**12. Событийно-ориентированное программирование. Основные компоненты в Lazarus и их характеристики.**

1. Форма (TForm) – основа GUI-приложения, контейнер для других компонентов.
2. Меню (TMainMenu) – позволяет вам поместить главное меню в программу (как File, Edit).
3. Меню (TPopupMenu) – позволяет создавать всплывающие меню.
4. Метка (TLabel) – служит для отображения текста на экране.
5. Поле ввода (TEdit) – стандартный управляющий элемент Windows для ввода. Он может быть использован для отображения короткого фрагмента текста и позволяет пользователю вводить текст во время выполнения программы.
6. Поле ввода (TMemo) – иная форма TEdit. Подразумевает работу с большими текстами.
7. Кнопка (TButton) – позволяет выполнить какие-либо действия при нажатии кнопки во время выполнения программы (OnClick).
8. Список (TListBox, TComboBox) – отображает список элементов.
9. Чекбокс (TCheckBox) – отображает строку текста с маленьким окошком рядом. В окошке можно поставить отметку, которая означает, что что-то выбрано.
10. Радиокнопка (TRadioButton) позволяет выбрать только одну опцию из нескольких.
11. Панель (TPanel) – управляющий элемент, используется в декоративных целях.
12. Кнопка (TBitBtn) – кнопка вроде TButton, однако на ней можно разместить картинку (glyph). TBitBtn имеет несколько предопределенных типов (bkClose, bkOK и др), при выборе которых кнопка принимает соответствующий вид.
13. Изображение (TImage) – отображает графическое изображение на форме. Воспринимает форматы BMP, ICO, WMF. Если картинку подключить во время дизайна программы, то она прикомпилируется к EXE файлу.

ОСНОВНЫЕ ОБЩИЕ СВОЙСТВА:

* Height, Width - высота и ширина компонента в пикселях;
* 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;
* Техт - 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 False , 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 True , то компонент можно будет использовать в процессе работы программы;
* AutoSize - если это свойство имеет значение True , то размеры оконного компонента будут автоматически меняться при изменении размера шрифта, или добавления в него неоконных компонентов;
* Hint ShowHint - если свойство ShowHint имеет значение True , то всплывает подсказка – текст, содержащийся в свойстве Hint;
* Focused - если это свойство имеет значение True , то данный компонент получил фокус;
* TabStop - если это свойство имеет значение True , то данный компонент получает фокус с помощью клавиши Tab, т.е. определяет, сможет ли пользователь переходить с помощью клавиши Tab к данному оконному элементу управления;
* TabOrder - порядковый номер выбора компонента клавишей Tab.

Основные события:

* OnClick - щелчок левой кнопкой мыши на объекте;
* OnKeyPress - нажимается и отпускается алфавитно-цифровая клавиша при условии, что объект находится в фокусе;
* OnCreate - вызывается при создании (загрузке) формы;
* OnClose - вызывается при закрытии формы;
* OnEnter – компонент получил фокус ввода;
* OnExit - компонент потерял фокус ввода;
* OnActivate – при активации формы (получении фокуса);
* OnShow – при отображении формы;
* OnChange – при изменении текста;
* OnTimer – срабатывает через заданный интервал;
* OnDblClick – при двойном клике.

**13. Подпрограммы в Lazarus. Основные способы передачи параметров в подпрограмму, их сравнение.**

Способы передачи параметров в подпрограмму:

1. По значению (параметр копируется в подпрограмму, изменения внутри подпрограммы не влияют на исходную переменную).

procedure MyProc(x: Integer); // По умолчанию передача по значению

1. По ссылке (работает с оригиналом переменной, изменения параметра сохраняются, переменная должна быть инициализирована перед вызовом)

procedure ProcName(var Param: Integer);

1. По константе (запрещает изменения параметра, гарантирует неизменность данных)

procedure Print(const Text: String);

1. Передача для вывода (параметр предназначен для возврата значения, начальное значение игнорируется, переменная может не инициализирована)

procedure ProcName(out Param: Integer);

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Критерий | по значению | var | const | out |
| Изменчивость | локальная | да | нет | да |
| Влияние на оригинал | нет | да | нет | да |
| Инициализация | любая (может быть инициализирована или нет) | требуется | любая | не требуется |
| Эффективность | низкая(значительная разница для крупных объектов) | высокая | высокая | высокая |
| Типы данных | простые | любые | сложные | любые |
| Безопасность | высокая | низкая | высокая | средняя |

Таким образом, выбор способа передачи зависит от конкретных требований вашей программы.

**14. Подпрограммы в Lazarus. Область видимости. Локальные и глобальные идентификаторы.**

Область видимости (scope) определяет область программы, в которой определённая переменная доступна для использования. Правильное использование областей видимости помогает предотвратить конфликты именования, улучшает читаемость кода и облегчает его поддержку.

Идентификатор — это уникальное имя, используемое для обозначения объектов в программе.

Основные области видимости:

1. Глобальная. Идентификатор объявлен вне всех подпрограмм, обычно в разделе var основной программы (program) или в interface-секции модуля (unit). Виден во всей программе.
2. Локальная. Идентификатор объявлен внутри конкретной подпрограммы (в ее разделе var или const). Виден только внутри этой подпрограммы.
3. Область видимости модуля. Идентификатор объявлен в interface-секции модуля. Виден всем модулям и основной программе, которые подключают данный модуль через uses. Идентификатор объявленный в implementation-секции модуля виден только внутри этого модуля.