

Введение. Знакомство с интегрированной средой Lazarus. Изучение основных свойств формы.

Цель: знакомство с интегрированной средой разработки программного обеспечения Lazarus. Изучение основных свойств формы. Инженерный калькулятор.

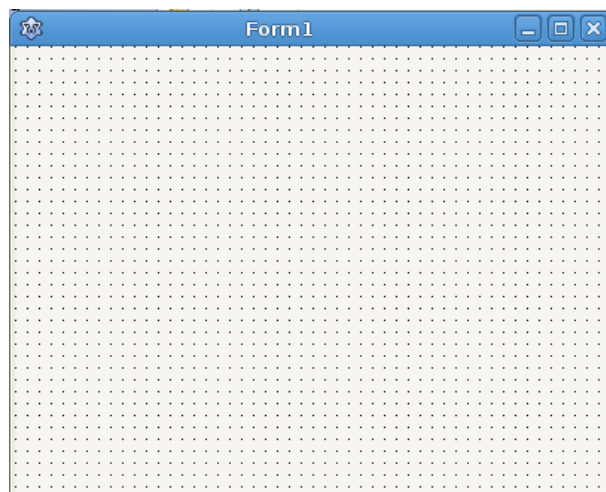
Справочный материал.

Форма – это важнейший визуальный компонент. Формы представляют собой видимые окна операционной системы и являются основной частью практически любого приложения. Термины "форма" и "окно" – синонимы, т. е. обозначают одно и то же.

Для работы с формой предназначен компонент **Form** класса **TForm**. С создания формы начинается конструирование приложения. В форме размещаются визуальные компоненты, образующие интерфейсную часть приложения, и системные (невизуальные) компоненты. Таким образом, в системе Lazarus форма является компонентом, который служит *контейнером* для всех других компонентов. В принципе можно создать и безоконное приложение, однако большинство приложений все же имеет видимое на экране окно, содержащее интерфейсную часть приложения.

Приложение может иметь несколько форм, одна из которых считается *главной* и при запуске программы отображается первой. При закрытии главного окна (формы) приложения прекращается работа всего приложения, при этом также закрываются все другие окна приложения. В начале работы над новым проектом Lazarus по умолчанию делает главной первую форму (с первоначальным названием Form1), но главной можно сделать любую форму приложения.

Типичная форма представляет собой прямоугольное окно с рамкой:



Большинство окон содержит *область заголовка*, в которой расположены значок заголовка, заголовок и ряд кнопок, позволяющие сворачивать, разворачивать (восстанавливать прежние размеры и положение) и закрывать окно, вызывать окно подсказки. Во многих формах отображаются также строка главного меню (под областью заголовка) и строка состояния (обычно в нижней части окна). При необходимости в форме могут автоматически появляться полосы прокрутки, предназначенные для просмотра содержимого окна. Остальная часть пространства окна называется *клиентской областью*. В ней можно размещать элементы управления, выводить текст и графику, манипулировать дочерними окнами.

Форма может быть модальной и немодальной. *Немодальная* форма позволяет без ее закрытия переключиться в другую форму приложения. *Модальная* форма требует обязательного закрытия

перед обращением к любой другой форме приложения.

Формы, которые отображают различные сообщения и требуют от пользователя ввода какой-либо информации, часто называют *диалоговыми окнами*. В свою очередь, диалоговое окно также может быть немодальным или модальным.

Характеристики формы

Как и любой другой визуальный компонент, форма имеет свойства, методы и события, общие для всех визуальных компонентов. Наряду с ними у формы есть и специфические свойства, методы и события, определяемые ее особым значением:

1. За *видимость* формы отвечает **Visible**. Оно может принимать значения **True** (Истина) и **False** (Ложь). Например строка кода:

```
Form1.Visible := False;
```

приведет к тому, что форма станет невидимой, при этом выполнение программы не будет остановлено.

2. *Размеры формы* можно изменять с помощью соответствующих свойств **Height** (Высота) и **Width** (Ширина) – изменяя значения этих свойств, Вы тем самым изменяете размеры формы. Например, строка кода:

```
Form1.Width := 100;
```

приведет к тому, что ширина окна (формы) станет равной 100 пикселей. Данные параметры можно менять как при разработке (в Инспекторе объектов), так и программно, в процессе выполнения программы.

