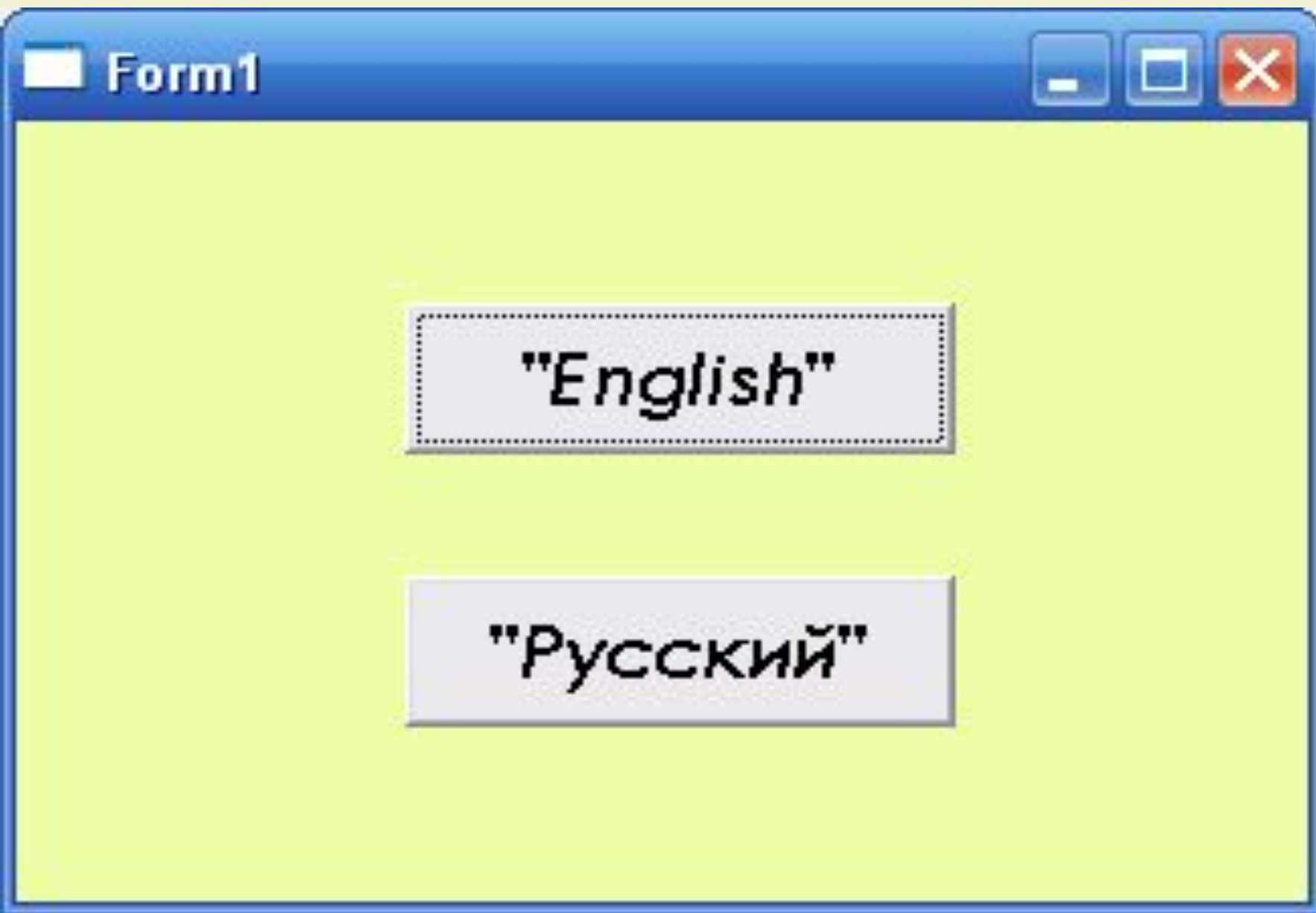
procedure TForm1.Button2Click(Sender: TObject);

begin

    Button1.Caption:='Английский';

    Button2.Caption:='Русский';

end;





# Обработка события OnClick и работа с компонентом Label

## Задание 10

Создайте приложение **Переключатель** с меткой **Label** и кнопкой . Вначале на кнопке должно быть написано **Включить**, а в метке – **Выключено**. Нажатие на кнопку приводит к появлению текста **Включено**, а надпись на кнопке меняется на **Выключить**. Повторное нажатие возвращает исходный текст **Выключено**, а надпись на кнопке становится **Включено**.

A screenshot of a Windows-style window titled 'Form1'. The window has a blue title bar with standard minimize, maximize, and close buttons. The main content area is white and contains three centered elements: the word 'Выключено' in blue text, a button with a dotted border containing the text 'Включить', and a button with a solid border containing the text 'Выход'.



## Фрагмент программы (обработчик события)

```
procedure TForm1.Label1Click(Sender: TObject);
begin
    if Label1.Caption='Выключить ' then
        begin
            Button1.Caption:='Выключить';
            Label1.Caption:='Включено';
        end
    else
        begin
            Button1.Caption:='Включить';
            Label1.Caption:='Выключено';
        end;
end;
```