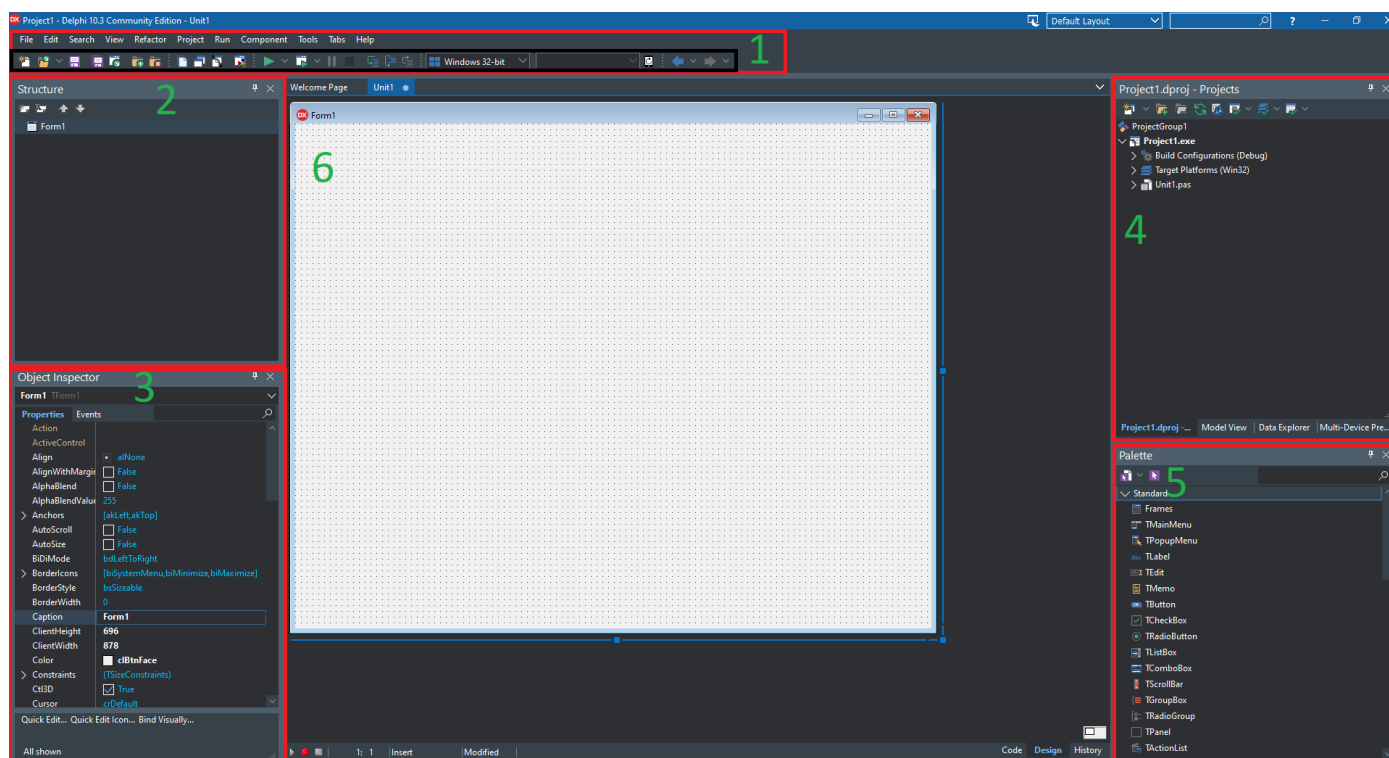


Описание интерфейса Delphi

Рис. Внешний вид среды визуальной разработки Delphi 10




1 - Главное меню с инструментальной панелью (выделено черным прямоугольником.).

Отдельные пункты меню, назначение которых понятно из их названий, объединяют сходные операции, относящиеся ко всему проекту в целом. Некоторые из пунктов меню содержат хорошо знакомые функции (открытие проекта, вывод печать, копирование в буфер и вставка из буфера).

