Тема:

Интегрированная среда разработки приложений LAZARUS

Вопрос 1- Общая характеристика среды.

Lazarus – это бесплатная и свободная графическая среда разработки программного обеспечения (IDE) на языке *Object Pascal* для компилятора *Free Pascal Compiler* (*FPC*), также свободного программного обеспечения.

Оба продукта распространяются под лицензией *GPL* (*General Public License* – универсальная общественная лицензия из *GNU* обширной коллекции свободного программного обеспечения).

В основе LAZARUS лежит объектно-ориентированное программирование (ООП) – это метод программирования, при использовании которого главными элементами программ являются объекты.

В основе ООП 3 понятия:

инкапсуляция: объединение данных с процедурами и функциями в рамках единого целого – объекта;

наследование: возможность построения иерархии объектов, с использованием наследования их характеристик;

полиморфизм: задание одного имени действию, которое передается вверх и вниз по иерархии объектов, с реализацией этого действия способом, соответствующим каждому объекту в иерархии - объекты разных классов могут по-разному выполнять один и тот же метод.


КЛАСС
ОБЪЕКТ
ОБЪЕКТ
ОБЪЕКТ
...
...
...
...
...
...
...
СВОЙСТВА
МЕТОДЫ
СОБЫТИЯ

Возможности и достоинства LAZARUS

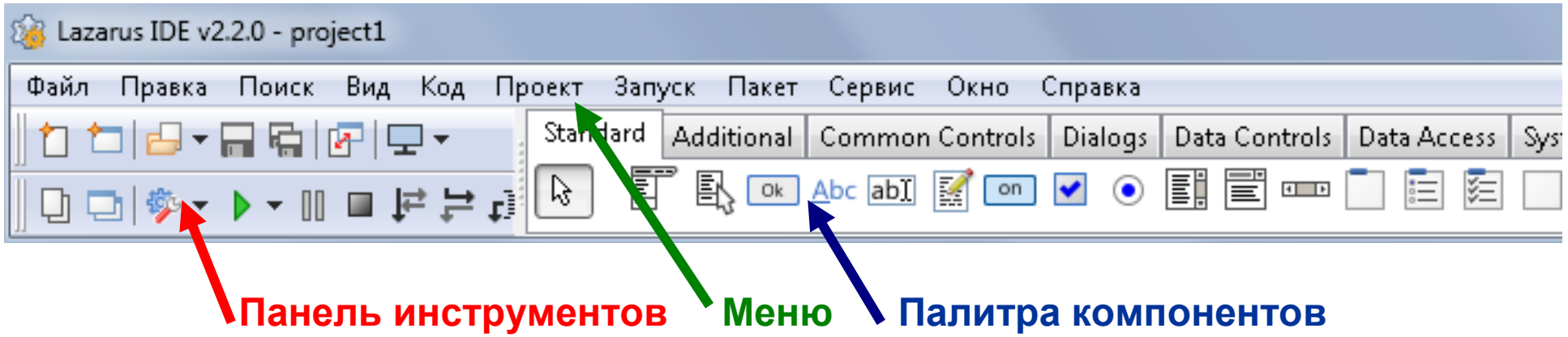
- ✓Поддерживает преобразование проектов Delphi
- ✓Реализован основной набор элементов управления
- ✓Редактор форм и инспектор объектов максимально приближены к Delphi
- ✓Интерфейс отладки (используется внешний отладчик GDB)
- ✓Полностью юникодный (UTF-8) интерфейс и редактор и поэтому отсутствуют проблемы с кодом, содержащего национальные символы
- ✓Мощный редактор кода, включающий систему подсказок, гипертекстовую навигацию по исходным текстам, автозавершение кода,
- ✓Поддержка множества типов синтаксиса Pascal: Object Pascal, Turbo Pascal, Mac Pascal, Delphi (поддерживаются со стороны компилятора)
- ✓Имеет собственный формат управления пакетами,
- ✓Автосборка самого себя (под новую библиотеку виджетов) нажатием одной кнопки
- ✓Поддерживаемые для компиляции ОС: Linux, Microsoft Windows (Win32, Win64), Mac OS X, FreeBSD, WinCE, OS/2

Вопрос 2.Палитра компонентов.

Среда разработки **LAZARUS** базируется на библиотеке визуальных компонентов *Lazarus Component Library*, которая содержит достаточное число компонентов, позволяющих создавать формы при помощи визуального проектирования графического интерфейса пользователя

При запуске *Lazarus* после установки на рабочем столе появляется набор несвязанных между собой окошек.

Первое окно, расположенное в самом верху рабочего стола, имеет название *Lazarus Editor – project.* **Это главное окно** управления проектом и оно содержит «Главное меню»,«Палитру Компонентов» (*Component Palette*) и панель инструментов, дублирующую меню

