

Краткая инструкция по использованию программы «ИС Дракон», версии 2008-05.04.2020, для разработки ПО на языке Lua.

Оглавление

Предисловие.....	1
Требования.....	1
Настройка программы.....	2
Создаем основу ДРАКОН программы.....	2
Добавляем программный код.....	3
Завершающие штрихи.....	3
Заключение.....	4
Сгенерированный исходный код на Lua с комментариями.....	4
Сгенерированный исходный код на Lua без комментариев.....	5
Использованные сокращения.....	5
Иллюстрации.....	5

Предисловие

При начале работы с программой «[ИС Дракон](#)» /разработчик: Тышов Геннадий Николаевич © 2008-2022 годы/, у пользователя могут возникать проблемы с первоначальной настройкой приложения, поскольку файл помощи [*DragonHelp.docx*], на который ссылается программа в опции меню *Справка* → *Дракон помощь* отсутствует, а примеры, которые можно найти в Интернете или на профильном [форуме](#), в основном, предназначены для более ранних версий этой программы.

Данная краткая инструкция как раз предназначена для того, чтобы помочь начать разработку ПО на языке [Lua](#) в данной системе.

Требования

Пользователь должен быть знаком с языком программирования [Lua](#) (если нет, то могу порекомендовать книгу, написанную автором этого языка, Робертом Иерусалимски: «Программирование на языке Lua» третье издание, которая была переведена на русский язык) и визуальным языком ДРАКОН (на сайте проекта [drakon.su](#), в разделе «[Книги Владимира Паронджанова Скачать бесплатно](#)» представлен ряд книг, детально описывающих этот язык и варианты его применения). Поскольку мы рассматриваем работу в программе «ИС Дракон», потребуется это программное обеспечение. Загрузить данное ПО можно бесплатно по [ссылке](#) (активна на момент публикации данного документа), представленной на [странице программы](#), но необходимо отметить, что это коммерческое программное обеспечение, которое предоставляет 15 дней на ознакомление с возможностями программы, после чего пользователь должен приобрести лицензию на его использование. Варианты лицензирования и цены представлены в документе «Активация.txt». Необходимо отметить адекватную стоимость лицензии, а также наличие скидок для студентов и школьников. Помимо самой программы, представленной в виде архива: **Dragon_2020-04-05.rar** или исполняемого файла **Dragon.exe**, потребуется еще архив **Prg_Lang.rar** с файлами, определяющими параметры генерации исходных текстов программ на одиннадцати языках, поддерживаемых приложением. На текущий момент (ноябрь 2022 года), поддерживается генерация исходного кода для следующих языков программирования: **VBA, 1C, BASCOM, C, Delphi, FSM, Java, JS, Lua, PHP** и **ST**. Данные файлы обычно располагаются в том же каталоге,

что и исполняемый файл приложения **Dragon.exe**, или в подкаталоге **Prg_Lang**. Вариант по умолчанию, задается в файле **LangProg.lpr**, который является копией одного из этих файлов (в зависимости от того какой язык программирования Вы планируете использовать).

Настройка программы

1. В меню «Вид», выбираем следующие опции: включаем режим «√ - Меню: Программирование», «• •Плакаты о светлом фоне», «√ - Номера», «√- А-,Б-,П-тексты, ссылки» ([рис.1](#)).
2. В меню «Программирование», выбираем опцию: «√ - С комментариями» ([рис. 2](#)).
3. В меню «Программирование» выбираем опцию «Выбор файла описания языка» и выбираем файл описания языка Lua: **Prg_Lua.lpr** ([рис. 3](#)).