Инструментальная панель содержит ряд кнопок, дублирующих некоторые наиболее часто используемые команды меню.

2 – Structure (Структура или дерево объектов). Это окно предназначено для наглядного отображения связей между отдельными компонентами, размещенными на активной форме или в активном модуле данных. Щелчок по любому компоненту в этом окне активизирует соответствующий компонент в окне формы и отображает свойства этого компонента в окне Инспектора объектов.

3 - Инспектор объектов. состоит из двух страниц: Properties (свойства) и Events (события).

Каждая страница разделена на две части. На странице свойств в левой части находится название свойства, а в правой – его значение. Значок «+» слева от названия указывает на то, что свойство состоит из нескольких подсвойств. Значениями свойств могут быть слова, числа, значения из раскрывающегося списка. Кнопка «» справа от свойства, раскрывает список возможных значений.

Примечание: если значением является число или текст, то после его набора лучше нажать Enter, иначе оно может быть не зафиксировано. При нажатии на «Esc» ввод отменяется.

Страница Events используется для задания реакции на событие. Состоит из двух частей. В левой - название события, а в правой – название процедуры, которая обрабатывает данное событие. Если в правой части ничего не написано, то программа на данное событие не реагирует.

4 - Менеджер проектов. Здесь мы можем посмотреть, какие файлы входят в созданный проект, открыть нужные модули, а также добавлять и удалять их.

5 - Палитра компонентов. Содержит панель библиотеки визуальных компонентов (Visual Component Library – VCL).

6 – Пространство для редактирования. В зависимости от выбранного модуля здесь редактируется дизайн формы и/или программный код.

Структура проекта Delphi

Проект – это разрабатываемое на языке программирования приложение.

Проект включает в себя не только форму с размещенными на ней компонентами, но и программные модули событийных процедур, которые описывают поведение объектов и взаимодействие объектов между собой.

Проект Delphi представляет собой набор программных единиц – модулей, которые хранятся в отдельных файлах.

*Примечание. В Delphi существуют файл проекта и файлы проекта. Это разные вещи. Файл проекта – это главный файл проекта (головная программа), имеющий расширение **.Dpr**, файлы проекта – это набор всех файлов приложения.*

*Файл с расширением **.Dpr** (главный файл проекта) содержит основную информацию о проекте. По умолчанию этот файл называется Project1.dpr.*

*Файл с расширением **.Pas** – это файл программного модуля. В нем хранится текст программы на языке Object Pascal. Для каждой формы, входящей в состав проекта, создается отдельный программный модуль. По умолчанию эти файлы называются Unit1.pas, Unit2.pas и т. д.*

*Файл с расширением **.Dfm** содержит информацию о внешнем виде формы. Этих файлов столько, сколько форм в проекте. Информация в них закодирована.*

*Файл с расширением **.Res** – это файл ресурсов проекта, в котором хранится информация о картинках, курсорах, иконках и т.п.*

*Файл с расширением **.Exe** – исполняемый файл приложения.*

*Файлы с расширениями **.~Df**, **.~Pa** – файлы со старыми версиями приложения.*

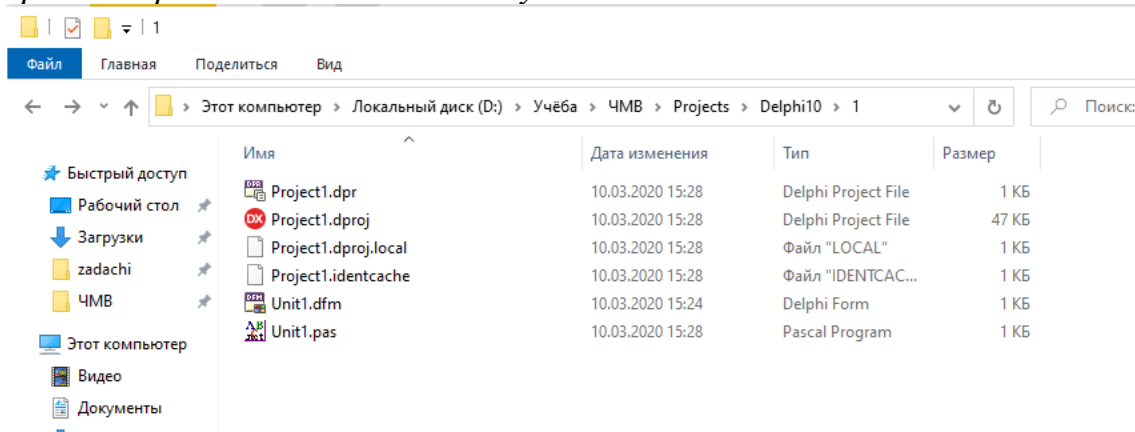
*У файлов с расширениями **.Pas**, **.Dfm**, **.~Df**, **.~Pa** – всегда одинаковое имя (по умолчанию Unit1).*

