

ТЕМА УРОКА:

ИНТЕГРИРОВАННАЯ СРЕДА РАЗРАБОТКИ

LAZARUS



учитель информатики Бахтыбекова Г.М.

ТЕСТ ПО ТЕМЕ "ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ"

1. Когда появились первые языки программирования?

- A. В 40-е года XX века.
- B. В 50-е года XX века.
- C. В 60-е года XX века.
- D. В 70-е года XX века.

2. Визуальное программирование относится к :

- A. 2 поколению:
- B. 3 поколению:
- C. 4 поколению:
- D. 5 поколению.

3. Укажите верный перечень принципа объектно-ориентированного программирования;

- A. инкапсуляция, наследование, полиморфизм;
- B. Наследование, использование функций, полиморфизм;
- C. Инкапсуляция, наследование, полиморфизм, , структурность;
- D. Инкапсуляция , наследование, полиморфизм, модульность.

4. На какой технологии построения алгоритмов основано структурное программирование?

- A. «снизу-вверх»:
- B. «сверху- вниз»:
- C. Последовательного выполнения:
- D. модульного построения.

5. Как называется метод программирования, в котором используются объекты?

- A. модульный;
- B. Визуальный;
- C. Структурный;
- D. объективно-ориентированный.

6. Какие виды программ не присущи структурному программированию?

- A. процедуры;
- B. Функции;
- C. Модули;
- D. Методы.

- **7. Языки программирования 2-го поколения характеризует:**
- А. система машинных команд конкретного компьютера;
- В. Система обозначений для абстрактного описания вычислений
- С. Система визуального проектирования пользовательских приложений, выполняющая автоматическую генерацию соответствующих программ;
- D. система мнемоник для обозначения машинных команд конкретного семейства компьютеров.

- **8. В объектно-ориентированном программировании каждый объект по отношению к своему классу является...**
- А. кодом;
- В. Экземпляром;
- С. Функцией;
- D. фрагментом.

- **9. Не является принципом объектно-ориентированного программирования...**
- А. инкапсуляция;
- В. Наследования;
- С. Полиморфизм;
- D. использования функций.



Визуальное программирование –
способ создания программы
путем манипулирования
графическими объектами
вместо написания ее текста.

Lazarus представляет собой свободную среду быстрой разработки программного обеспечения для компилятора Free Pascal, аналогичную Delphi.

Данный проект базируется на оригинальной кроссплатформенной библиотеке визуальных компонентов Lazarus Component Library/

Кроссплатформенное программное обеспечение – программное обеспечение, работающее более чем на одной аппаратной платформе и/или операционной системе.

Основные сведения:

Автор	Cliff Baeseman, Shane Miller, Michael A. Hess и др.
Разработчики	сообщество
Написана на	Object Pascal
Интерфейс	графический
Операционная система	Linux, Mac OS X и др. UNIX-подобные, Windows
Языки интерфейса	Русский и ещё 37 языков
Последняя версия	1.0 (28 август 2012)
Сайт	lazarus.freepascal.org

Функции и достоинства

- ✓ 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 отсутствие проблем с кодом, содержащего национальные символы
- ✓ Мощный редактор кода, включающий систему подсказок, гипертекстовую навигацию по исходным текстам, автозавершение кода и рефакторинг
- ✓ Поддержка множества типов синтаксиса Pascal: Object Pascal, Turbo Pascal, Mac Pascal, Delphi (поддерживаются со стороны компилятора)
- ✓ Имеет собственный формат управления пакетами
- ✓ Авто сборка 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

Недостатки

- ✓ Нет полной совместимости с Delphi (хотя в отличие от Delphi предоставляет возможность создавать более кроссплатформенные приложения)
- ✓ Отсутствие полной документации, хотя она доступна в виде Wiki — учебников, которые могут редактировать сами пользователи
- ✓ При компиляции с настройками проекта по умолчанию получается большой файл, размер которого может достигать 14,1 Мб.