Создаем основу ДРАКОН программы

1. Меню «Файл» -> «Создать новый лист ...» (Ctrl-N), создаем файл в формате «ИС Дракон», сохраняем с определенным названием, допустим «**Drakon-Lua.drt**» ([рис. 4](#)).
2. ПКМ в пределах созданного листа «**Drakon-Lua.drt**», выбираем опцию «Примитив» (для данного примера, в дальнейшем, скорее всего Вы будете использовать схему «Силуэт» для более сложных программ). Система добавляет схему «Примитив» ([рис. 5](#)).
3. ЛКМ на созданной схеме «Примитив» (схема будет подсвечена рамкой зеленого цвета), затем ПКМ в пределах выделенной схемы: выбираем опцию «Свойство схемы -> • •Прг. – программа. Язык с метками» ([рис. 6](#)).
4. ПКМ в пределах листа «**Drakon-Lua.drt**», выбираем опцию «Гном», добавляется схема «Гном» для компоновки программы на Lua ([рис. 7](#)).
5. Настройка схемы «Гном»: ЛКМ на схеме «Гном» (схема будет подсвечена рамкой зеленого цвета), затем ПКМ в пределах выделенной схемы: выбираем опцию «Свойство схемы -> • •Прг. – программа. Язык с метками» ([рис. 8](#)).
6. Продолжение настройки схемы «Гном»: добавляем иконку «Модуль». Для этого ЛКМ в схеме «Гном», около знака вопроса под иконкой с надписью «?Заголовок». Появится маленький прямоугольник, подсвеченный зеленым цветом ([рис. 9](#)). ПКМ на этом прямоугольнике, выбираем опцию «Модуль» ([рис. 10](#)). Результат добавления представлен здесь: ([рис. 11](#)).
7. Опционально: Ваша программа может содержать дополнительные функции, определенные в других схемах и вызываемые за счет использования иконки «Вставка». В данном примере, добавим еще одну схему, в которой определим функцию для печати сообщения. Процедура аналогична пунктам 2 и 3. Результат здесь ([рис. 12](#)).
8. Опционально: можно добавить название в заголовках листа и схем для более удобной работы с этими объектами. Для этого необходимо 2хЛКМ на заголовке листа / схемы и в появившемся окне ввода задать необходимое имя. Приведен пример для схемы функции – указываем название «Схема Функция» ([рис. 13](#)). Для того, чтобы закрыть это окно ввода: ЛКМ в любом свободном месте листа или нажать клавишу «Esc». В результате у нас есть лист с названием: «Демо программа на Lua», основная схема «Схема программы», схема функции «Схема Функции» и компоновочная схема «Схема Гном».
9. Настраиваем компоновочную схему: ЛКМ на иконке «Модуль», в открывшемся окне редактора указываем имя выходного файла, в котором будет сохранен текст программы на Lua, например: «**dlua.lua**» ([рис. 14](#)). Внимание, по умолчанию файл с исходным текстом программы будет сохранен в той же директории, что и «**Dragon.exe**», если Вы хотите сохранить его в подкаталоге, например подкаталоге «**proj**», укажите имя файла в следующем виде: «**projdlua.lua**».
10. Добавляем схемы программы и функции в иконку «Модуль», схемы «Гном»: для этого выполняем ЛКМ на необходимой схеме, затем ПКМ и выбираем опцию «Ссылка на схему для иконы Модуль» ([рис. 15](#)). Важно: в Lua описания функций должны быть определены до их использования в программе, поэтому важна очередность включения этих модулей. В нашем случае, сначала включим схему «Схема Функция», а затем «Схема Программа». После выбора опции «Ссылка на схему для иконы Модуль» (см. выше), копируем полученный текст: «**Drakon-Lua.drt, 3**» для схемы «Функция», переходим в схему «Гном», ЛКМ на иконке «Модуль», откроется окно редактирования, переходим на новую строку и

вставляем полученный текст ([рис.16](#)). Повторяем для основного модуля. В результате, иконка «Модуль» будет выглядеть так ([рис. 17](#)).

Добавляем программный код

1. В схеме «Программа» добавим две иконки: «Действие» и «Вставка». Для этого ЛКМ под иконкой с надписью «?Заголовок», затем ПКМ на квадратике, который подсвечен зеленым цветом, выбираем опцию «Действие», аналогично под иконкой «Действие» добавляем иконку «Вставка».
2. В схеме «Функция» под заголовком добавляем иконку «Действие» аналогично пункту 1 и изменяем название в иконке «?Заголовок» на «функция печати». Название иконки «Действие» меняем на «печать из функции».
3. В иконке «Действие» на схеме «Функция» ЛКМ на квадратике №3, расположенном под иконкой, с правой стороны, откроется окно редактирования П-текста (программного кода), в которое мы вставляем те команды, которые будет выполнять эта функция ([рис. 18](#)). В нашем примере это просто вывод сообщения: **`print("Hello from function pr()\n")`**
4. После того, как мы закроем окно редактирования, квадратик, который мы выбирали, изменит цвет на черный, что означает – иконка содержит программный код. Но функция в Lua должна начинаться с служебного слова **`function`** и заканчиваться служебным словом **`end`**. Для того, чтобы привести ее в соответствие с требованиями языка Lua, внесем следующие изменения в программный код заголовка схемы «функция печати». Для этого ЛКМ на квадратике №3, расположенном под заголовком, откроется следующее окно: ([рис. 19](#)).
5. Заменяем первую строку «..... Нет текста» на заголовок функции: **`function pr()`**, а третью строку «..... Нет текста» на: **`end`**
6. Важно: вторую строку «`==`» оставляем без изменений!!! Ее нельзя стирать, удалять или изменять. У Вас должно получиться что-то вроде этой картинки ([рис. 20](#)). Закрываем окно редактирования клавишей «Esc» или ЛКМ на свободном пространстве листа.
7. Возвращаемся к основной схеме программы. Поменяем названия заголовка и иконки действия. Заголовок назовем «Дракон - Lua», а действие «Печать приветствия».
8. Добавим программный код в иконку действия: ЛКМ на квадратике №3 под этой иконкой и в редакторе добавим ([рис. 21](#)) программный код: **`print("Hello from Drakon - Lua!\n")`**
9. Финальный штрих: добавим вызов нашей созданной функции из иконки «Вставка»: ЛКМ на квадратике №3, в редакторе кода добавляем вызов функции: **`pr()`** ([рис. 22](#)).

