

План

- ▶ Интегрированная среда Delphi.
- ▶ Управление проектом.
- ▶ Размещение компонентов на форме.
- ▶ Реакция на события.

1. Интегрированная среда Delphi

- ▶ Среда Delphi - это сложный механизм, обеспечивающий высокоэффективную работу программиста. Она реализуется несколькими одновременно раскрытыми на экране окнами.
- ▶ Высокопроизводительный компилятор в машинный код
- ▶ Объектно-ориентированная модель компонент
- ▶ Визуальное (а, следовательно, и скоростное) построение приложений из программных прототипов
- ▶ Масштабируемые средства для построения баз данных

Основные окна среды Delphi

- ✓ главное окно,
- ✓ окно Дерева объектов (Object Tree View)
- ✓ окно Инспектора объектов (Object Inspector)
- ✓ окно браузера
- ✓ окно формы
- ✓ окно кода программы

Delphi 7 - Project1

File Edit Search View Project Run Component Database Tools CnPack GExperts Window Help <None>

Standard Additional Win32 System Data Access Data Controls dbExpress BDE ADO WebServices InternetExpress Internet WebSnap Decision Cube Dialogs Win 3.1 Samples

Object TreeView

- Form1

Form1

Object Inspector

Form1 TForm1

Properties Events

| | |
|---------------|----------------|
| Action | |
| ActiveControl | |
| Align | alNone |
| AlphaBlend | False |
| AlphaBlendVal | 255 |
| Anchors | [akLeft,akTop] |
| AutoScroll | True |
| AutoSize | False |
| BiDiMode | bdLeftToRight |
| BorderIcons | [biSystemMenu] |
| BorderStyle | bsSizeable |
| BorderWidth | 0 |
| Caption | Form1 |
| ClientHeight | 462 |
| ClientWidth | 448 |
| Color | clBtnFace |

All shown

Unit1.pas

```
Unit1
TForm1(TForm)
Variables/Constants
Uses

unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes,
  Dialogs;
type
  TForm1 = class(TForm)
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
```

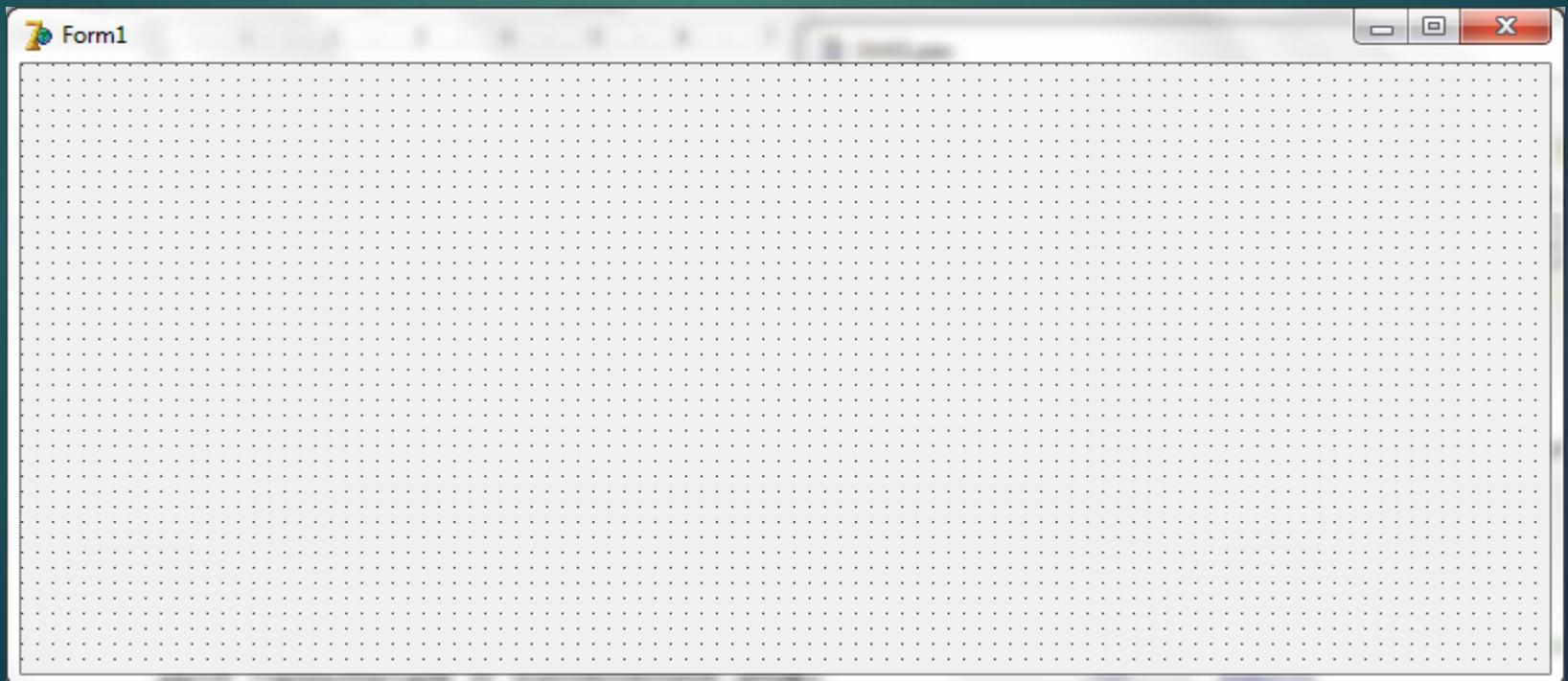
Windows taskbar showing: fixed - Total Comm..., Введение в програм..., Презентация1 - Pow..., Delphi 7