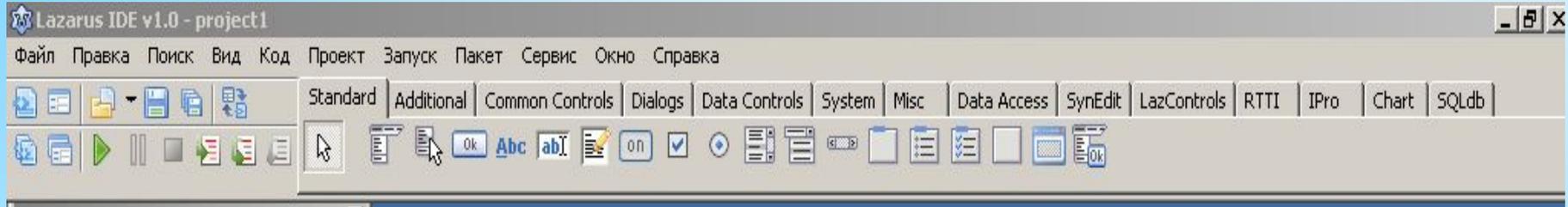
ПО, написанное на Lazarus

- **Total Commander** — 64-разрядная версия TC 8.0 компилируется с помощью Lazarus x64
- **Double Commander** — двухпанельный файловый менеджер с открытым исходным кодом, работающий под Linux и Microsoft Windows.
- **easyMP3Gain** — аудиоредактор для нормализации громкости mp3-файлов.
- **GreenGnome** — свободная среда рабочего стола для Microsoft Windows (аналог рабочего стола для Linux).
- **PeaZip** — свободный и бесплатный кроссплатформенный портативный архиватор и графическая оболочка для других архиваторов.
- **Ubuntu Control Center** — центр управления операционной системой ubuntu.
- **Transmission Remote GUI** — Front-end программа для управления BitTorrent

Процесс создания приложения :

- 1. Формирование окна программы-** расположение необходимых элементов, задание размеров, изменение свойств;
- 2. Написание программного кода,** описание свойств элементов, доступных только во время работы приложения, описание реакций на событие появления окна, нажатие на кнопку и других;
- 3. Отладка программы.**

1. Главное окно.