Завершающие штрихи

Мы практически закончили подготовку и готовы перейти к генерации исходного кода. Для этого ЛКМ на заголовке листа «Демо программа на Lua» (заголовок станет зеленого цвета) и используем опцию программы: **Программирование -> Сохранить программный код листа** (или клавиша F9). По степени бесполезности, наша программа на Lua может занять почетное второе место после наиболее распространённого примера, печатающего фразу «Hello World!», но нам удалось пройти все необходимые шаги по созданию программы на Lua с помощью системы «ИС Дракон»! Теперь можно усложнять задачу и перейти к созданию более сложных приложений. Но даже сейчас мы проработали все основные шаги по созданию приложений, причем задействовали и опцию создания и использования функции на Lua в нашей программе.

В реальной жизни, программы намного сложнее, вот несколько дополнительных рекомендаций:

- если в программе требуется использование сторонних библиотек, определения дополнительных констант или функций – все это можно сделать, используя опцию редактирования квадратика №3 иконки «Модуль» схемы «Гном». В режиме редактирования, замените первую строку «..... Нет текста» необходимыми Вам данными и процедурами. Например, для использования библиотеки **IUPLUA** построения графического интерфейса на Lua добавьте: **`require("iuplua")`**
- Важно пользоваться рекомендациями раздела «Добавляем программный код» и не удалять вторую строку «`==`». Если какой-то из разделов не используется, то лучше его заменить на комментарий, то есть заменить «..... Нет текста» на «-- Нет текста».

- Система «ИС Дракон» генерирует выходные файлы в кодировке ASCII cp1251 (Windows). На текущий момент кодировка UTF-8 не поддерживается. Учитывайте это при работе с файлами, которые генерирует эта программа. Скорее всего Вам потребуется сторонняя программа, которая сможет преобразовать файлы в требуемую Вам кодировку. Какую программу использовать – все зависит от Ваших предпочтений. Чаще всего я использую удобный редактор исходного кода [CudaText](#).

Заключение

Надеюсь, что информация, представленная в данном документе, окажется полезной для Вас. Ваши дополнения и замечания можно направлять по [контактным адресам](#) на сайте моего проекта [AGB Corp](#). Пример выполнения сгенерированной программы в ПО **lupLuaScripter** ([рис. 23](#)).

Сгенерированный исходный код на Lua с комментариями

```
1  -- ИС Дракон. Маршрутный транслятор. Выпуск 05.04.2020
2
3  -- 3.2.0-Гном |Схема Гном|Файл: Drakon-Lua.drt
4  -- 3.2.3-Модуль Начальный текст. ||
5  -- ..... Нет текста
6
7  -- Выполнена компиляция - 09.11.2022 19:44:32
8  -- 3.3.0-Примитив |Схема Функция|функция печати|Файл: Drakon-Lua.drt
9
10 -- 3.3.1-Заголовок Начальный текст. ||функция печати||
11 function pr()
12
13 -- 3.3.3-Действие ||печать из функции||
14 print("Hello from function pr()\n")
15
16 -- 3.3.2-Конец ||||
17
18 -- 3.3.1-Заголовок Конечный текст. ||функция печати||
19 end
20
21
22 -- Выполнена компиляция - 09.11.2022 19:44:32
23 -- 3.1.0-Примитив |Схема Программа|Дракон - Lua|Файл: Drakon-Lua.drt
24
25 -- 3.1.1-Заголовок Начальный текст. ||Дракон - Lua||
26 -- ..... Нет текста
27
28 -- 3.1.3-Действие ||Печать приветствия||
29 print("Hello from Drakon - Lua!\n")
30
31 -- 3.1.4-Вставка ||функция печати||
32 pr()
33
34 -- 3.1.2-Конец ||||
35
36 -- 3.1.1-Заголовок Конечный текст. ||Дракон - Lua||
37 -- ..... Нет текста
38
39
40 -- 3.2.3-Модуль Конечный текст. ||
41 -- ..... Нет текста
42
```