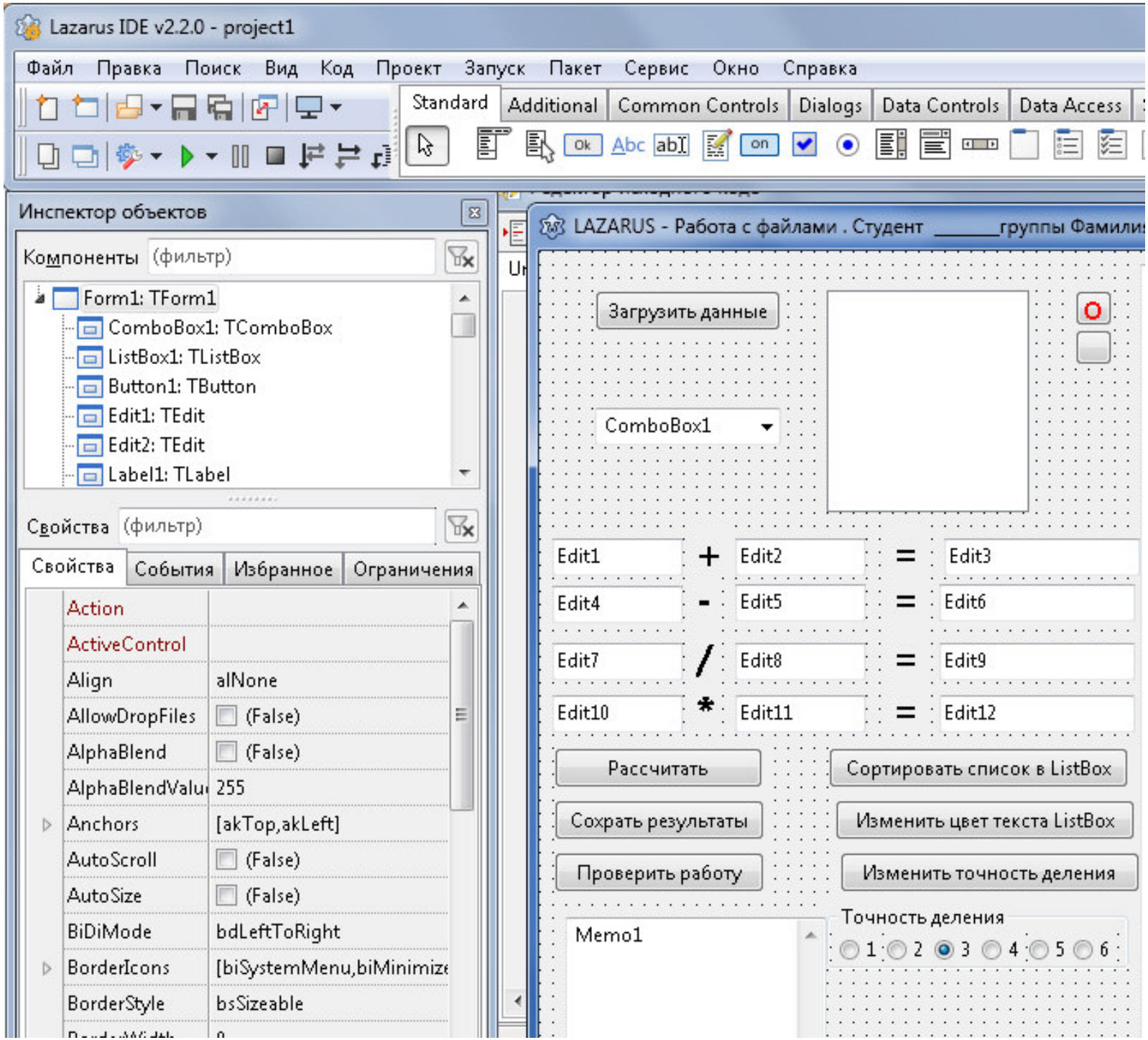


Вопрос 3. Инспектор объектов

(Свойства компонентов Основные события компонентов).

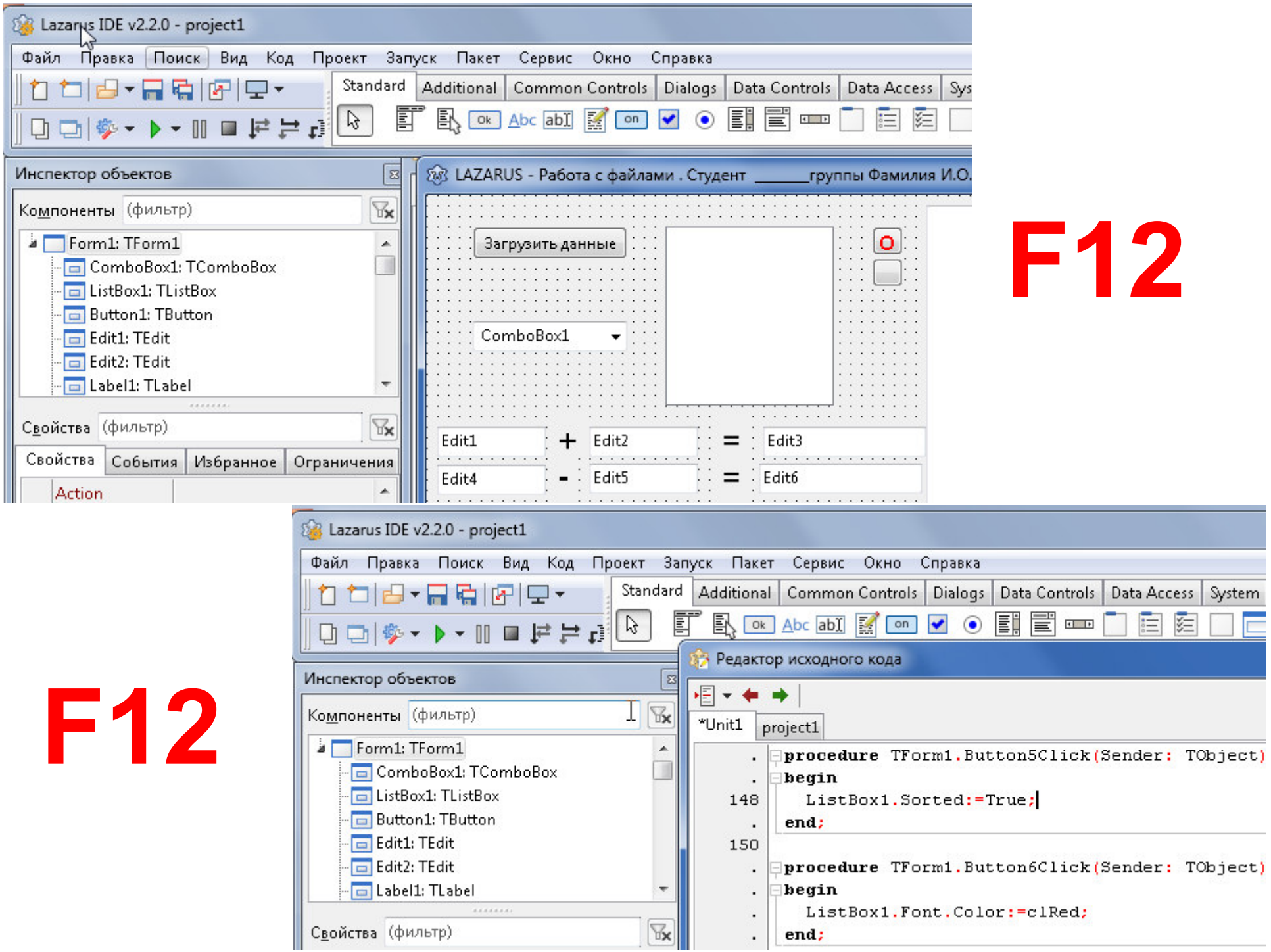
Под главным окном ***Lazarus*** слева располагается окно «Инспектора Объектов», а справа от него – «Редактор Исходного кода» (*Lazarus Source Editor*).

Может присутствовать и другое окно, озаглавленное *Form1*, расположенное поверх «Редактора Исходного Кода».