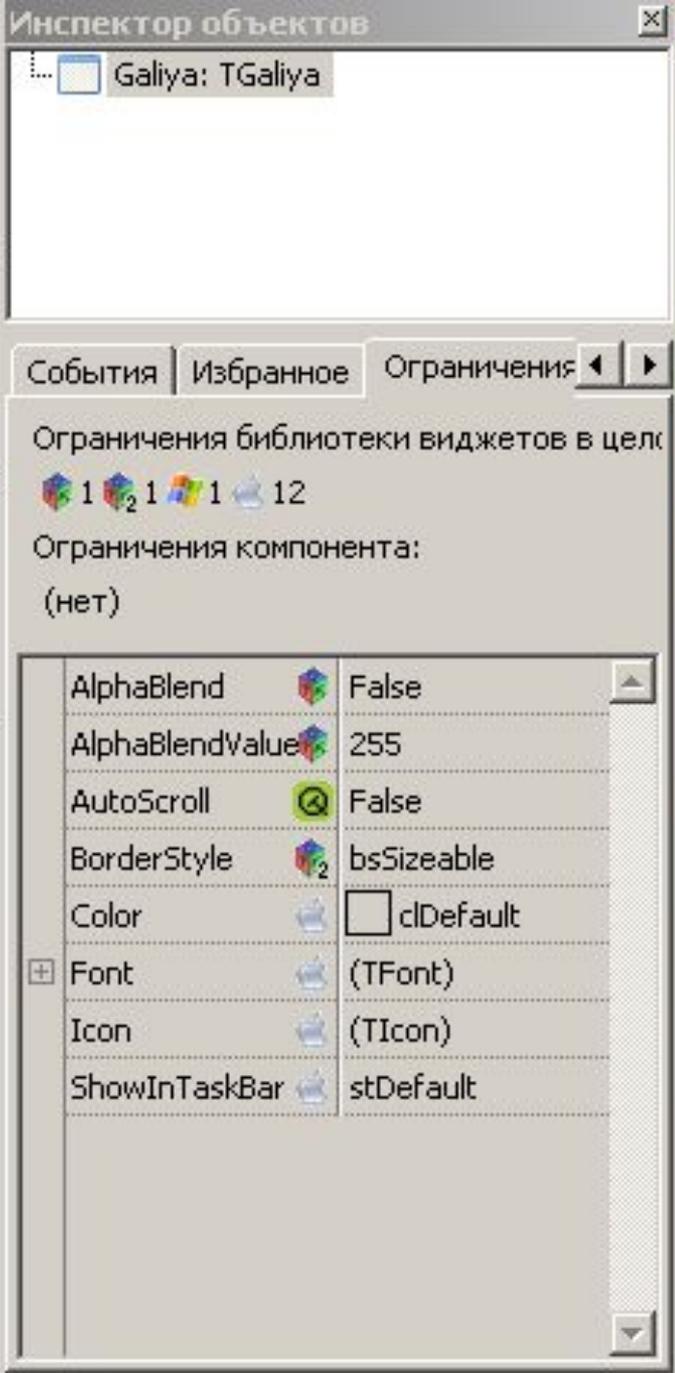


Здесь располагаются:

меню, панель инструментов и палитра компонентов.

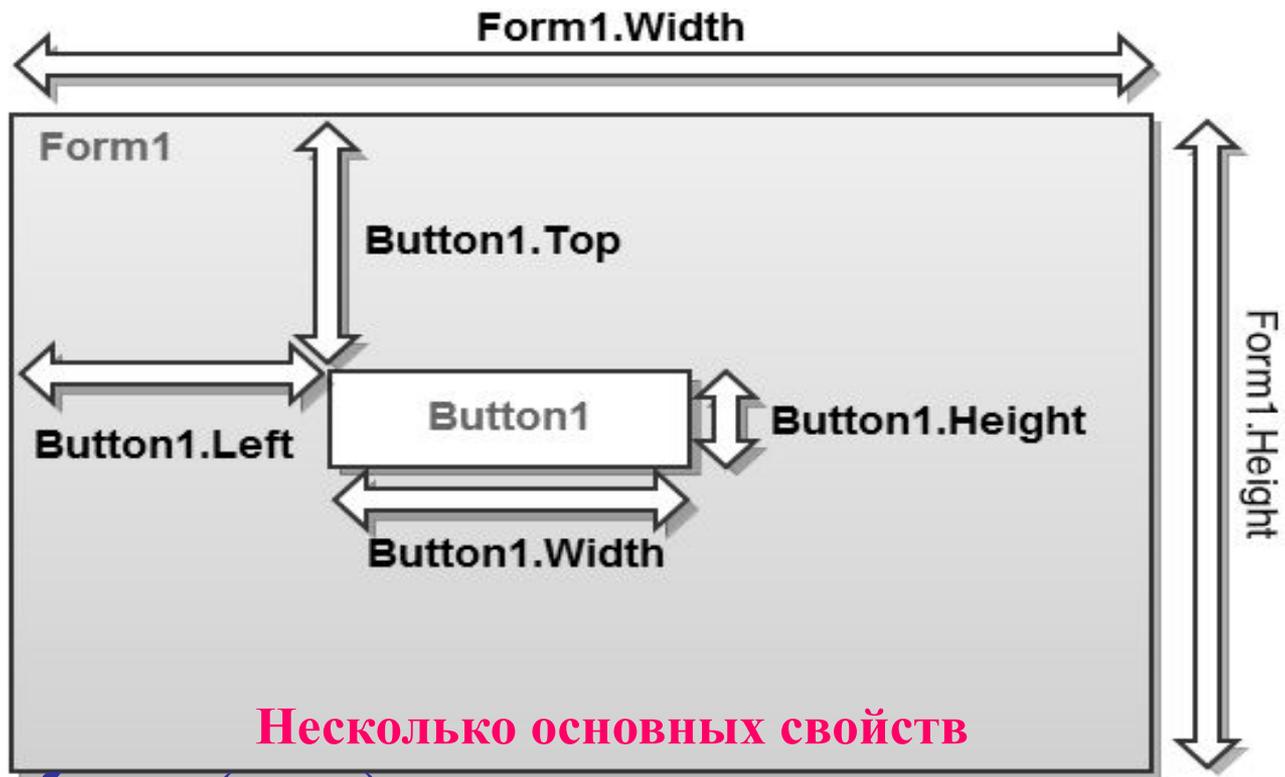
На палитре компонентов, представляющую множество тематических вкладок, располагаются визуальные и невидуальные компоненты будущей программы.