Сгенерированный исходный код на Lua без комментариев

Если отключить опцию меню Программирование -> - С комментариями, то результирующий код становится короче:

```
1  -- ИС Дракон. Маршрутный транслятор. Выпуск 05.04.2020
2
3  -- 3.2.0-Гном |Схема Гном|Файл: Drakon-Lua.drt
4  -- 3.2.3-Модуль Начальный текст. ||
5  -- ..... Нет текста
6
7
8  -- Выполнена компиляция - 09.11.2022 19:47:54
9  -- 3.3.0-Примитив |Схема Функция|функция печати|Файл: Drakon-Lua.drt
10
11 function pr()
12
13 print("Hello from function pr()\n")
14
15 end
16
17
18 -- Выполнена компиляция - 09.11.2022 19:47:54
19 -- 3.1.0-Примитив |Схема Программа|Дракон - Lua|Файл: Drakon-Lua.drt
20
21 -- ..... Нет текста
22
23 print("Hello from Drakon - Lua!\n")
24
25 pr()
26
27 -- ..... Нет текста
28
29
30 -- 3.2.3-Модуль Конечный текст. ||
31 -- ..... Нет текста
```

Использованные сокращения

- 2хЛКМ — двойной щелчок левой клавишей мыши;
- ЛКМ — левая клавиша мыши;
- ПКМ — правая клавиша мыши;

Иллюстрации

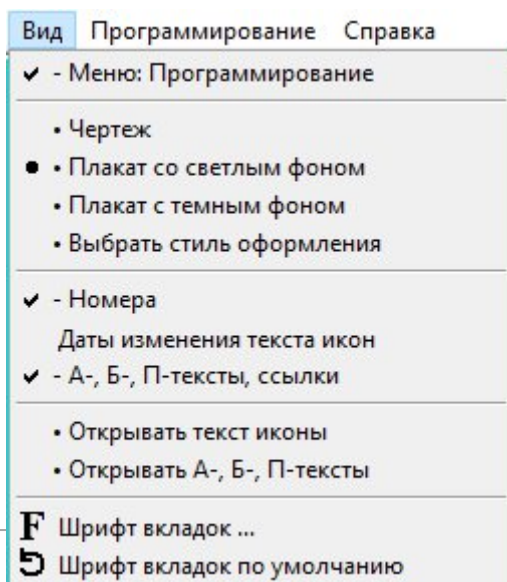


Рис. 1 Опции меню «Вид»

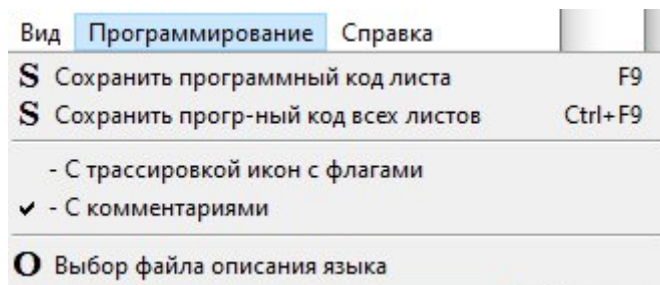


Рис. 2 Опции меню «Программирование»