System tray: 22:41:54, 30 JAN

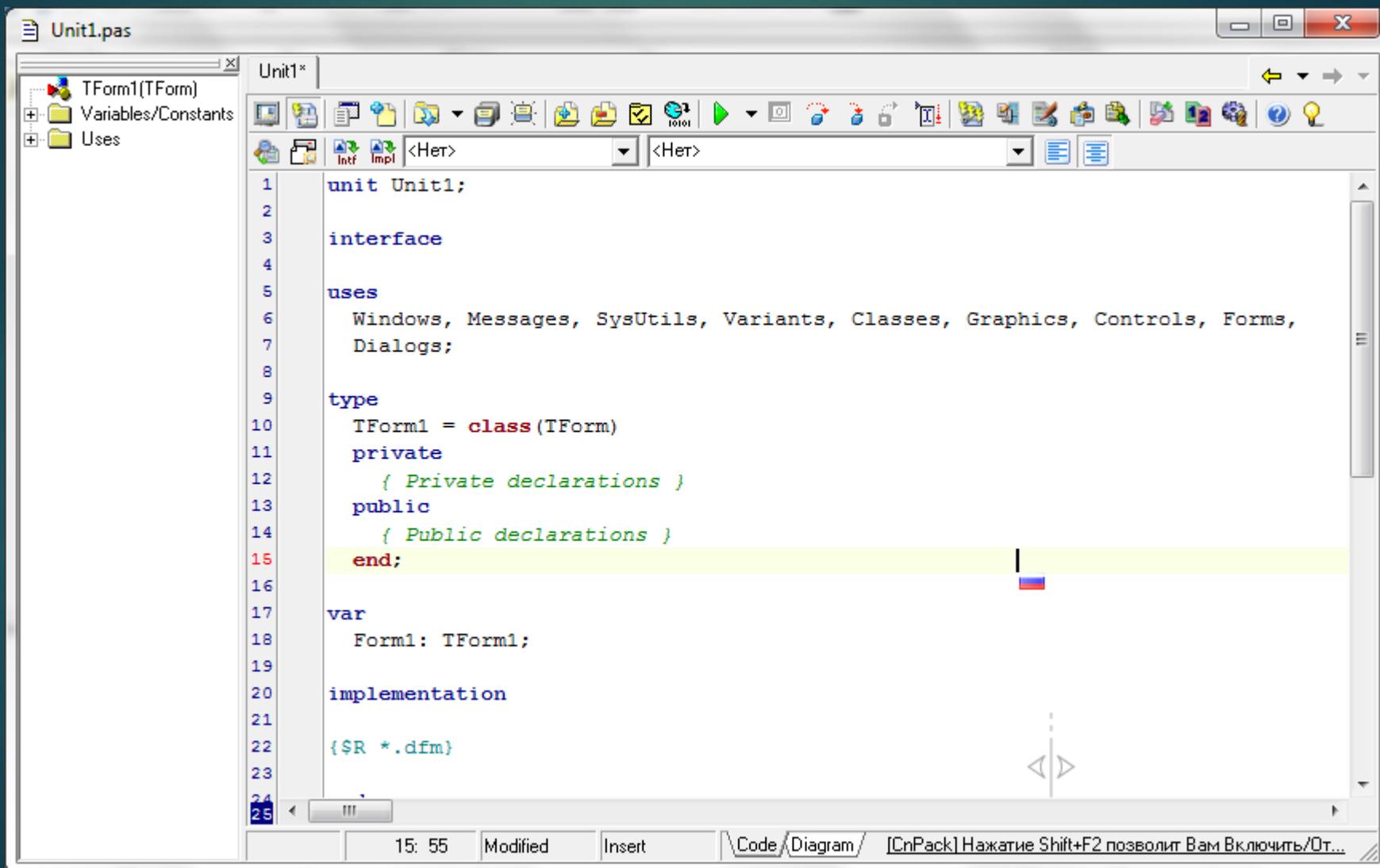
Главное окно Delphi. Здесь находится основное меню, различные панели инструментов и палитра компонентов, состоящая из множества вкладок.