Невизуальные компоненты видны только на этапе проектирования приложения.



2. Инспектор объектов :

- **Свойства** – отображает доступные свойства выбранного компонента .
- **События** – содержит возможные обработчики событий для выбранного компонента.
- **Избранное**
- **Ограничения.**



Name - имя объекта (текст).

Caption - заголовок (текст). Текстовая 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 (то на чем расположен объект) (целое число).

Left - расстояние от левого края объекта до левого края родительского объекта (целое число).

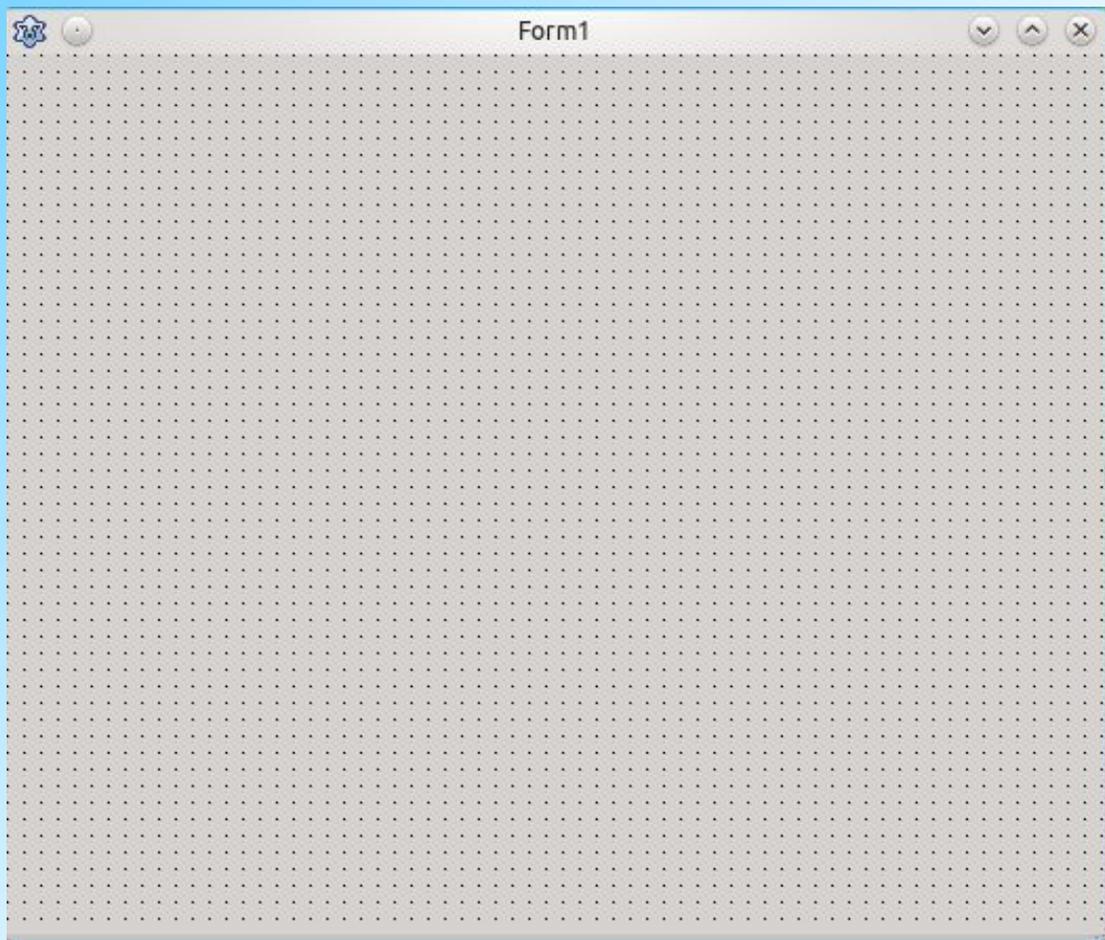
```
1  unit Unit1;  
.  
.  
.  
5  interface  
.  
  uses  
    Classes, SysUtils, FileUtil, Forms, Controls, Graphics, Dialogs;  
.  
10 type  
  TForm1 = class(TForm)  
  private  
    { private declarations }  
  public  
    { public declarations }  
15  end;  
.  
.  
.  
  var  
    Form1: TForm1;  
20  
  implementation  
.  
  {$R *.lfn}  
.  
25  end.  
26
```

3. РЕДАКТОР КОДА

В нем следует набирать текст программы. В начале работы над новым проектом это окно редактора кода содержит сформированный Lazarus шаблон программы.