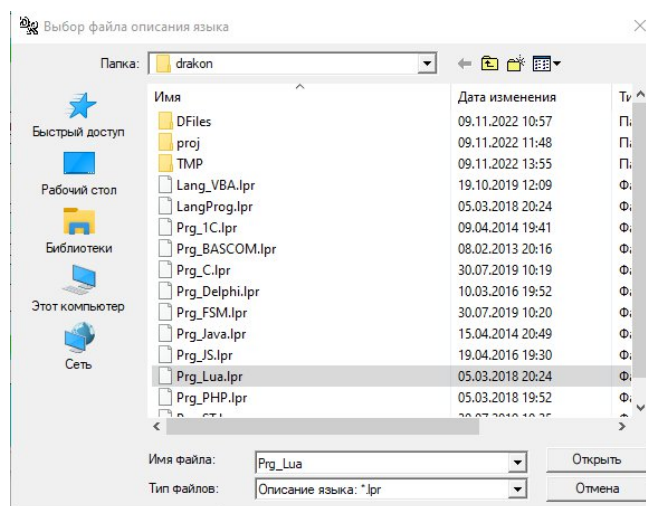


Рис. 3 Выбор описания языка Lua: Prg_Lua.lpr

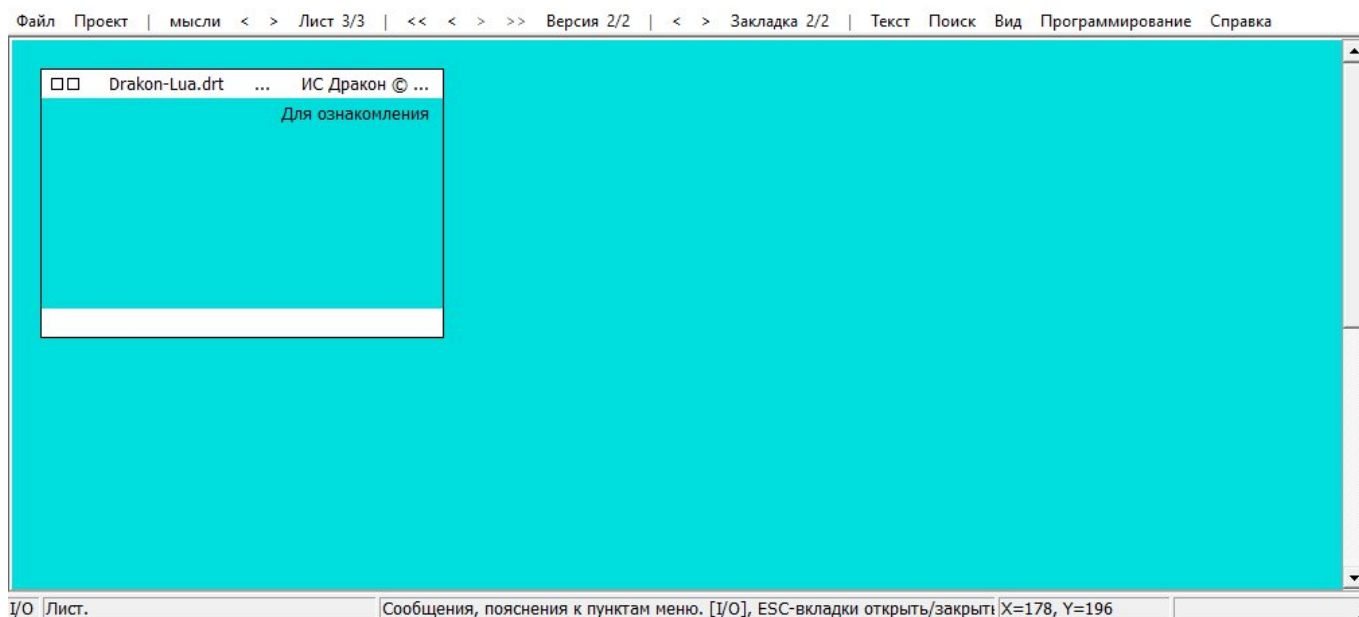


Рис. 4 Основное окно программы «ИС Дракон»

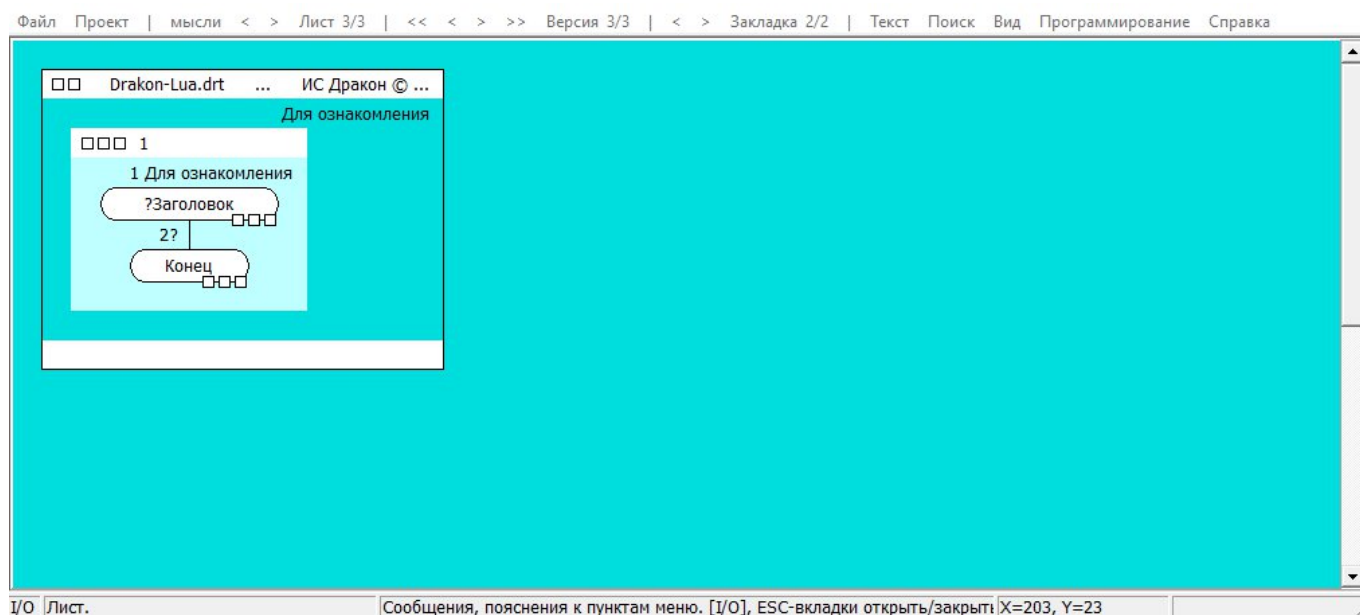


Рис. 5 Лист «Drakon-Lua.drt» со схемой «Примитив»