*У файлов с расширениями **.Dpr**, **.Exe**, **.Res** – также одинаковое имя (по умолчанию Project1).*

Примечание.

• При сохранении нового проекта приложения рекомендуется создавать отдельную папку.

• Копирование файлов проекта на дискету или в другую папку выполняется только с помощью файлового менеджера Windows (например, проводника). В среде Delphi это делать не следует.



Основные свойства компонентов TLabel, TButton, TEdit, TMemo

Caption (TLabel, TButton) - Связывает текстовую строку с управляющим компонентом: надпись на кнопке, текст метки, заголовок окна для формы

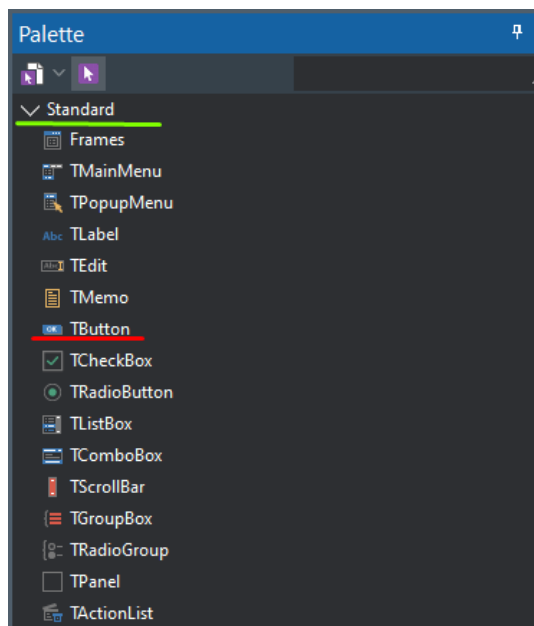
Text (TEdit) - Содержимое поля для текстового окна, которое будет видно при загрузке формы

Font (TLabel, TButton, TEdit, TMemo) - Внешний вид (размер, цвет и т.д.) шрифта для надписей.