3. *Стиль формы* определяется свойством **FormStyle** типа **TFormStyle**, принимающим следующие значения:

- **fsNormal** (стандартный стиль, используемый для большинства окон, в том числе и диалоговых);
- **fsMDIChild** (дочерняя форма в многодокументном приложении);
- **fsSplash** (форма без рамки – используется как правило для начальных заставок приложений);
- **fsMDIForm** (родительская форма в многодокументном приложении);
- **fsStayOnTop** (форма, которая после запуска всегда отображается поверх других окон) – обычно используется при выводе системной информации или информационной панели программы.

Форма может изменять стиль динамически – в процессе выполнения программы, например, при выборе пункта меню.

4. Группа свойств **BorderIcons**, включающая в себя свойства **biHelp**, **biMaximize**, **biMinimize** и **biSystemMenu** отвечают за наличие у заголовка окна кнопок справки, сворачивания/разворачивания окна и системное меню, вызываемое щелчком левой кнопки мыши по иконке приложения. Данные свойства имеют тип **Boolean**, т.е. могут принимать значения **True** или **False**. Если свойству поставить в соответствие значение **False**, то соответствующая данному свойству кнопка не будет доступна пользователю.

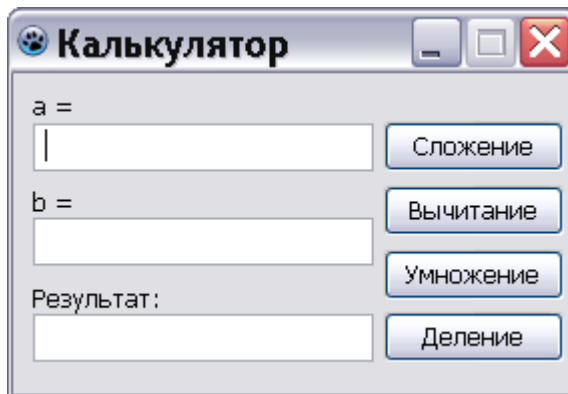
5. Каждая форма имеет ограничивающую *рамку*. Вид и поведение рамки определяет

свойство **BorderStyle** типа **TFormBorderStyle**. Оно может принимать следующие значения:

- **bsDialog** (диалоговая форма);
 - **bsSingle** (форма с неизменяемыми размерами);
 - **bsNone** (форма не имеет видимой рамки и заголовка и не может изменять свои размеры) – часто используется для заставок;
 - **bsSizeable** (обычная форма с изменяемыми размерами) – по умолчанию, имеет строку заголовка и может содержать любой набор кнопок;
 - **bsToolwindow** (форма панели инструментов);
 - **bsSizeToolWin** (форма панели инструментов с изменяемыми размерами).
6. *Размещение* формы при отображении определяет свойство **Position** типа **TPosition**. Оно может принимать значения, перечисленные ниже:
- **poDesigned** (форма отображается в той позиции и с теми размерами, которые были установлены при ее конструировании) – значение по умолчанию. Положение и размеры формы определяются свойствами **Left**, **Top**, **Width** и **Height**. Если приложение запускается на мониторе с более низким разрешением, чем у того, на котором оно разрабатывалось, часть формы может выйти за пределы экрана;
 - **poScreenCenter** (форма выводится в центре экрана, ее высота и ширина – свойства **Height** и **Width** – не изменяются);
 - **poDefault** (Операционная система автоматически определяет начальную позицию и размеры формы) – при этом значении программист не имеет возможности управлять этими параметрами, поэтому оно не допускается для форм многодокументных приложений;
 - **poDefaultPosOnly** (Операционная система определяет начальную позицию формы, ее размеры не изменяются);
 - **poDefaultSizeOnly** (Операционная система определяет начальные ширину и высоту формы и помещает форму в позицию, определенную при разработке);
 - **PoDesktopCenter** (форма выводится в центре экрана, ее высота и ширина не изменяются);
 - **PoMainFormCenter** (форма выводится в центре главной формы приложения, ее высота и ширина не изменяются) – это значение используется для вторичных форм, при применении его для главной формы оно действует как значение **poScreenCenter**;
 - **PoOwnerFormCenter** (форма выводится в центре формы, которая является ее владельцем, высота и ширина формы не изменяются) – если для формы не указан владелец (свойство **Owner**), то данное значение аналогично значению **poMainFormCenter**.
7. Свойство **WindowState** типа **TWindowState** определяет *состояние отображения формы* и может принимать одно из трех значений:
- **wsNormal** (обычное состояние) – по умолчанию;
 - **wsMinimized** (свернута);
 - **wsMaximized** (развернута).

Практические задания.

Задание №1: Изменить вид формы программы «Калькулятор» по образцу:



Выполнение задания.

Выполнение задания начнем с запуска **Lazarus** и открытия проекта калькулятора, который Вы сделали в ходе лабораторной работы №1. Затем приступим к изменению свойств формы и компонентов.

1. Для начала в Инспекторе объектов изменим следующие свойства формы:
 - свойству **Caption** зададим значение «Калькулятор»;
 - свойству **Height** зададим значение «160»;
 - свойству **Width** зададим значение «280»
2. Затем изменим свойства компонентов, расположенных на форме:
 - a) Свойствам компонента **Label1** необходимо задать следующие значения:
 - свойству **Left** задать значение «8»;
 - свойству **Top** задать значение «8»;
 - b) Свойствам компонента **Label2** необходимо задать следующие значения:
 - свойству **Left** задать значение «8»;
 - свойству **Top** задать значение «56»;
 - c) Свойствам компонента **Label3** необходимо задать следующие значения:
 - свойству **Left** задать значение «8»;
 - свойству **Top** задать значение «104»;
 - d) Свойствам компонента **Edit1** необходимо задать следующие значения:
 - свойству **Left** задать значение «8»;
 - свойству **Top** задать значение «25»;
 - свойству **Width** задать значение «170»
 - e) Свойствам компонента **Edit2** необходимо задать следующие значения:
 - свойству **Left** задать значение «8»;
 - свойству **Top** задать значение «72»;
 - свойству **Width** задать значение «170»
 - f) Свойствам компонента **Edit3** необходимо задать следующие значения:
 - свойству **Left** задать значение «8»;
 - свойству **Top** задать значение «120»;
 - свойству **Width** задать значение «170»
 - g) Свойствам компонента **Button1** необходимо задать следующие значения:
 - свойству **Left** задать значение «190»;

- свойству **Top** задать значение «24»;
 - свойству **Width** задать значение «80»
- h) Свойствам компонента **Button2** необходимо задать следующие значения:
- свойству **Left** задать значение «190»;
 - свойству **Top** задать значение «56»;
 - свойству **Width** задать значение «80»
- i) Свойствам компонента **Button3** необходимо задать следующие значения:
- свойству **Left** задать значение «190»;
 - свойству **Top** задать значение «88»;
 - свойству **Width** задать значение «80»
- j) Свойствам компонента **Button4** необходимо задать следующие значения:
- свойству **Left** задать значение «190»;
 - свойству **Top** задать значение «119»;
 - свойству **Width** задать значение «80»

Обратите внимание на то, что некоторые свойства можно менять одновременно для нескольких компонентов. Вы наверняка обратили внимание на то, что для компонентов **Label1**, **Label2**, **Label3** и компонентов **Edit1**, **Edit2**, **Edit3** значение свойства **Left** одинаково, следовательно его можно изменить для всех этих компонентов одновременно, для этого необходимо выбрать эти компоненты – удерживая нажатой левую клавишу **Shift**, щелкайте левой кнопкой мыши по указанным компонентам на форме (они будут выделяться рамочкой), затем, в Инспекторе объектов задайте свойству **Left** значение «10» и нажмите кнопку **Enter**, Вы увидите, что свойство **Left** поменяется для всех выбранных компонентов.

3. Прделайте ту же операцию для свойств **Left** и **Width** компонентов **TButton**, установив значение свойству **Left**, равное «185» и для свойства **Width**, равное «90».
4. Запустите проект на компиляцию и сборку исполняемого файла.
5. Вот мы и получили необходимую нам программу, соответствующую представленному образцу.

Задание №2: *Измените следующие свойства формы и законспектируйте наблюдаемые изменения во внешнем виде и поведении формы:*

- задайте свойству **BorderStyle** значение «**bsSingle**», скомпилируйте проект и запишите наблюдаемые изменения;
- задайте свойству **FormStyle** значение «**fsStayOnTop**», скомпилируйте проект и запишите наблюдаемые изменения;
- задайте свойству **Position** значение «**poDesktopCenter**», скомпилируйте проект и запишите наблюдаемые изменения;
- задайте свойству **WindowState** значение «**wsMinimized**», скомпилируйте проект и запишите наблюдаемые изменения.
- Верните все измененные в задании №2 свойства в значение «по умолчанию».