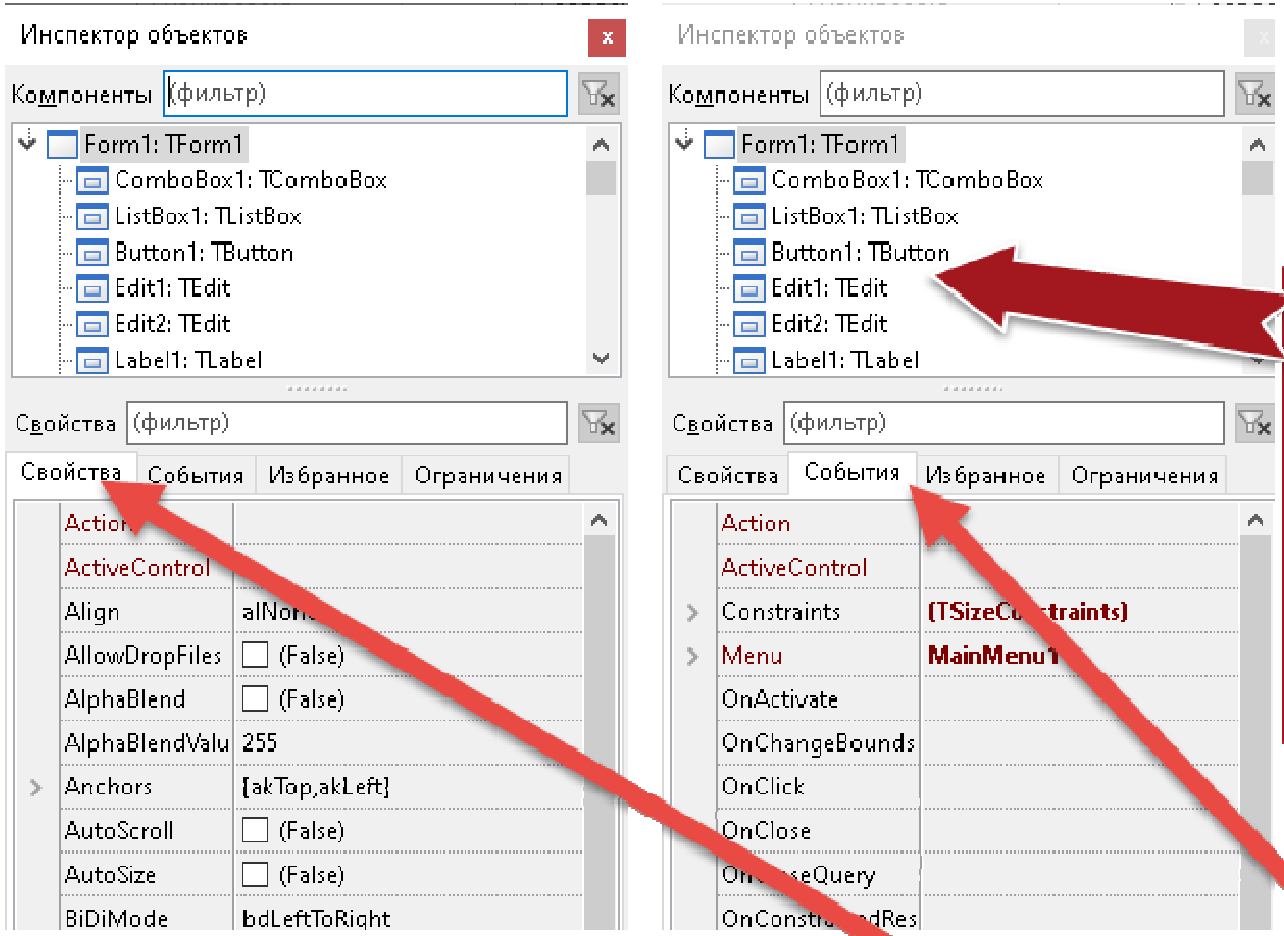

Lazarus IDE v2.2.0 - project1
Файл Правка Поиск Вид Код Проект Запуск Пакет Сервис Окно Справка
Standard Additional Common Controls Dialogs Data Controls Data Access
Инспектор объектов
Компоненты (фильтр)
Form1: TForm1
ComboBox1: TComboBox
ListBox1: TListBox
Button1: TButton
Edit1: TEdit
Edit2: TEdit
Label1: TLabel
Свойства (фильтр)
Свойства События Избранное Ограничения
Action
ActiveControl
Align alNone
AllowDropFiles (False)
AlphaBlend (False)
AlphaBlendValu 255
Anchors [akTop,akLeft]
AutoScroll (False)
AutoSize (False)
BiDiMode bdLeftToRight
BorderIcons [biSystemMenu,biMinimize
BorderStyle bsSizeable
LAZARUS - Работа с файлами . Студент _____группы Фамили
Загрузить данные
ComboBox1
Edit1 + Edit2 = Edit3
Edit4 - Edit5 = Edit6
Edit7 / Edit8 = Edit9
Edit10 * Edit11 = Edit12
Рассчитать
Сортировать список в ListBox
Сохрать результаты
Изменить цвет текста ListBox
Проверить работу
Изменить точность деления
Memo1
Точность деления
1 2 3 4 5 6

Если его не видно, то можно нажав клавишу *F12*, переключится между «Редактором Исходного Кода» и «Окном Формы».

Окно формы – это место, где разрабатывают графический интерфейс программы, а в «Редакторе Исходного Кода» отображается разрабатываемый *Pascal* - код приложения