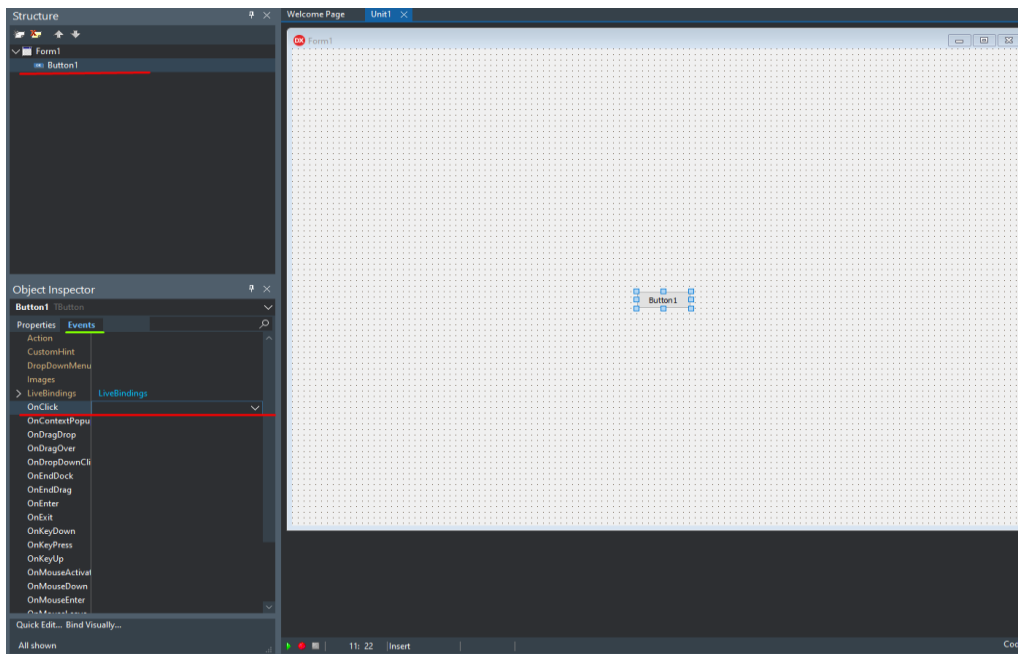
Name (TLabel, TButton, TEdit, TMemo) - имя формы/компонента. Имя, которое указанное здесь, будет использоваться в качестве названия объекта. Если это форма, то в коде будет создан шаблон объекта с таким же именем, но с добавлением в начале имени буквы "T". Формам и компонентам необходимо давать понятные имена. Так легче будет понять, для чего они предназначены. Ведь намного удобней будет работать с компонентом, если у него имя будет OpenButton или CloseFileButton, а не просто Button1 и Button2.

Создание обработчика события в Delphi(OnClick) используя кнопку (TButton).

Сначала необходимо поместить на форму компонент TButton. Для этого в палитре компонентов во вкладке Standard находим TButton, двойным щелчком помещаем на форму.



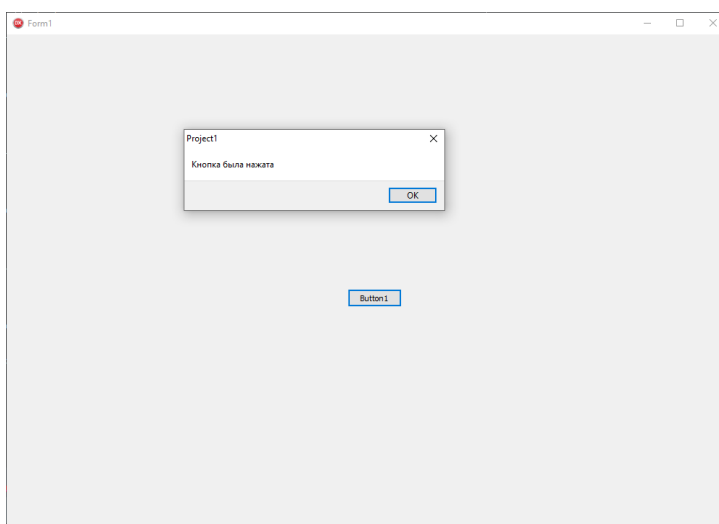
Затем двойным нажатием по кнопке на форме или двойным нажатием по пустому полю события OnClick во вкладке Events (При этом способе, кнопка должна быть выделена).



Delphi создает в редакторе кода процедуру - обработчик события OnClick. По умолчанию этой процедуре присваивается имя в виде имени компонента (в нашем случае имени нашей кнопки Button1) Таким образом имя нашей процедуры будет Button1Click, а созданная Delphi процедура будет выглядеть так:

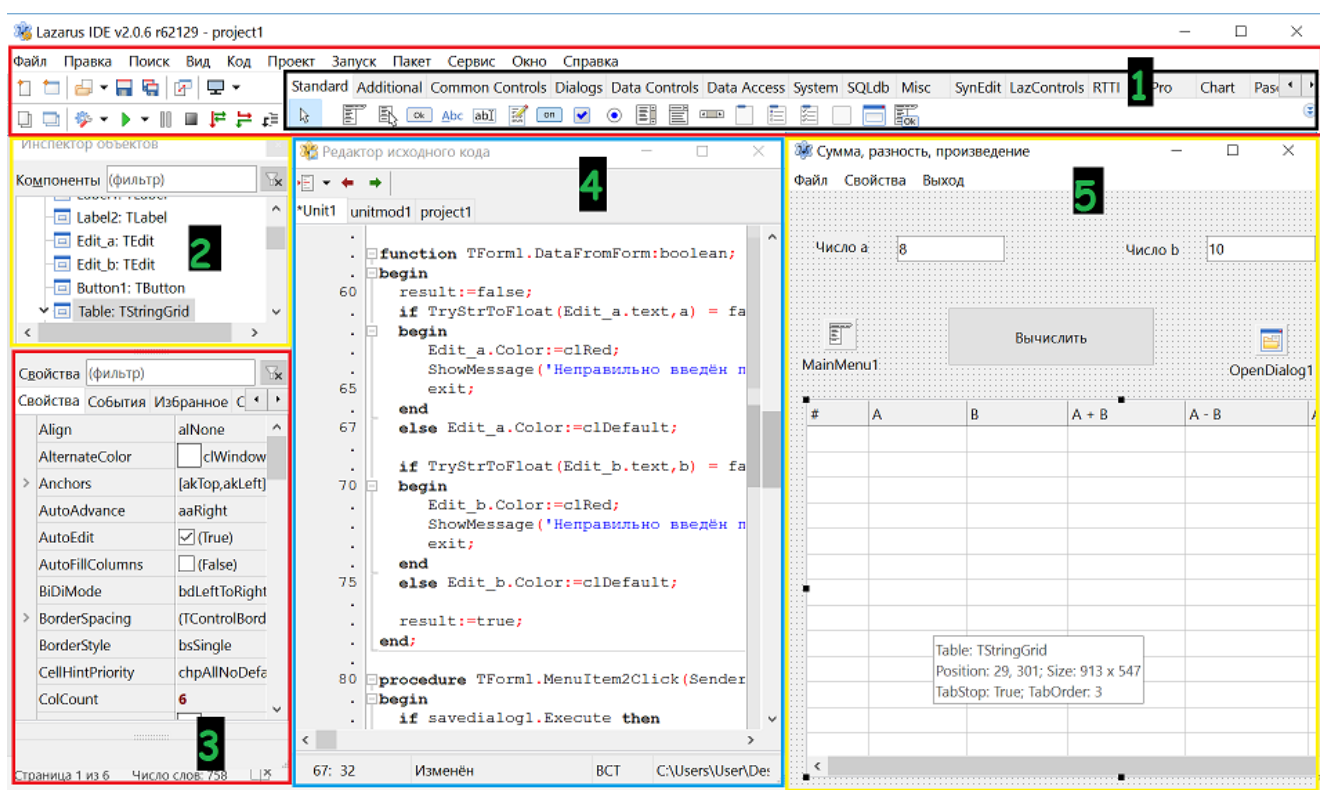
```
procedure TForm1.Button1Click(Sender: TObject);
begin
end;
```

```
procedure TForm1.Button1Click(Sender: TObject);
begin
    ShowMessage('Кнопка была нажата');
end;
```



Описание интерфейса Lazarus

Как видно, интерфейс данной IDE очень похож на Delphi 7. Точно также есть полосы главного меню, структура и инспектор объектов, окно с кодом\формой. Работа с Lazarus практически ничем не отличается от работы на Delphi 7, т. к. Lazarus задумывался как бесплатный аналог Delphi.




1 - Главное меню с инструментальной панелью (выделено черным прямоугольником.).

Отдельные пункты меню, назначение которых понятно из их названий, объединяют сходные операции, относящиеся ко всему проекту в целом. Некоторые из пунктов меню содержат хорошо знакомые функции (открытие проекта, вывод печать, копирование в буфер и вставка из буфера).

Инструментальная панель содержит ряд кнопок, дублирующих некоторые наиболее часто используемые команды меню.