*Строки пронумерованы, все служебные слова выделяются жирным цветом, знаки препинания становятся красными, строки с ошибками выделяются коричневым цветом, комментарии могут заключаются в { или (**), начинаться с // и выделяются синим.*

Текст программы разбивается на процедуры и функции, которые работают независимо

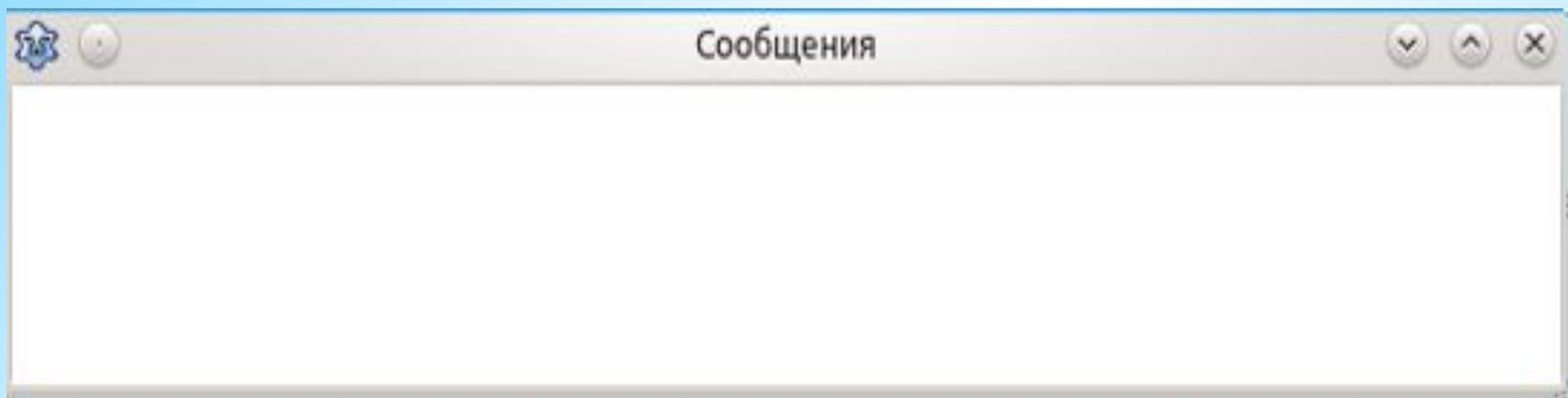


4. Проектировщик форм:

при запуске Lazarus автоматически предлагает пользователю новый проект, окно под названием Form 1, и назначает его главным окном.

Перенося на него элементы из палитры компонент, тем самым оформляете его. Главное окно в проекте может быть только одно, все другие создаваемые окна будут дочерними

5. Окно Сообщения служит для вывода сообщений при компиляции и сборке проекта.



Файл-Создать-Проект и выберите **Приложение**, или
выполните команду
Проект –Создать проект.

Для сохранения проекта
Файл- Сохранить как.