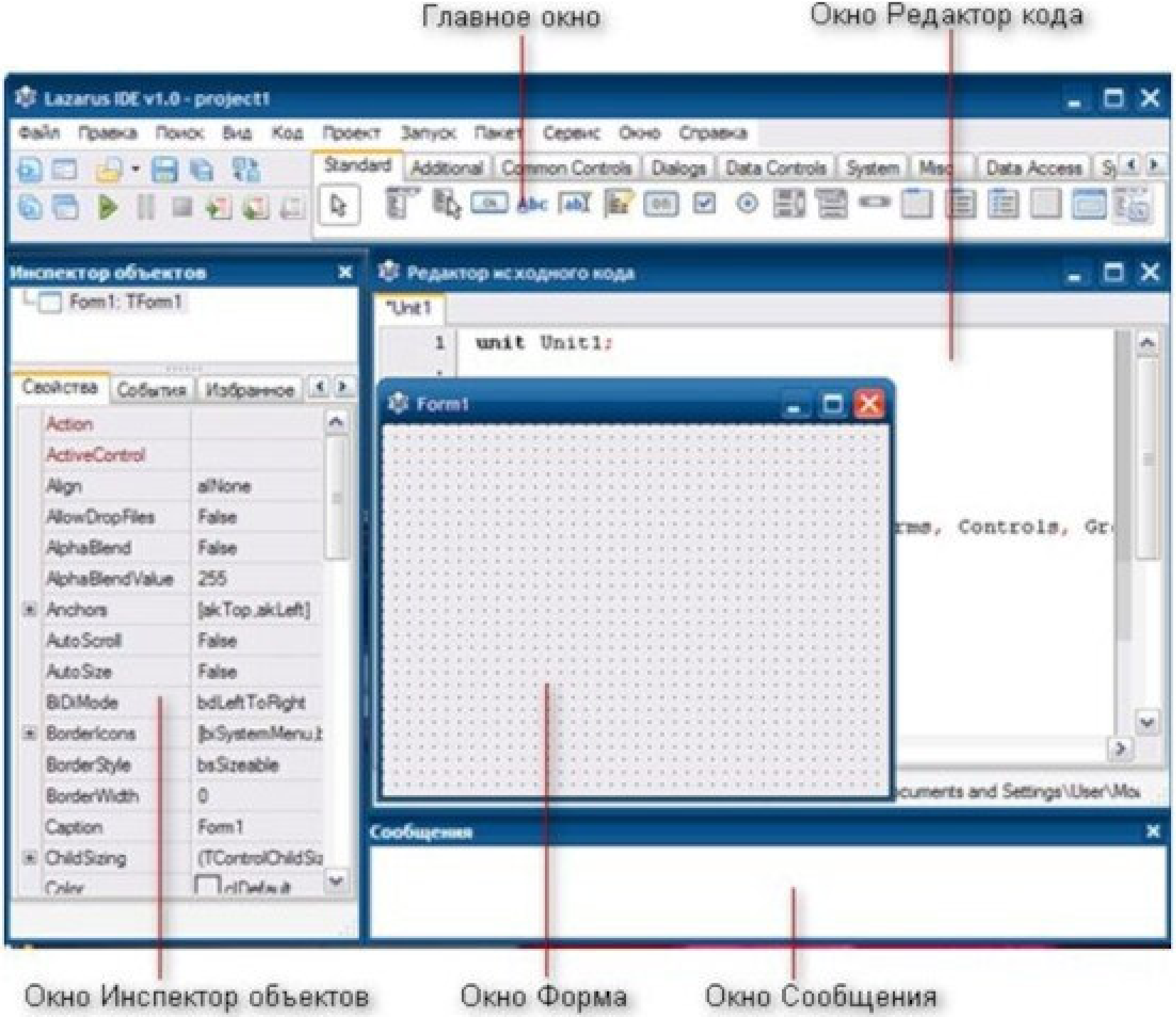

Lazarus IDE v2.2.0 - project1
Файл Правка Поиск Вид Код Проект Запуск Пакет Сервис Окно Справка
Standard Additional Common Controls Dialogs Data Controls Data Access
Инспектор объектов
Компоненты (фильтр)
Form1: TForm1
ComboBox1: TComboBox
ListBox1: TListBox
Button1: TButton
Edit1: TEdit
Edit2: TEdit
Label1: TLabel
Свойства (фильтр)
Свойства События Избранное Ограничения
Action
LAZARUS - Работа с файлами . Студент _____группы Фамилия И.О.
Загрузить данные
ComboBox1
Edit1 + Edit2 = Edit3
Edit4 - Edit5 = Edit6
F12
Lazarus IDE v2.2.0 - project1
Файл Правка Поиск Вид Код Проект Запуск Пакет Сервис Окно Справка
Standard Additional Common Controls Dialogs Data Controls Data Access System
Инспектор объектов
Компоненты (фильтр)
Form1: TForm1
ComboBox1: TComboBox
ListBox1: TListBox
Button1: TButton
Edit1: TEdit
Edit2: TEdit
Label1: TLabel
Свойства (фильтр)
Редактор исходного кода
*Unit1 project1
procedure TForm1.Button5Click(Sender: TObject)
begin
ListBox1.Sorted:=True;
end;
procedure TForm1.Button6Click(Sender: TObject)
begin
ListBox1.Font.Color:=clRed;
end;
F12

Окно Инспектор объектов



Объекты используемые в проекте

Вкладки объекта: Свойства, События, Избранное, Ограничения


Главное окно
Окно Редактор кода
Lazarus IDE v1.0 - project1
Файл Правка Поиск Вид Код Проект Запуск Пакет Сервис Окно Справка
Standard Additional Common Controls Dialogs Data Controls System Misc Data Access
Инспектор объектов
Form1: TForm1
Свойства События Избранное
Action
ActiveControl
Align alNone
AllowDropFiles False
AlphaBlend False
AlphaBlendValue 255
Anchors [akTop,akLeft]
AutoScroll False
AutoSize False
BiDiMode bdLeftToRight
BorderIcons [biSystemMenu,b
BorderStyle bsSizeable
BorderWidth 0
Caption Form1
ChildSizing (TControlChildSi
Редактор исходного кода
Unit1
1 unit Unit1;
Form1
Controls, Gr
Documents and Settings\User\Mo
Сообщения
Окно Инспектор объектов
Окно Форма
Окно Сообщения

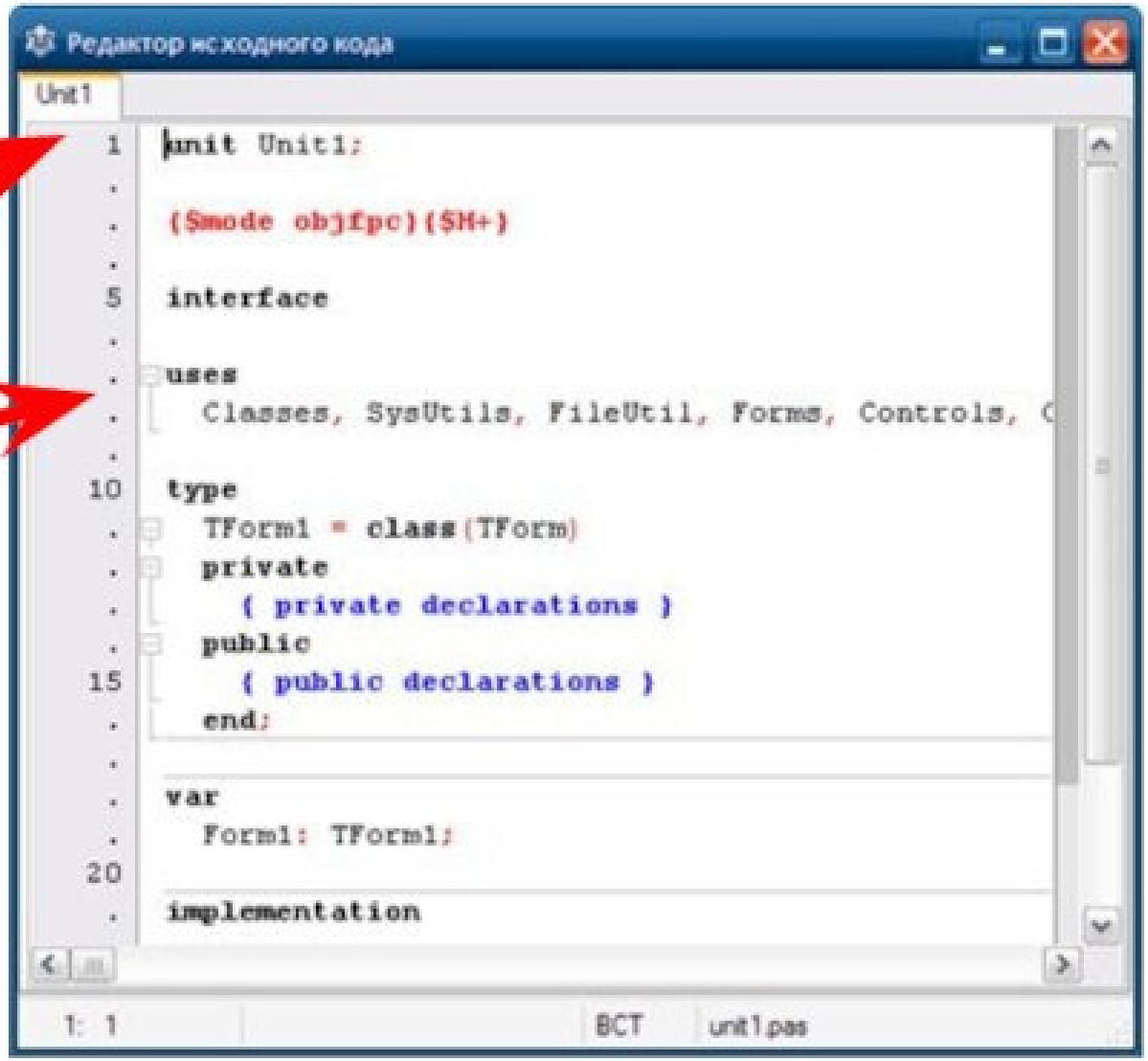
Окно Редактора кода

На момент первого запуска оно имеет заголовок Unit1

Окно для написания программного кода программы

Текст программы разбивается на части – процедуры и функции.

Основную работу программист производит именно здесь.



Главное текстовое меню содержит следующие пункты: *File*, *Edit*, *Search*, *View*, *Project*, *Run*, *Components*, *Tools*, *Environment*, *Windows*, *Help*.

И в локализованном (русифицированном) варианте – Файл, Правка, Поиск, Вид, Проект, Запуск, Пакет, Сервис, Окружение, Окно, Справка.

После подготовки кода программы выполняют запуск компилятора и сборки проекта (см. панель инструментов или меню).

Собрать (Build): запускается сборка (т. е. компиляция) любых файлов проекта, которые были изменены со времени последней сборки.

Собрать все (Build all): запускается сборка всех файлов проекта, независимо от наличия изменений

Вопрос 2 - Палитра компонентов.

Библиотека визуальных компонентов (*Visual Component Library* **(*VCL*)**) содержит большое количество классов **(тип = класс)**, предназначенных для быстрой разработки приложений.

Библиотека написана на *Object Pascal*.

В ***VCL*** содержатся невизуальные и визуальные компоненты, а также другие классы, начиная с абстрактного класса ***TObject*.**

При этом **все компоненты являются классами**, но не все классы являются компонентами.

Все классы ***VCL*** расположены на определенном уровне иерархии и образуют дерево (иерархию) классов.

Знание происхождения объекта оказывает значительную помощь при его изучении, так как потомок наследует все элементы объекта-родителя.

Так, если свойство *caption* принадлежит классу ***TControl***, то это свойство будет и у его потомков, например у классов ***TButton*** и ***TCheckBox*** и у **компонентов** – кнопки *Button* и независимого переключателя *CneckBox* соответственно.

Вкладки (их имена достаточно понятны и не требуют разъяснений): *Standart*; *Additional*; *Common Controls*; *Dialogs*; *Misc*; *Data Controls*; *Data Access*; *System*; *SynEdit*.


Класс
Группа объектов одного типа «Кнопки»
Объект
Что?
Пример «Кнопка»
Свойства
Какой?
Пример: размеры, цвет …
Метод
Что делает?
можно нажать, перетащить…
Событие
Когда происходит?
При щелчке ЛКМ…

Создание графического интерфейса проекта

Установите новые значения для свойств:

Объект	Имя	Свойство	Значение
Форма	Form1	Caption	Моя первая программа
Надпись	Label1	Caption	Пустая строка
Кнопка	Button1	Caption	Вывести сообщение

Создание графического интерфейса проекта

Установите для метки Lable1 свойства:

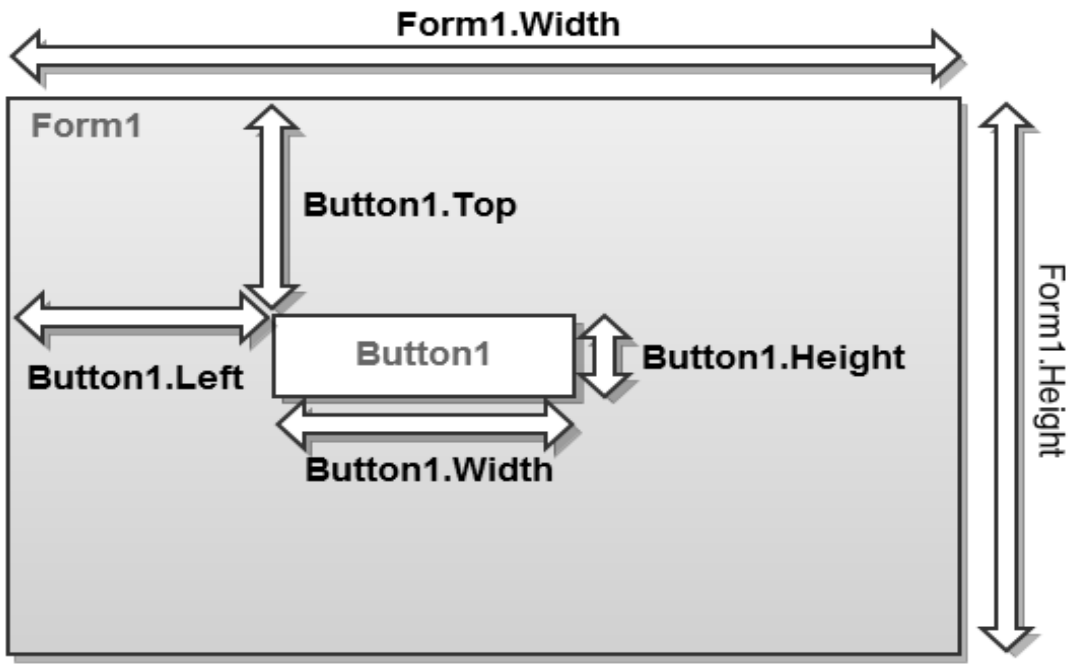
Объект	Имя	Свойство	Значение
Надпись	Label1	Font	Arial
Надпись	Label1	Size	20
Надпись	Label1	Style	Полужирный

Существует **несколько способов определения и изменения свойств компонента**.

1. Если компонент разместить на дизайнере формы и посмотреть на свойства в инспекторе объектов, то можно наблюдать за изменением некоторых свойства при перемещении компонента по форме.

2. Кроме того, с помощью инспектора объектов, можно самостоятельно и независимо выбрать значение, связанное со свойством объекта, и ввести новое значение.

3. Также можно явно изменить свойства объекта, если ввести новое значение свойства в редакторе исходного кода. Новое значение также отобразится в инспекторе объектов.



Несколько основных свойств

Name - имя объекта (текст).

Caption - заголовок (текст). Надпись отображаемая на объекте.

Width - ширина объекта по горизонтали (целое число)

Height - высота объекта по вертикали (целое число)

Top - расстояние от верхнего края объекта до верхнего края родительского объекта (целое число).

Left - расстояние от левого края объекта до левого края

Свойство или процедура компонента	Применение	Типы данных
Memo1.Lines[n]	обращение к n-ой строке	результат String n — integer
Memo1.Lines.Count	Количество строк	integer
Memo1.Lines.Clear	Очистка содержимого	
Memo1.Lines.Add(S)	Добавление строки S	S — String
Memo1.Lines.Delete(n)	Удаление строки n	n — integer
Memo1.Lines.Insert(n,S)	Добавление строки S в позицию n	n — integer S — String
Memo1.Lines.LoadFromFile(P)	Загрузка текста из фала	P — String
Memo1.Lines.SaveToFile(P)	Сохранение в файл (P — путь к файлу)	P — String
Memo1.Lines.Delete(n)	Удаление строки n	n — integer
Memo1.Lines.Delete(n)	Удаление строки n	n — integer

Вопрос 3 - Основные события компонентов.

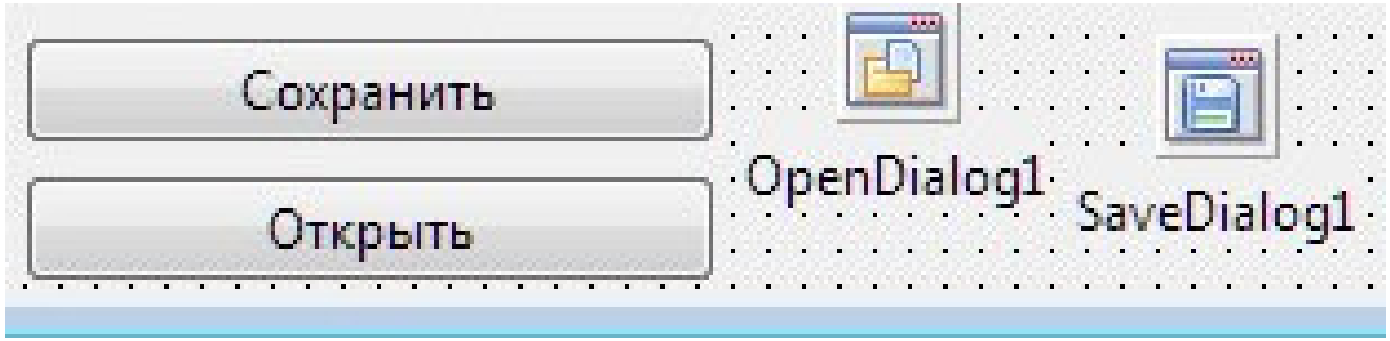
Многие действия перечислены на вкладке «События» Инспектора объектов (*Object Inspector*).

При выборе записи в списке *ComboBox*, появляется выпадающий список, показывающий любые действия, которые уже определены в *IDE*, и позволяет вам выбрать одно из них, чтобы оно было связано с этим событием.

Для большинства элементов управления достаточно обеспечить написание кода для *OnClick*, хотя для более сложных объектов может быть необходимо также обеспечить действие *OnEntry* (когда курсор мыши входит в управления и передает ему фокус) и *OnExit* (когда курсор мыши покидает элемент управления) или написать обработчик событий для *OnChange* или *OnScrol*.

Всплывающее меню, которое появляется при нажатии правой кнопкой мыши на объекте в дизайнере форм, содержит в качестве своего первого пункта «Создать событие по умолчанию», и выбор этой опции, будет иметь тот же эффект, что и выбор с многоточием в инспекторе объектов по умолчанию для данного события. Как правило, на событие *OnClick* средствами *IDE* будет создано поле со стандартным программным кодом в редакторе.

Добавляем события нажатия кнопок «Сохранить», «Открыть», выполняющие открытие и сохранение записей.




```
if SaveDialog1.Execute then
Memo1.Lines.SaveToFile(SaveDialog1.FileName);
```

```
if OpenDialog1.Execute then
Memo1.Lines.LoadFromFile(OpenDialog1.FileName
);
```

OnChange. Действия, которые будут приняты, если любое изменение будет обнаружено.
OnClick. Предлагаемое действие, когда нажата кнопка мыши.
Click. Метод для эмуляции в коде действия однократного нажатия на элементе управления.
OnDragDrop. Предлагаемое решение во время события *OnDragDrop*.
OnEditingDone. Предлагаемое решение, когда пользователь закончил все изменения или модификации объекта.
OnEntry. Действие в результате события, когда курсор мыши входит в область, занимаемую объектом, и переносит фокус на этот объект.

OnExit. Предлагаемое действие, когда мышь перемещается из области объекта с потерей фокуса на объекте.

OnKeyPress. Действие, по событию соответствующее любому нажатию кнопки.

OnKeyDown. Действия, активирующиеся, когда клавиша нажата, а фокус находится в элементе управления.

OnKeyUp. Действия, выполняющиеся, когда клавиша освобождена, в то время как фокус находится в этом элементе управления.

OnMouseMove. Действия, реализующиеся, когда курсор мыши перемещается над элементом управления, находящемся в фокусе.

OnMouseDown. Действия, выполняющиеся, когда кнопка мыши нажата и в то же время находится в элементе управления, находящемся в фокусе.
OnMouseUp. Действия, выполняющиеся, когда кнопка мыши не нажата, а курсор находится над этим элементом управления. Означает, что кнопка мыши была ранее нажата и была освобождена. Случай, когда курсор входит в область элемента управления, но кнопка мыши еще не была нажата, определен событиями *OnEntry* или *OnMouseEnter*.
OnResize. Действие, совершаемое при изменении размеров элемента управления.

Вопрос 4 - Состав проекта.

Порядок создания приложения

1. Создать новый проект:

Файл – Создать – Приложение – ОК

Сохранить проект в отдельной папке!

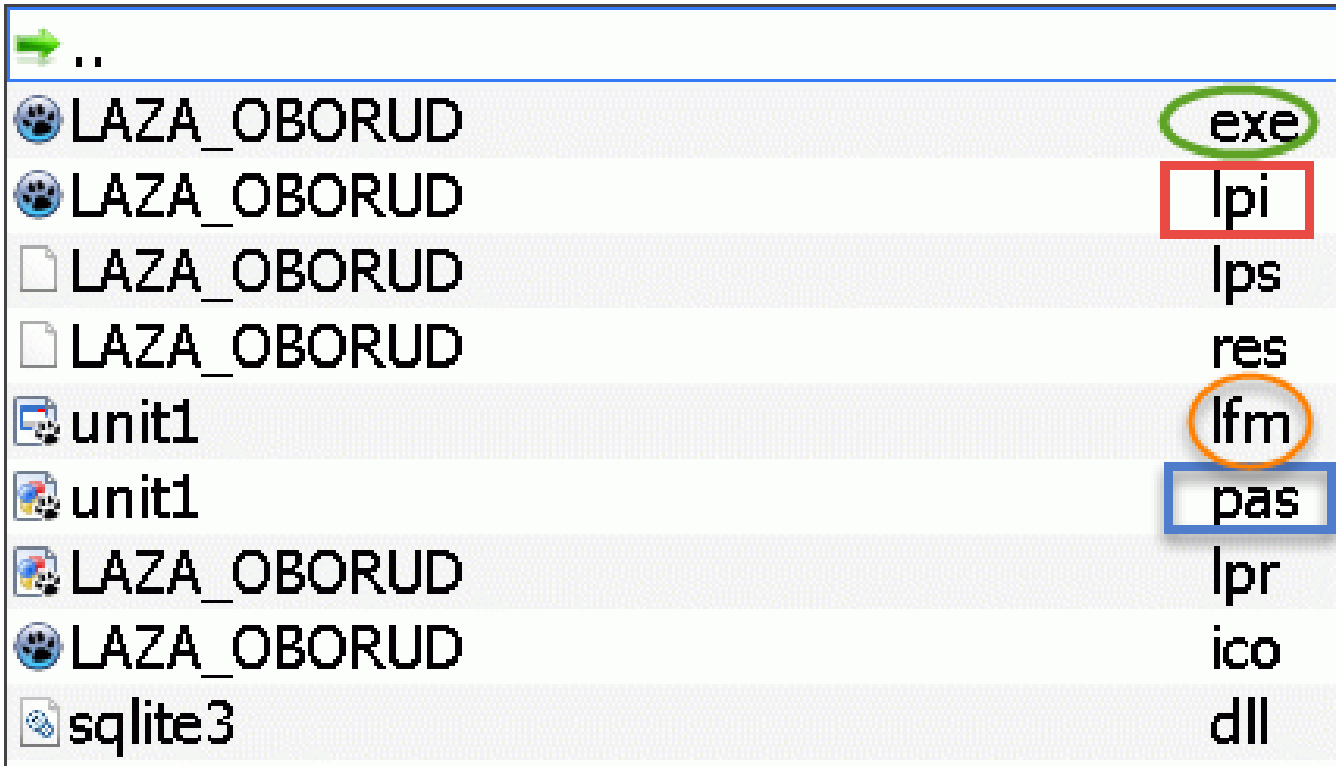
2. Создать графический интерфейс проекта;

3. Написать программный код;

4. Отладить программу.

I. Создание нового проекта

1. Открываем Лазарус.
Выполняем команду: Файл – Создать – Приложение – ОК.
Сохраняем проект в СПЕЦИАЛЬНО созданную папку!
Для этого: Файл – сохранить как – выбрать папку
2. Проверим, что сохранены все файлы.
3. Закроем проект щелчком по крестику в главном окне программы.
4. Найдем в папке файл project1.lpi – это основной файл проекта. Запустим его.



Проект также включает:

файл *.*exe* – основной исполняемый файл программы;

файл *.*lpi* – основной файл проекта *Lazarus*;

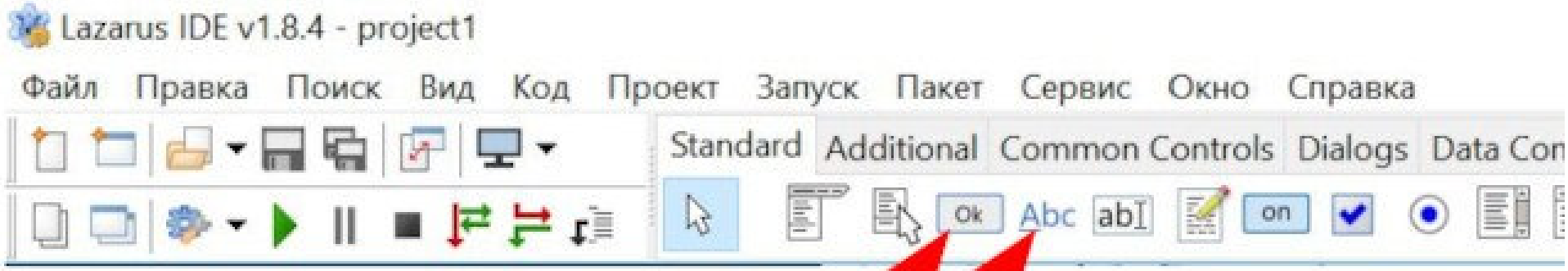
файл *.*lpr* – исходный код основной программы;

файлы *.*pas* – модули, содержащие коды форм;

файлы *.*lfm* – файлы, в которых хранятся описания форм модулей;

файл *.*lrs* – автоматически генерируемый файл ресурсов.

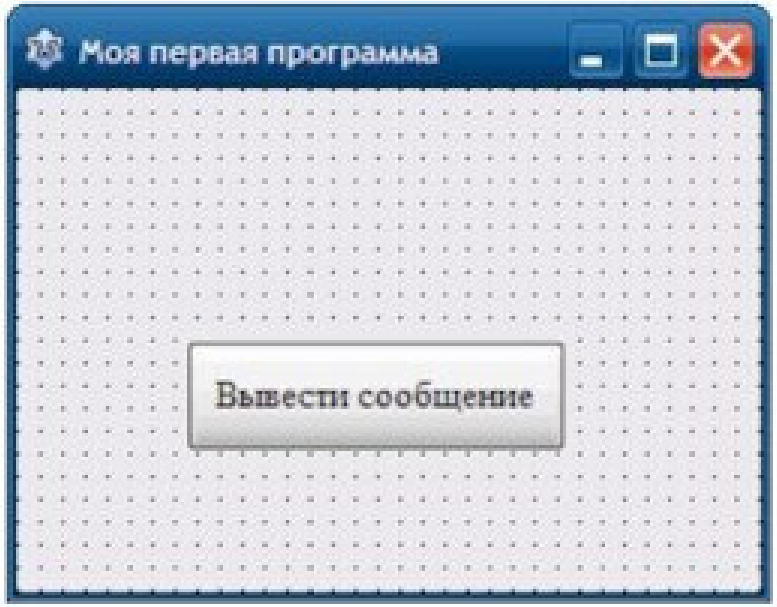
II. Создание графического интерфейса проекта



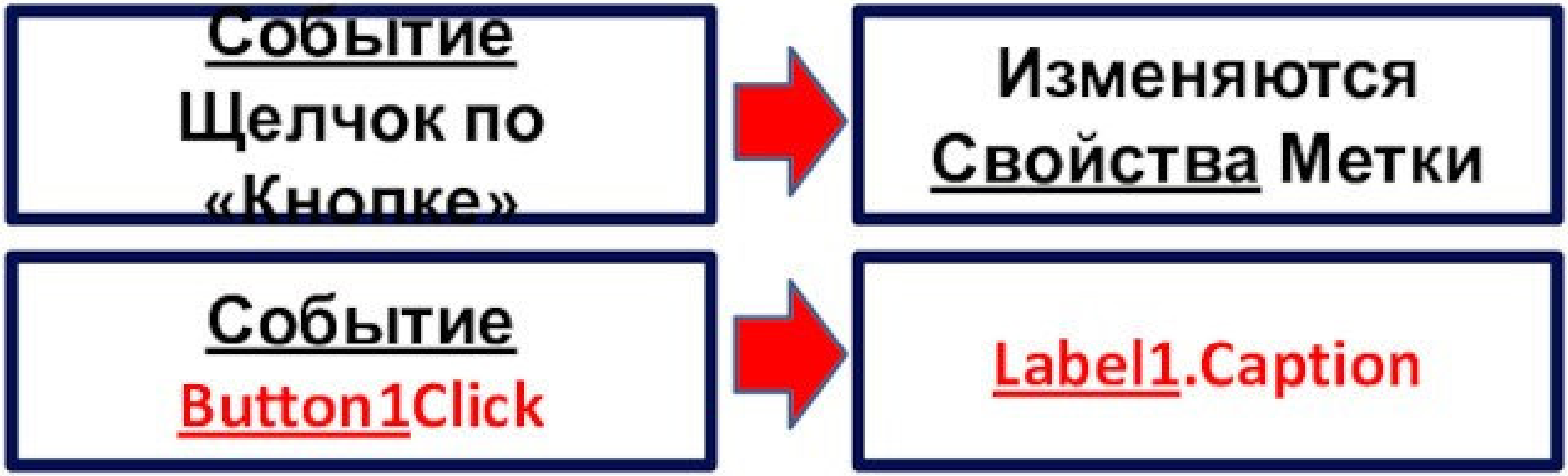
Разместите на форме компоненты:

Кнопку (Button)

Надпись (Label)

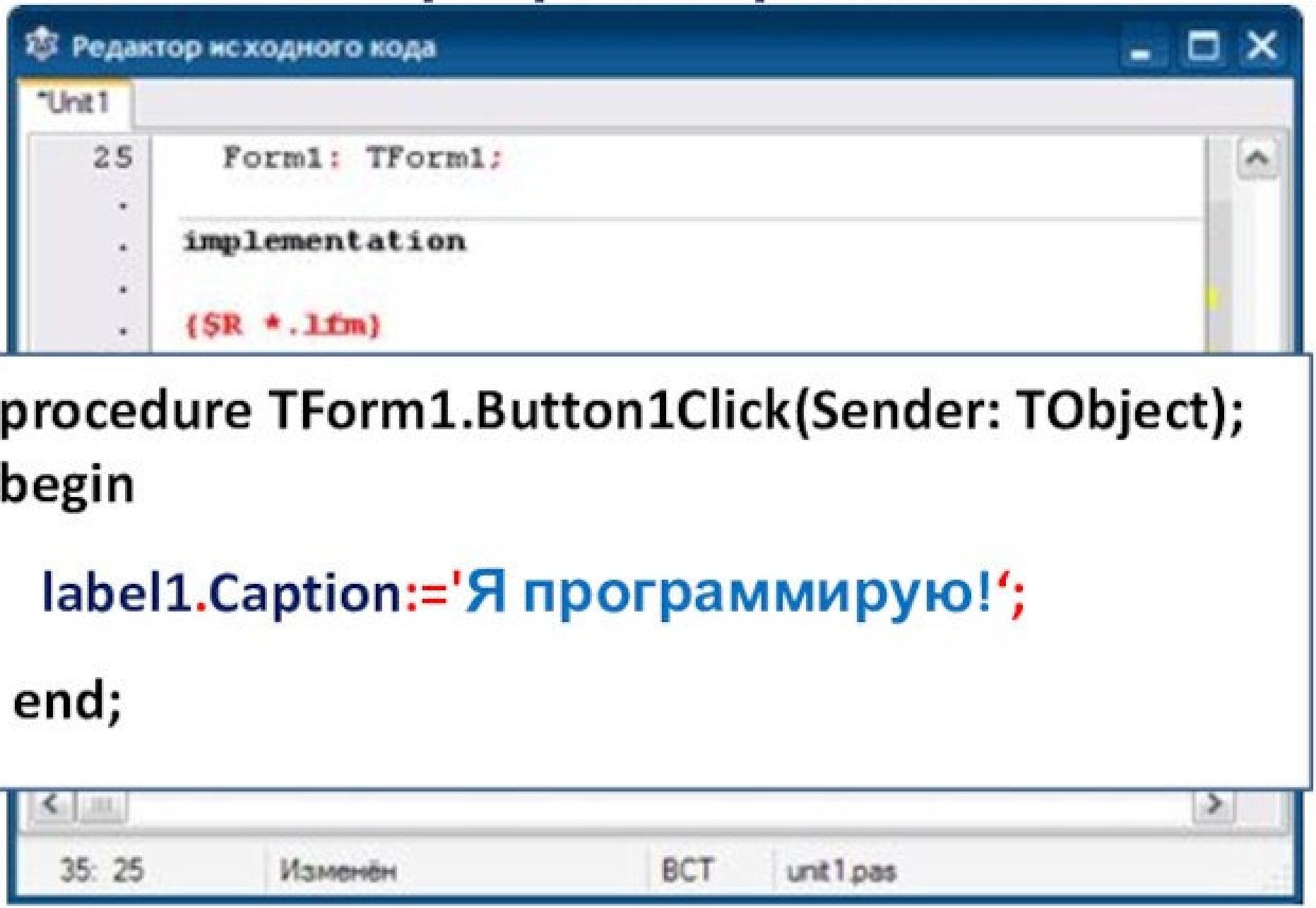


III. Программирование




```
label1.Caption:='Я программирую!';
```

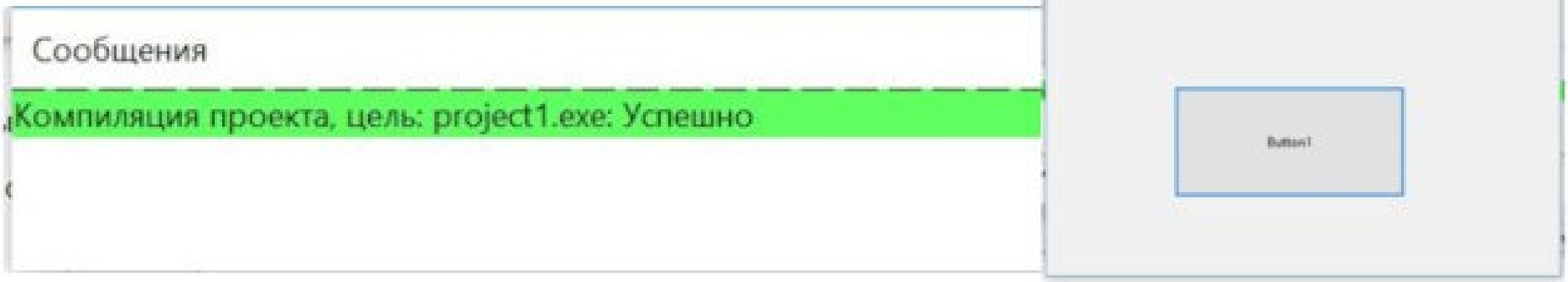
III. Программирование




```
procedure TForm1.Button1Click(Sender: TObject);
begin
  label1.Caption:='Я программирую!';
end;
```

IV. Отладка программы

- щелкнуть по кнопке **Запустить** на панели инструментов;
- выбрав команду **Запуск - Запустить** в главном меню;
- нажав клавишу **<F9>**



В папке проекта создается **EXE**