2 – 3. Инспектор объектов и Структура. Это окно предназначено для наглядного отображения связей между отдельными компонентами, размещенными на активной форме или в активном модуле данных. Щелчок по любому компоненту в структуре активизирует соответствующий компонент в окне формы и отображает свойства этого компонента в окне Инспектора объектов.

Инспектор объектов состоит из двух страниц: Properties (свойства) и Events (события).

Каждая страница разделена на две части. На странице свойств в левой части находится название свойства, а в правой – его значение. Значок «+» слева от названия указывает на то, что свойство состоит из нескольких подсвойств. Значениями свойств могут быть слова, числа, значения из раскрывающегося списка. Кнопка «» справа от свойства, раскрывает список возможных значений.

Примечание: если значением является число или текст, то после его набора лучше нажать Enter, иначе оно может быть не зафиксировано. При нажатии на «Esc» ввод отменяется.

Страница Events используется для задания реакции на событие. Состоит из двух частей. В левой - название события, а в правой – название процедуры, которая обрабатывает данное событие. Если в правой части ничего не написано, то программа на данное событие не реагирует.

4 – Редактор исходного кода. В этом окне редактируется программный код любого открытого модуля. Каждая вкладка с именем – это какой-то определённый файл. Для удобства редактирования есть кнопки с переходом на определённый раздел и перемещения указателя по изменениям в коде.

5 – Редактор формы. В этом окне редактируется конкретная форма, т. е. изменяется положение, размер компонента, добавляются или удаляются компоненты и т. п. операции.

Структура проекта Lazarus

Проект – это разрабатываемое на языке программирования приложение.

Проект включает в себя не только форму с размещёнными на ней компонентами, но и программные модули событийных процедур, которые описывают поведение объектов и взаимодействие объектов между собой.

Проект Delphi представляет собой набор программных единиц – модулей, которые хранятся в отдельных файлах.

*Примечание. В Delphi существуют файл проекта и файлы проекта. Это разные вещи. Файл проекта – это главный файл проекта (головная программа), имеющий расширение **.Lpr**, файлы проекта – это набор всех файлов приложения.*

Файл с расширением **.Lpr** (главный файл проекта) содержит основную информацию о проекте. По умолчанию этот файл называется Project1.lpr.

Файл с расширением **.Pas** – это файл программного модуля. В нем хранится текст программы на языке Object Pascal. Для каждой формы, входящей в состав проекта, создается отдельный программный модуль. По умолчанию эти файлы называются Unit1.pas, Unit2.pas и т. д.

Файл с расширением **.Lfm** содержит информацию о внешнем виде формы. Этих файлов столько, сколько форм в проекте. Информация в них закодирована.

Файл с расширением **.Res** – это файл ресурсов проекта, в котором хранится информация о картинках, курсорах, иконках и т.п.

Файл с расширением **.Exe** – исполняемый файл приложения.

Файлы с расширениями **.~Lf**, **.~Pa** – файлы со старыми версиями приложения.

У файлов с расширениями **.Pas**, **.Lfm**, **.~Df**, **.~Pa** – всегда одинаковое имя (по умолчанию Unit1).

У файлов с расширениями **.Lpr**, **.Exe**, **.Res** – также одинаковое имя (по умолчанию Project1).

Примечание.

- При сохранении нового проекта приложения рекомендуется создавать отдельную папку.

- Копирование файлов проекта на дискету или в другую папку выполняется только с помощью файлового менеджера Windows (например, проводника). В среде Lazarus это делать не следует.

Основные свойства компонентов TLabel, TButton, TEdit, TMemo

Caption (TLabel, TButton) - Связывает текстовую строку с управляющим компонентом: надпись на кнопке, текст метки, заголовок окна для формы

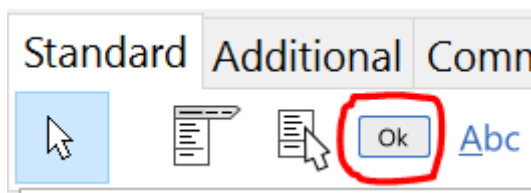
Text (TEdit) - Содержимое поля для текстового окна, которое будет видно при загрузке формы

Font (TLabel, TButton, TEdit, TMemo) - Внешний вид (размер, цвет и т.д.) шрифта для надписей.