Конструктор формы. Здесь мы визуально видим, как будет выглядеть форма программы, здесь мы будем создавать интерфейс, перенося на форму различные компоненты, и расставляя их таким образом, чтобы интерфейс выглядел привлекательным. Нам часто придется переключаться между конструктором форм и редактором кода, делается это клавишей F12.



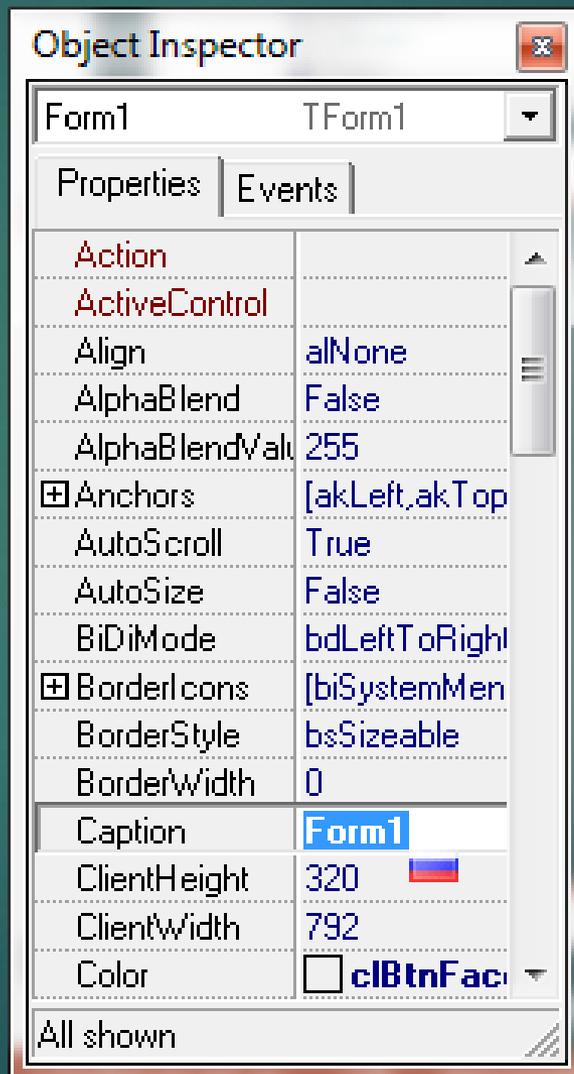
Редактор кода. Здесь мы видим исходный код программы, который создан самой Delphi. Тут же мы будем вводить наш собственный КОД.



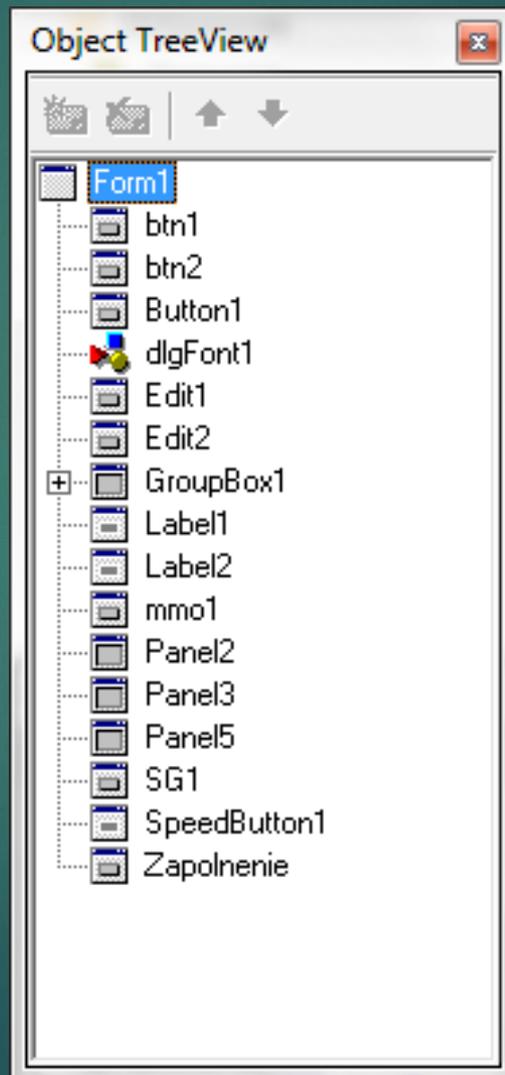
```
Unit1.pas
Unit1*
TForm1(TForm)
Variables/Constants
Uses
1  unit Unit1;
2
3  interface
4
5  uses
6    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
7    Dialogs;
8
9  type
10   TForm1 = class(TForm)
11     private
12       { Private declarations }
13     public
14       { Public declarations }
15   end;
16
17 var
18   Form1: TForm1;
19
20 implementation
21
22 {$R *.dfm}
23
24
25
```

15: 55 Modified Insert \Code/Diagram/ [CnPack] Нажатие Shift+F2 позволит Вам Включить/От...

Объектный инспектор. Он предназначен для управления объектами проекта и состоит из двух вкладок – Properties (Свойства) и Events (События).



Дерево объектов. Здесь мы видим, какой именно объект в данный момент является текущим. Это окно будет особенно полезно, когда на форме появится множество компонентов.



2. Управление проектом

► **Проект Delphi.** Проект — это набор исходных файлов создаваемого приложения

- Главный файл проекта *.DPR.
- модуль программы (unit), *.PAS
- Файл главной формы *.DFM
- Файл *.RES
- Файл *.DOF
- Файл *.CFG



После компиляции программы получаются файлы с расширениями:

- ✓ DCU - скомпилированные модули pas и dfm
- ✓ EXE - исполняемый файл
- ✓ ~PA, ~DP - backup файлы Редактора

Способы управления проектом

- ▶ **команды меню “File”**
- ▶ **Использование Менеджера Проектов View\Project Manager**
- ▶ **меню Project | Options**

Сохранение проекта.

- ▶ создать папку для программы.
- ▶ выбрать пункт меню File | Save All. Сохранить нужно будет два файла. Первый - модуль (unit), над которым Вы работали, второй - главный файл проекта, который "владеет" Вашей программой.

3. Размещение компонентов на форме

Размещение объекта на форме :

1. Выбрать и щелкнуть.
2. Выбрать, щелкнуть и нарисовать.
3. Двойной щелчок.

Выбор группы объектов:

1. Удерживая левую кнопку мыши и передвигая ее.
2. Удерживая нажатой клавишу <Shift>.

- ▶ С помощью обрамляющих черных квадратиков можно изменять размеры компонента.
- ▶ Все визуальные (видимые) компоненты имеют свойства
 - ▶ Left (Слева),
 - ▶ Top (Сверху),
 - ▶ Width (Ширина)
 - ▶ Height (Высота),
 - ▶ Visible (Видимость)
 - ▶ Color (Цвет)

4. РЕАКЦИЯ НА СОБЫТИЯ

- ▶ Функциональность программы определяется совокупностью ее реакций на те или иные события. Каждый компонент помимо свойств характеризуется также набором событий, на которые он может реагировать.
- ▶ Чтобы заставить программу реагировать на нажатие кнопки, необходимо написать на языке object pascal фрагмент программы, который называется *обработчиком события*

Delphi самостоятельно делает заготовку для процедуры обработчика события, например:

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

- ▶ Sender, передаваемый во все обработчики событий объект - источник события

- ▶ Существует другой способ – динамического [Изменения на этапе конструирования называются статическими, а в ходе прогона программы - динамическими.] изменения свойств на этапе прогона программы, например:

```
procedure TForm1.FormCreate(Sender: TObject);  
begin  
    Button1.Caption:= 'Звук';  
end;
```

Примеры событий

- ▶ `OnClick`
- ▶ `OnKeyUp`, `OnKeyPress`, `OnKeyDown`
- ▶ `OnEnter`, `OnExit`
- ▶ `OnMouseDown`, `OnMouseUp`

НЕКОТОРЫЕ ИТОГИ

- Процесс создания Delphi-программы разбивается на две фазы: фазу конструирования формы и фазу кодирования.

Конструирование формы осуществляется с помощью выбора компонентов из палитры и размещения их на форме.

- Программист может перемещать любой размещенный на форме компонент и изменять его размеры с помощью мыши.
- Чтобы придать компоненту нужные свойства, используется страница Properties Инспектора объектов.

Кодирование

Чтобы компонент мог откликаться на то или иное событие, программист должен создать обработчик события и указать его имя на странице Events Инспектора объектов.

- Обработчик события оформляется в виде процедуры, имеющей составное имя.
- Свойства компонента могут изменяться на этапе прогона программы.