













































ПРИЛОЖЕНИЕ 1. ОПИСАНИЕ КОМПОНЕНТОВ, НАИБОЛЕЕ ЧАСТО ИСПОЛЗУЕМЫХ ПРИ ПРОЕКТИРОВАНИИ ПРИЛОЖЕНИЙ

Компо- ент	Описание
Стандартные компоненты (Standard)	
	<p>TMainMenu (Главное меню формы). Создает линейку меню для формы. После двойного щелчка мыши открывается дизайнер меню Menu Designer, с помощью которого можно установить обработчики событий по каждому пункту меню</p>
	<p>TPopupMenu (Контекстное меню). После двойного щелчка мыши открывается дизайнер меню Menu Designer</p>
	<p>TButton (Командная кнопка)</p>
	<p>TLabel (Надпись). Текст метки нельзя интерактивно изменить при работе приложения, его можно изменить только программно</p>
	<p>TEdit (Однострочное редактируемое текстовое поле). Предназначен для ввода и редактирования строки текста</p>
	<p>TMemo (Многострочное редактируемое текстовое поле)</p>
	<p>TToggleBox (Командная кнопка, похожая на TButton, которую можно «вжать»)</p>
	<p>TCheckBox (Флажок)</p>
	<p>TRadioButton (Переключатель)</p>
	<p>TListBox (Список)</p>
	<p>TComboBox (Раскрывающийся список)</p>
	<p>TScrollBar (Полоса прокрутки)</p>
	<p>TGroupBox (Контейнер с рамкой и надписью для группы компонентов)</p>

	<b>TRadioGroup</b> (Контейнер с рамкой и надписью для группы связанных переключателей)
	<b>TCheckGroup</b> (Контейнер с рамкой и надписью для группы связанных флажков)
	<b>TPanel</b> (Панель для размещения других компонентов)
	<b>TFrame</b> (Рамка-контейнер для размещения других компонентов)
	<b>TActionList</b> (Список пользовательских команд, которые могут использоваться различными элементами приложения)
<b>Дополнительные компоненты (Additional)</b>	
	<b>TBitBtn</b> (Кнопка с изображением)
	<b>TSpeedBtn</b> (Кнопка-пиктограмма без заголовка, например, для кнопок панели инструментов)
	<b>TImage</b> (Графический образ). Объект для отображения или редактирования растрового рисунка
	<b>TShape</b> (Стандартная геометрическая фигура)
	<b>PaintBox</b> (Окно для рисования)
	<b>TNoteBook</b>
	<b>MaskEdit</b> (Редактор ввода по шаблону)
	<b>CheckListBox</b> (Группа флажков). Похож на <b>ListBox</b> , но перед каждым элементом списка имеется флажок
	<b>ScrollBar</b> (Панель-контейнер, изменяемого размера, с отображением, при необходимости, полосами прокрутки)
	<b>StringGrid</b> (Текстовая таблица). Создает таблицу

	TDrawGrid
Дополнительные компоненты (Common Controls)	
	TTrackBar (Ползунок)
	TProgressBar (Индикатор прогресса)
	TUpDown (Счетчик)
	TreeView (Иерархическое дерево, например файловая структура диска)
	TListView (Иерархический список, например список файлов папки)
	TTabControl (Набор вкладок)
	TPageControl
Дополнительные компоненты (System)	
	Timer (Таймер). Способен инициировать события через регулярные промежутки времени. Не визуальный компонент. Для таймера главное событие – истечение заданного интервала времени. Используют для выполнения определенных действий через заданный интервал времени
Дополнительные компоненты (Dialogs)	
	OpenDialog (Окно открытия файла)
	SaveDialog (Окно сохранения файла)
	FontDialog (Панель выбора шрифтов)
	ColorDialog (Панель настройки цветов)

	FindDialog (Панель поиска текста)
	ReplaseDialog (Панель поиска и замены текста)
	PrintDialog (Окно настройки печати)

## ПРИЛОЖЕНИЕ 2. ОСНОВНЫЕ ОБЩИЕ СВОЙСТВА КОМПОНЕНТОВ LAZARUS

Одними и теми же свойствами могут обладать разные компоненты.

Свойство	Значение
Height, Width	Высота и ширина компонента в пикселях
Left	Положение левой кромки компонента относительно формы
Top	Положение верхней кромки компонента относительно формы
Align	Положение компонента относительно формы: alTop (сверху), alBotton (снизу), alLeft (слева), alRight (справа), alNone (определяется свойствами Left и Top)
Color	Цвет фона компонента
Font	Внешний вид (размер, цвет и т.д.) шрифта для надписей
Caption	Связывает текстовую строку с управляющим компонентом: надпись на кнопке, текст метки, заголовок окна для формы
Text	Содержимое поля для текстового окна, которое будет видно при загрузке формы
Visible	Если это свойство имеет значение True, то компонент будет видимым на форме, а если – False, то он будет невидимым

Enabled	Если это свойство имеет значение <code>False</code> , то компонент недоступен, т.е. щелчок мыши по этому элементу не даст эффекта, при этом текст, размещенный на нем, будет серым. Если это свойство имеет значение <code>True</code> , то компонент можно будет использовать в процессе работы программы
AutoSize	Если это свойство имеет значение <code>True</code> , то размеры оконного компонента будут автоматически меняться при изменении размера шрифта, или добавления в него неоконных компонентов
Hint ShowHint	Если свойство <code>ShowHint</code> имеет значение <code>True</code> , то всплывает подсказка – текст, содержащийся в свойстве <code>Hint</code>
Focused	Если это свойство имеет значение <code>True</code> , то данный компонент получил фокус. Поскольку элементов на форме много, то только один компонент на форме может получить фокус . Фокус объекту можно передать методом <code>SetFocus</code> в программном модуле. Компонента может получить фокус, только если оно видимо и доступно (его свойства <code>Visible</code> и <code>Enabled</code> имеют значения <code>True</code> ).
TabStop	Если это свойство имеет значение <code>True</code> , то данный компонент получает фокус с помощью клавиши <code>Tab</code> , т.е. определяет, сможет ли пользователь переходить с помощью клавиши <code>Tab</code> к данному оконному элементу управления
TabOrder	Порядковый номер выбора компонента клавишей <code>Tab</code>

### Примеры использования в программе:

Установка свойства в программе	Результат
<code>Edit1.AutoSize:=True;</code>	Автоматическое изменение размеров текстового окна
<code>Edit1.Text:=FloatToStr(a);</code>	Вывод в текстовое окно значения вещественной переменной <code>a</code>
<code>Button1.TabOrder:=3;</code>	После третьего нажатия клавиши <code>Tab</code> командная кнопка получит фокус

Button1.Hint:=’Кнопка завершает работу программы’;	После наведения курсора мыши на командную кнопку появится подсказка о ее назначении
--	---

### ПРИЛОЖЕНИЕ 3. ОСНОВНЫЕ МЕТОДЫ КОМПОНЕНТОВ LAZARUS

Одни и те же методы можно применить к разным объектам.

Метод	Описание метода
Hide	Скрывает компонент
Show	Показывает компонент
SetFocus	Передает фокус объекту, определенному в вызове метода. Поскольку клавиатура одна, а элементов на форме, которые могут ее использовать несколько, необходимо выделить элемент, которому в данный момент передается клавиатурный ввод. Компонент с фокусом ввода имеет значение True в своем свойстве Focused

Примеры использования в программе:

Использование метода	Результат воздействия метода на объект
Edit1.Clear	Очищается текстовое окно
Button1.SetFocus	Фокус передается командной кнопке
Form5.Hide	Форма Form5 удаляется с экрана
Form1.Show	Форма Form1 выводится на экран

#### ПРИЛОЖЕНИЕ 4. ПЕРЕЧЕНЬ СОБЫТИЙ

Все компоненты пользовательского интерфейса, которые размещаются на формах, обладают определенным перечнем событий, при совершении которых над соответствующим компонентом будут выполняться запрограммированные действия (событийная процедура).

##### События, инициируемые мышью:

Событие	Действие пользователя, которое вызывает данное событие
OnClick	Щелчок левой кнопкой мыши на объекте
OnDbClick	Двойной щелчок левой кнопкой мыши на объекте
OnMouseDown	Нажимается кнопка мыши при условии, что указатель мыши находится на объекте
OnMouseUp	Отпускается кнопка мыши при условии, что указатель мыши находится на объекте
OnMouseMove	Указатель мыши движется по верху объекта

Обработчики этих событий имеют следующие параметры:

Параметр	Описание
Sender	Объект, для которого выполняется действия мыши
Button	Указывает нажатую кнопку мыши: mbLeft , mbMiddle , mbRight
Shift	Указывает были ли нажаты клавиши Alt , Shift , Ctrl
X,Y	Координаты точки, в которой произошло событие

### События , инициируемые клавиатурой :

Событие	Действие пользователя, которое вызывает данное событие
OnKeyDown	Нажимается клавиша при условии, что объект находится в фокусе
OnKeyUp	Отпускается клавиша при условии, что объект находится в фокусе
OnKeyPress	Нажимается и отпускается алфавитно-цифровая клавиша при условии, что объект находится в фокусе

Обработчики этих событий имеют следующие параметры:

Параметр	Описание
Sender	Объект для которого выполняется нажатие клавиши
Key	Содержит код ASCII нажатой клавиши
Shift	Указывает были ли нажаты клавиши Alt , Shift , Ctrl

### События перемещения и сброса объектов

Операции перемещения и сброса позволяют перемещать целые компоненты или отдельные их элементы из одного компонента в другой. Для этих операций определены следующие события:

Событие	Действие пользователя, которое вызывает данное событие
OnDragDrop	Отпущена кнопка мыши. Выполнен сброс объекта
OnDragOver	Пользователь перетаскивает объект на элемента управления. Событийная процедура пишется для этого элемента управления
OnStartDrag	Левая кнопка мыши нажата и удерживается. Процесс перемещения объекта начался
OnEndDrag	Процесс перемещения и сброса объекта завершен



Замечание. Для объектов, участвующих в этих событиях, должны быть установлены следующие свойства: свойство DragMode перемещаемого компонента равно dmAutomatic ; свойство объекта DragKind , на который перемещаем должно быть равно dkDrag .

### События , инициируемые для компонентов

Событие	Описание
OnChange	Происходит изменение значения определенного свойства объекта. Например, для объекта TEdit это изменение свойства Text
OnEnter	Компонент получил фокус ввода
OnExit	Компонент потерял фокус ввода
OnActivate	Приложение или форма становится активной
OnCreate	Вызывается при создании (загрузке) формы
OnPaint	Вызывается при перерисовке формы
OnClose	Вызывается при закрытии формы
OnTimer	Вызывается через указанный промежуток времени

## ПРИЛОЖЕНИЕ 5. СТАНДАРТНЫЕ ПРОЦЕДУРЫ И ФУНКЦИИ

Библиотечные модули Lazarus содержат большое количество стандартных подпрограмм (процедур и функций). Ниже приведены наиболее часто используемые.

### МАТЕМАТИЧЕСКИЕ ФУНКЦИИ

Обращение	Тип параметров	Тип результата	Описание
Abs(x)	целый / вещественный	целый / вещественный	Модуль числа
Sin(x)	вещественный	вещественный	Синус (аргумент задается в радианной мере)
Cos(x)	вещественный	вещественный	Косинус (аргумент задается в радианной мере)
ArcTan(x)	вещественный	вещественный	Арктангенс
pi	аргумента нет	вещественный	Число $\pi$
Exp(x)	вещественный	вещественный	Экспонента числа $e^x$
Ln(x)	вещественный	вещественный	Натуральный логарифм
Sqr(x)	вещественный	вещественный	Квадрат числа
Sqrt(x)	вещественный	вещественный	Корень квадратный
Round(x)	вещественный	целый	Округление числа (round(3.5)=4 )
Trunc(x)	вещественный	целый	Отсекание дробной части числа (trunc(3.5)=3 )
Frac(x)	целый/ вещественный	тип параметра	Дробная часть числа (frac(3.2)=0.2)
Int(x)	целый / вещественный	тип параметра	целая часть числа (frac(3.2)=3)
Random(n)	целый	целый	Формирование случайного целого числа из диапазона [0, n-1]
Random*(x)	вещественный	вещественный	Формирование случайного вещественного числа из диапазона [0, x-1]

Математические функции, определенные в модуле <i>Math</i>			
ArcCos(x)	вещественный	вещественный	Арккосинус
Cosecant(x)	вещественный	вещественный	Косеканс (1/Sin(x))
Cotan(x)	вещественный	вещественный	Котангенс
Sec(x)	вещественный	вещественный	Секанс (1/Cos(x))
Tan(x)	вещественный	вещественный	Тангенс
Log10(x)	вещественный	вещественный	Логарифм числа по основанию 10
Log2(x)	вещественный	вещественный	Логарифм числа по основанию 2
LogN(N,x)	вещественный	вещественный	Логарифм числа $x$ по основанию $N$
Sign(x)	целый	целый	Определяет знак числа (возвращает 1 или -1)
IntPower(y,x)	$y$ - вещественный, $x$ - целый	вещественный	Возведение $y$ в целочисленную степень $x$
Power(y,x)	Оба вещественные	вещественный	Возведение $y$ в вещественную степень $x$

## ФУНКЦИИ ПРЕОБРАЗОВАНИЯ ТИПОВ ДАННЫХ

Обращение	Тип параметра	Тип результата	Описание
<b>Преобразование строк в другие типы</b>			
StrToInt(s)	строковый	целый	Преобразование символов строки <i>S</i> в целое число
StrToFloat(s)	строковый	вещественный	Преобразование символов строки <i>S</i> в вещественное число
BoolToStr(T,True)	T – логический	строковый	Преобразование логического типа в строку 'True' или 'False' .
<b>Обратное преобразование</b>			
FloatToStr(x)	вещественный	строковый	Преобразование вещественного числа <i>x</i> в строку
IntToStr(x)	целый	строковый	Преобразование целого числа <i>x</i> в строку
StrToBool(s)	строковый	логический	Преобразование строкового типа в логический. Если строка не равна 'True' или 'False' , возникает ошибка
FloatToStrF(x,Format, Precision, Digits)	вещественный	строковый	Преобразует вещественное число <i>x</i> в строку символов с учетом формата и параметров Precision и Digits .

### Параметры функции FloatToStrF:

Значение Format	Описание
ffExponent	Отображает число в экспоненциальной форме Precision – общее количество цифр в мантиссе; Digits – количество цифр в десятичном порядке. Например, FloatToStrF(x,ffExponent,8,3 ) выводит число в виде 1,5088805E+001

ffFixed	<p>Отображает число в формате с фиксированной запятой.  Precision – общее количество цифр;  Digits – количество цифр в дробной части  Например, FloatToStrF(x,ffExponent,8,3) <span style="float: right;">ВЫВО-</span>  дит число в виде 15,089</p>
---------	---

### ПРОЦЕДУРЫ ОБЩЕГО НАЗНАЧЕНИЯ

Ниже перечисленные процедуры не имеют параметров.

Обращение	Описание
Beep	Выдает звуковой сигнал, вызывая функцию Windows Api MessageBeep
Exit	Выполняет немедленный выход из текущей процедуры.
Break	Выполняет немедленный выход из текущего оператора цикла for, while или repeat. Управление передается на следующий после цикла оператор.
Continue	Прерывает текущую итерацию оператора цикла for, while или repeat и переходит на начало следующей итерации.

## СПИСОК ЛИТЕРАТУРЫ

1. Алексеев Е.Р., Чеснокова О.В., Кучер Т.В. Самоучитель по программированию на Free Pascal и Lazarus. - Донецк.: ДонНТУ, Технопарк ДонНТУ УНИТЕХ, 2009. - 503 с.
2. Культин, Н. Б. Основы программирования в Delphi 2010 / Никита Культин. – СПб: БХВ-Петербург, 2010. – 434 с.