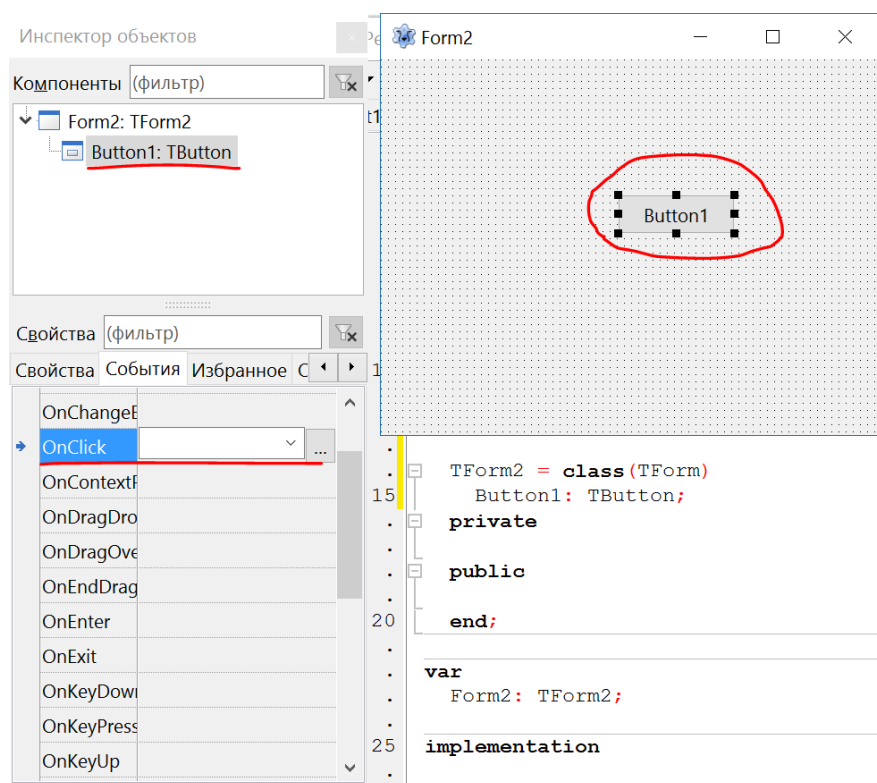
Name (TLabel, TButton, TEdit, TMemo) - имя формы/компонента. Имя, которое указанное здесь, будет использоваться в качестве названия объекта. Если это форма, то в коде будет создан шаблон объекта с таким же именем, но с добавлением в начале имени буквы "T". Формам и компонентам необходимо давать понятные имена. Так легче будет понять, для чего они предназначены. Ведь намного удобней будет работать с компонентом, если у него имя будет OpenButton или CloseFileButton, а не просто Button1 и Button2.

Создание обработчика события в Lazarus (OnClick) используя кнопку (TButton).

Сначала необходимо поместить на форму компонент TButton. Для этого в палитре компонентов во вкладке Standard, находим TButton, щелчком помещаем на форму.



Затем двойным нажатием по кнопке на форме или двойным нажатием по пустому полю события OnClick во вкладке Events (При этом способе, кнопка должна быть выделена).



Lazarus создает в редакторе кода процедуру - обработчик события OnClick. По умолчанию этой процедуре присваивается имя в виде имени компонента (в нашем случае имени нашей кнопки Button1) Таким образом имя нашей процедуры будет Button1Click, а созданная Lazarus процедура будет выглядеть так:

```
1 procedure TForm2.Button1Click(Sender: TObject);  
2 begin  
34 |  
35 end;
```

А далее добавляем какие-либо операции:

```
. procedure TForm2.Button1Click(Sender: TObject);  
. begin  
34     ShowMessage('Привет, мир!');  
35 end;
```

Нажимаем в меню на команду Запуск (F9) и проверяем:

