Кодировка XML файла для организации окна интерактивной справки в программной среде clEV3r

**<?xml version="1.0"?>** - версия XML

**<doc>** - весь текст документации вставляется в раздел <doc> ... </doc>

**<members>** - список разноплановых частей (членов) документации <members> ... </members>

**<member name="Type:Name">** - начальный тэг для задания новой части документации (члена), где:

**Type** - тип части (члена) документации:

**T** - объект языка BasicPlus c именем **Name**. Либо стандартный объект (встроенный в язык BasicPlus), либо объект поставляемый из библиотеки \*.dll (не на языке BasicPlus), либо объект, задаваемый в модуле языка BasicPlus с расширением \*.bpm, который нужно импортировать командой:

import "Name"

в окне справки отображается в верхнем списке после иконки модуля:

 M - метод объекта c именем Name - функция в модуле \*.bpm. В окне справки отображается на втором уровне списка после иконки метода:

после блока описания модуля.

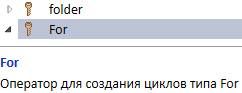
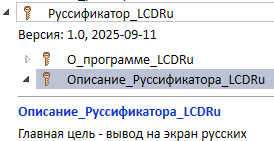
**P** - свойство объекта c именем Name - глобальная переменная в модуле \*.bpm, объявленная с типом number или string без начальной инициализации.

В окне справки отображается на втором уровне списка после иконки свойства:

**E** - переменная Name процедурного типа в которую можно записать адрес подпрограммы для обратного вызова. Используется, например, для задания подпрограммы, запускаемой в параллельном потоке.

Отмечается своей пиктограммой в виде жёлтой галочки.

**W** - заголовок для верхнего пункта списка справки (в данном случае Keywords или Команды). Далее во втором уровне списка справки идет перечисление ключевых слов (команд) языка BasicPlus:

**K** - ключевое слово (команда) языка BasicPlus. Отображается на втором уровне списка после иконки ключевого слова:

Отмечается своей пиктограммой в виде жёлтого ключика.

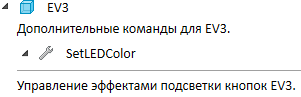
Само ключевое слово также выводиться в виде заголовка для текста пункта справки, описывающего назначение ключевого слова и заданного в тегах <summary> … </summary> сразу после тега <member name = “K:Name”>.

Кроме того, в принципе можно задавать с тем же тегом просто пункт справки для описания некоторых особенностей использования модулей языка.

К сожалению, пиктограммы типов в программной среде clEV3r отображаются только в общем дереве списка справки. В страничках, выводящих справку по отдельному пункту, пиктограммы вообще не отображаются. Во всплывающим списке подсказки для выбора объекта, переменной или команды языка BasicPlus отображаются правильные пиктограммы. А вот во всплывающих списках подсказки для выбора метода или свойства объекта для модулей \*.bpm, написанных на языке BasicPlus пиктограммы отображаются правильно, а для стандартных объектов (модулей) отображаются для всех пунктов только пиктограммы метода, и вместо пиктограммы свойства также отображается пиктограмма метода, что затрудняет понимание надо или нет ставить после имени круглые скобки, как для метода или не ставить, как для свойства. Но надо отметить, что свойств (переменных) в стандартных методах почти что и нет.

**<summary>** - описательная часть текущей части (члена) документации <summary>...</summary>

В окне справки отображается сразу после соответствующего пункта списка,

 со значком раскрытой ветки списка:

**<param name="Name">** - задание формального параметра метода (функции) с именем **Name**.

Параметры функции перечисляются в том порядке, в каком они заданы в определении

метода (функции) в круглых скобках в модуле, задающем объект. После тега должно

быть описание параметра до тега **</param>**. Также перед выводом описаний

 параметров выводится строка определения метода с параметрами:

**<returns>** - задание возвращаемого из метода значения. Задается описание до </returns>

Должно задаваться после описаний формальных параметров метода.

 Описание возвращаемого значения выводиться после слова **“Возвращает:”**

К сожалению, пока это реализовано только для тех стандартных объектов, которые

задаются через подключаемые библиотеки \*.dll. Для модулей, написанных на языке

BasicPlus для программной среды clEV3r возвращаемое значение для функций, пока

не реализовано. Но можно использовать этот пункт описания для вывода информации

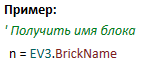
о возвращаемых параметра функции, заданных через параметр out.

**<example>** - задание примера. При двойном щелчке левой кнопкой мышки на примере, код этого

**<text>** примера копируется в окно редактора по месту расположения курсора.

**…** Пример задается до тега </example> и должен состоять из двух частей:

**</text>** описания примера, задаваемого в тегах <text> … </text> и самого кода, задаваемого **<code>**  в тегах <code> … </code>. Пример выводится после заголовка **“Пример:”**

 **…**

**</code>**

При этом код выводится как строки команд языка с соответствующей подсветкой.

В окне редактора программной среды clEV3r при щелчке левой кнопки мышки на названии объекта или названии метода, если они описаны в файле \*.xml и если отрыто окно справки, то производится переход на пункт списка справки с этим модулем или методом и открытие соответствующего пункта списка с описанием.

При вводе новой команды в окне редактора и при открытом окне справки, в окне справки вместо списка будет отображаться страница с описанием вводимого объекта.

Редактор при вводе очередного символа имени формирует текущий список возможных объектов, переменных и ключевых слов (команд) и выводит список в виде выпадающего подменю. Список возможных модулей и ключевых слов берется из окна справки. То есть, если в справке есть такой объект (модуль), то его имя будет отображаться в списке возможных оъектов. При этом следует учитывать, что стандартные объекты и объекты поставляемые из библиотек \*.dll (не на языке BasicPlus) не требуют импортирования через команду import, но похоже все таки должны быть описаны в файлах справки \*.xml. Модули \*.bpm, написанные на языке BasicPlus требуют импортирования через команду import и, более того, должны быть доступны из импортирующей их программы \*.bp.

При продвижении по выпадающему списку модулей, описание текущего модуля будет также отображаться в окне справки вместо списка. Нажатием клавиши **Enter** можно выбрать текущий объект, переменную или команду языка.

При вводе точки [.] после имени объекта, появляется выпадающее подменю со списком доступных методов объекта (кроме объявленных в модуле \*.bpm после команды private). А вот список методов формируется из самих подключаемых (импортируемых) модулей, а не из файлов справки. Стандартные объекты (модули) среды clEV3r и объекты, поставляемы из библиотек \*.dll (из каталога ./Lib) в импорте не нуждаются. Как отмечалось выше, модуль \*.bpm должен быть импортирован командой import и доступен в импортирующей программе.

При продвижении по выпадающему списку методов, описание текущего метода будет также отображаться в окне справки вместо списка. Нажатием клавиши **Enter** можно выбрать текущий метод. После чего он будет введен в строку редактора. В окне справки выведется список справки. К сожалению пункт справки с только что введенным методом не будет открыт, например, для подсказки какие надо вводить параметры. Для этого надо щелкнуть по названию метода левой кнопкой мышки.

Формирование списка справки для справочного окна программной среды clEV3r осуществляется (насколько я разобрался) следующим способом. В начале в верхний пункт “**Операции:** ” загружается стандартная справка по операциям из обычного текстового файла “Operators.txt”. Затем из файла “KeyWords.xml” скачивается справка по ключевым словам (командам) языка BasicPlus. Затем из каталога ./Lib считываются остальные файлы справки \*.xml. При этом недопустимо, чтобы в имени пункта справки (объекта-модуля, поле с тегом “T:” или метода, поле с тегом “M:”) были точки - “.”, то есть имя пункта справки не должно быть составным или содержать пробелы. Пункт (поле с тегом “T:”), задающий объект (модуль) должен быть один на файл \*.xml, вернее просто будет воспринята последняя по тексту запись в файле с таким тегом. Пункты справки по объектам, заданных в xml файлах, сортируются в алфавитном порядке независимо от имен файлов, где они заданы. Но если в имени файла встречается символ “\_”, то объекты из таких файлов сортируются отдельно и выводятся впереди всех остальных объектов, заданных в файлах без символа “\_”. Таким образом, например, объект LCDRu (из модуля “LCDRu.bpm”), заданный в файле с именем “0\_LCDRu” будет размещен в самом верху до объекта Assert, а тем более до стандартного объекта LCD в дереве списка пунктов справки.