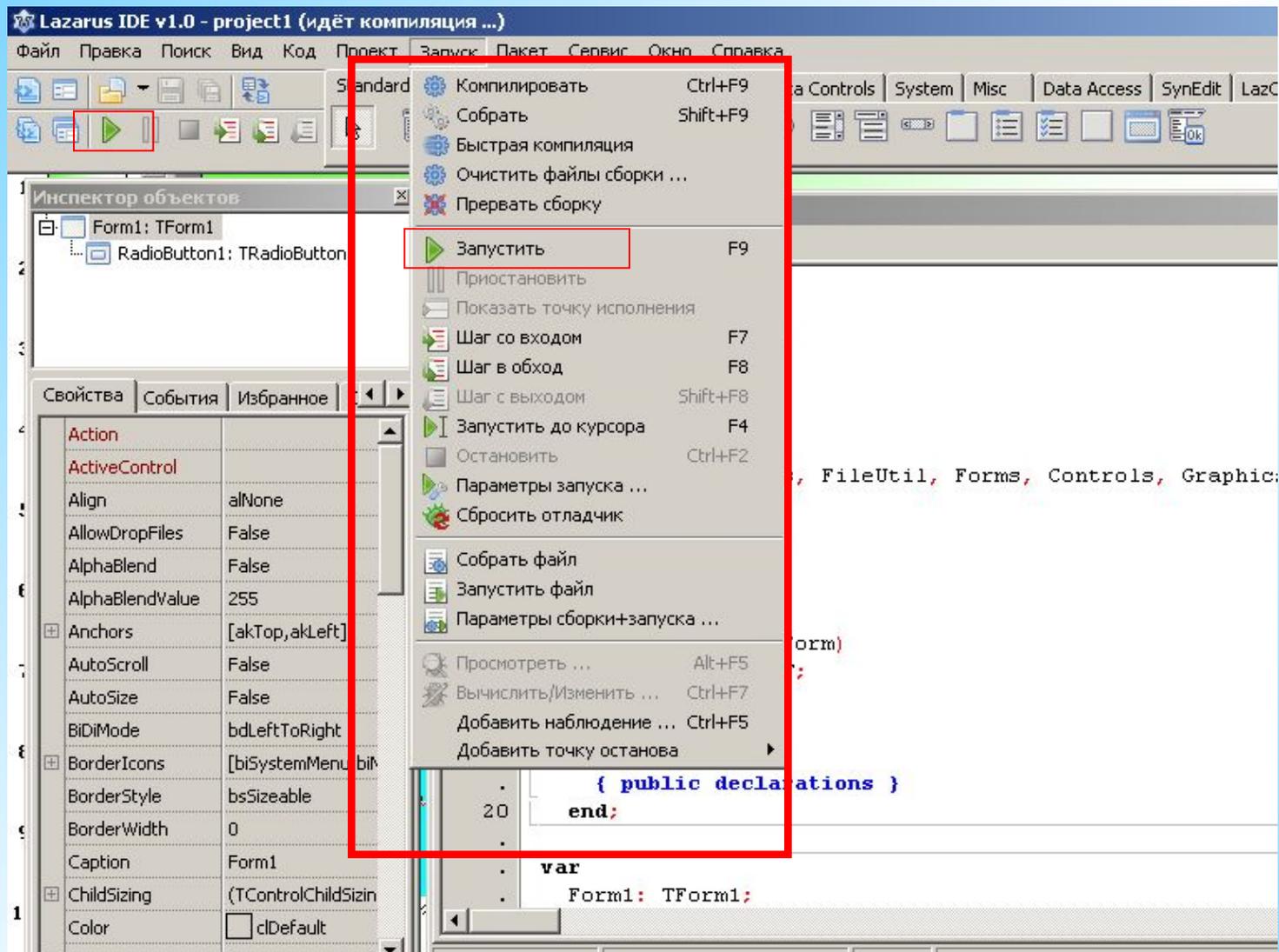
Каждый проект сохраняется в отдельный каталог!

Файлы проекта:

Модуль программы - Unit 1.pas

Проект – project1.lpi

**Файл Unit.lfm- файл с данными о проектировщике
форм**



5. Запуск Приложения на выполнение можно сделать нажатием клавиши F9, или через пункт в меню Запуск или выбором соответствующей кнопки на панели инструментов

- ✓ **Как запустить программу Lazarus?**
- ✓ **Из каких компонентов состоит окно программы?**
- ✓ **В чем отличие не визуальных от визуальных компонентов?**
- ✓ **С помощью какого свойства меняется заголовок у компонента?**
- ✓ **Проекты сохраняются в одном файле или нет?**

- 1. Запустите программу с компьютера.
- 2. Создайте новую форму.
- 2. Измените имя формы.
- 3. Поместите компонент Button на форму, измените его свойства.
- 4. Поместите компонент Label. Введите текст « Я программирую !».
- 5. Измените шрифт, цвет, расположение.
- 3. Сохраните проект.