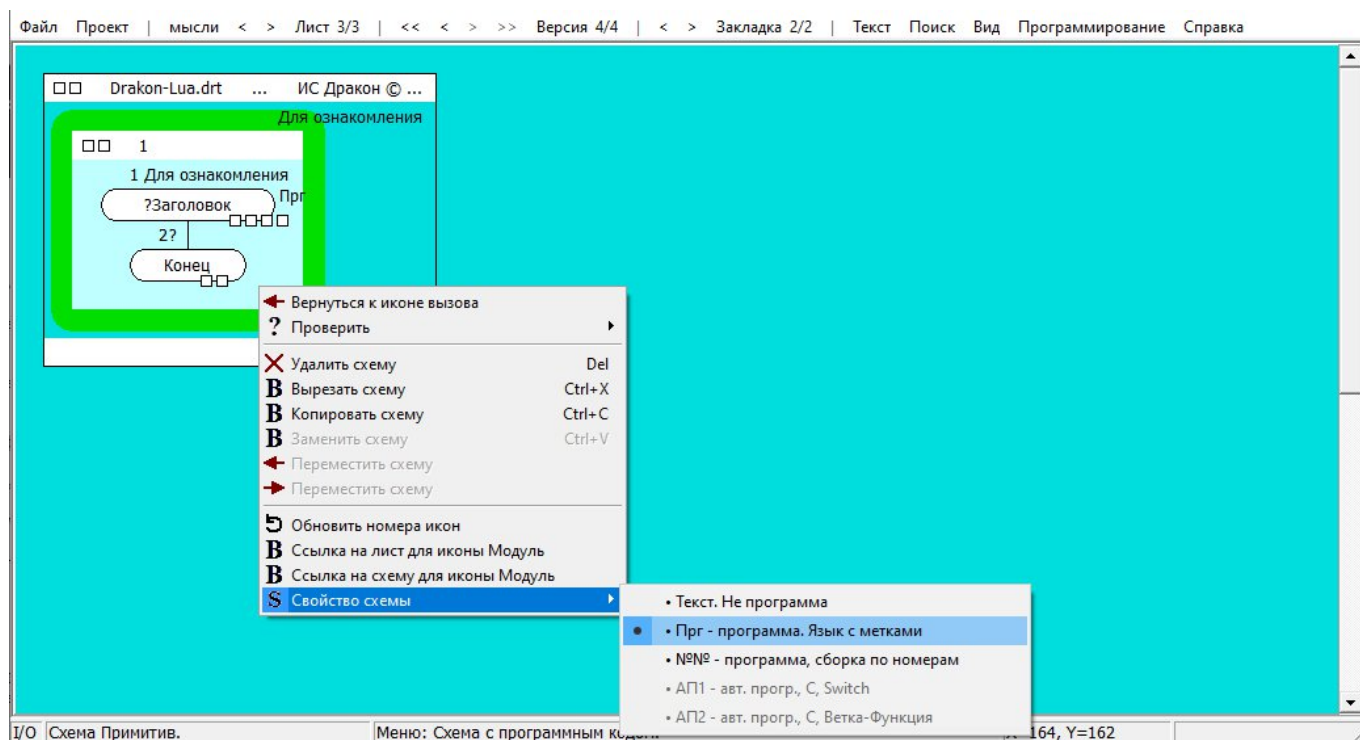


Рис. 6 Опции схемы «Примитив»: Прг. – программа. Язык с метками



Рис. 7 Добавление схемы «Гном»

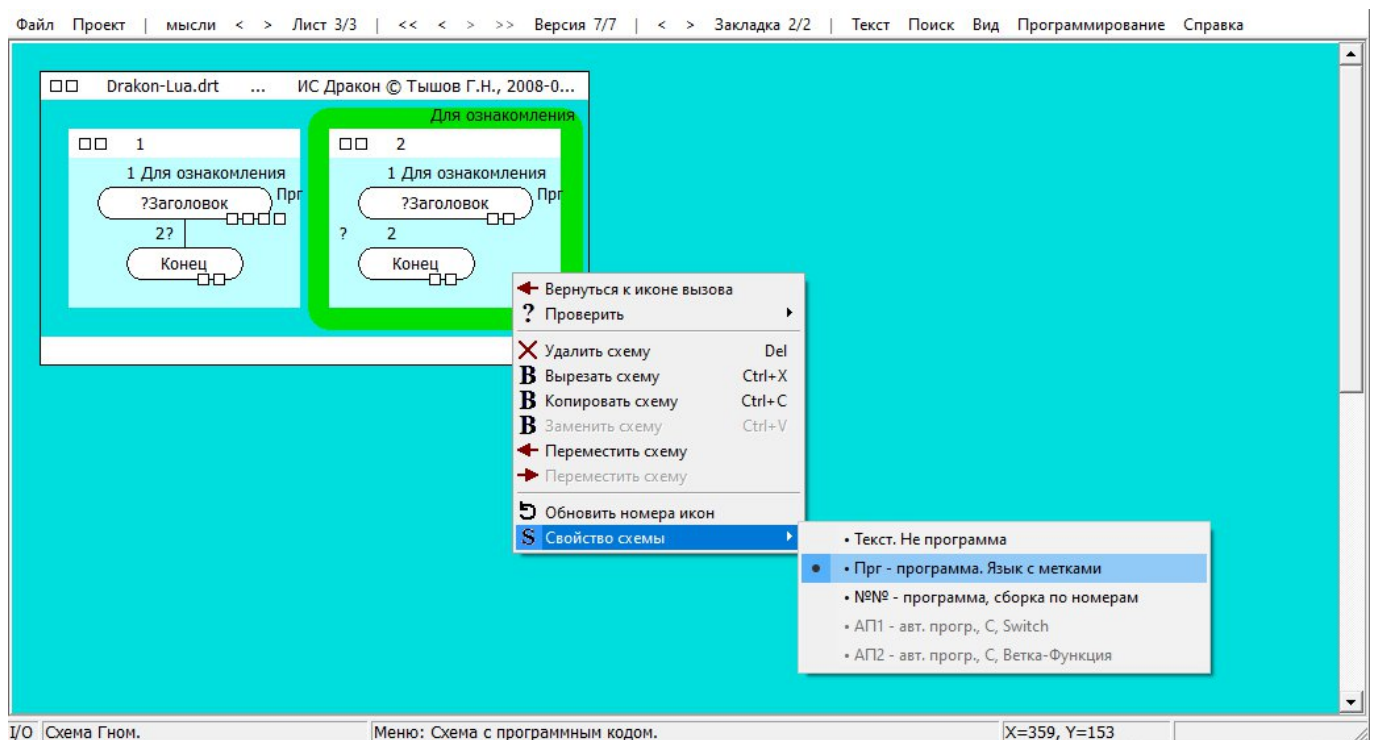


Рис. 8 Опции схемы «Гном»: Прг. – программа. Язык с метками

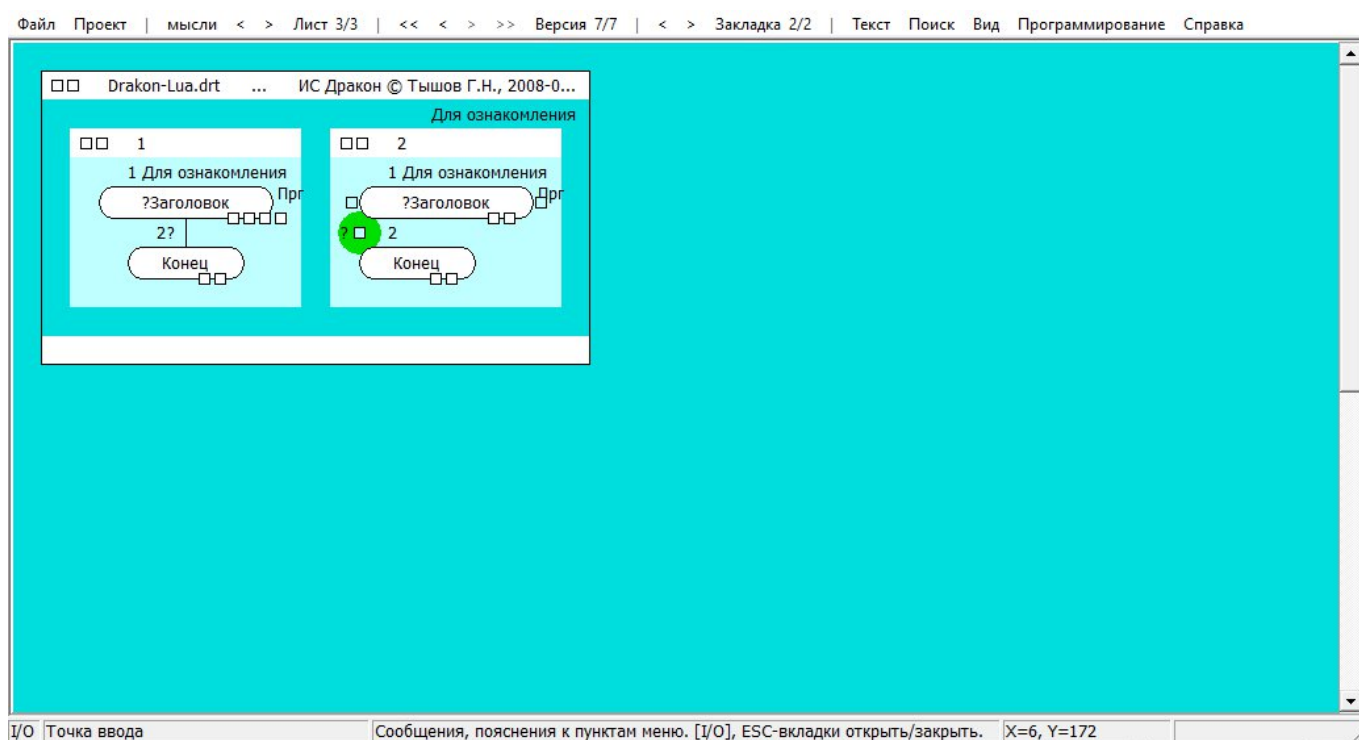


Рис. 9 Опция схемы «Гном» для добавления иконки «Модуль»

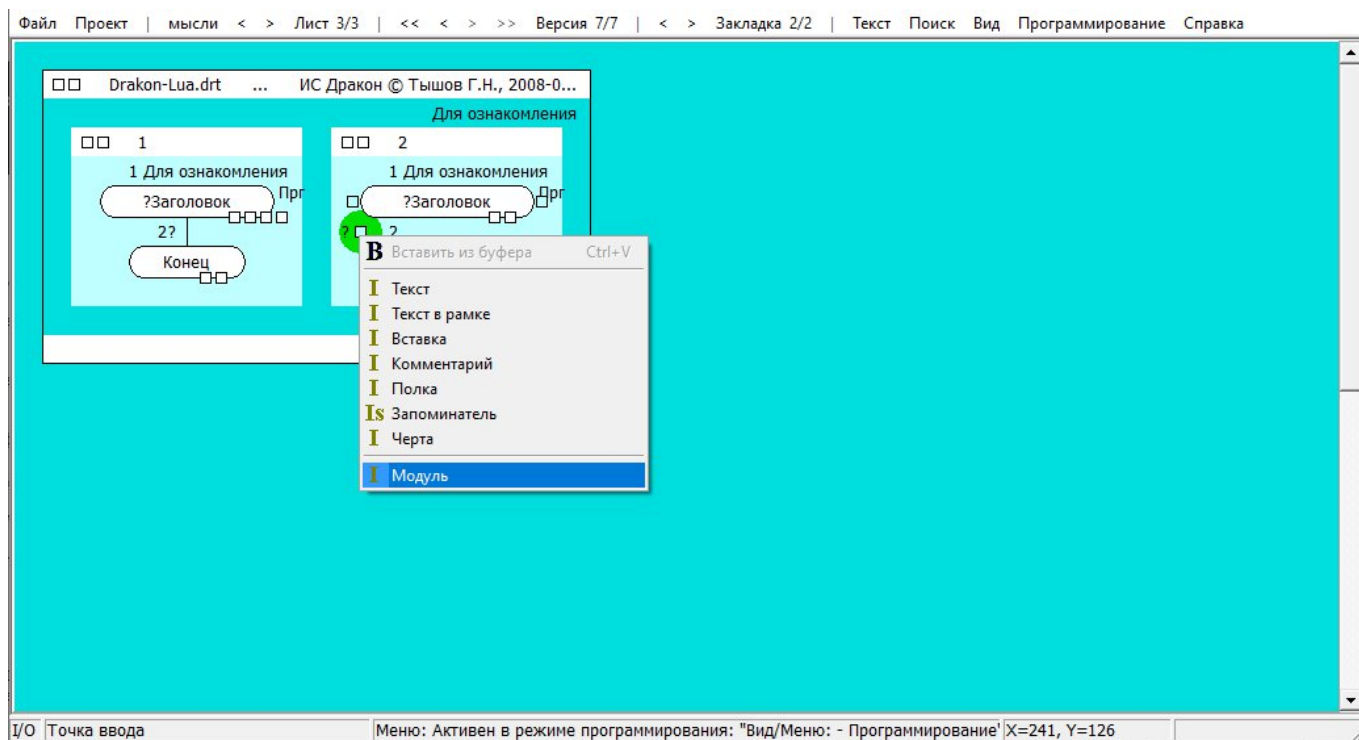


Рис. 10 Добавление иконки «Модуль»

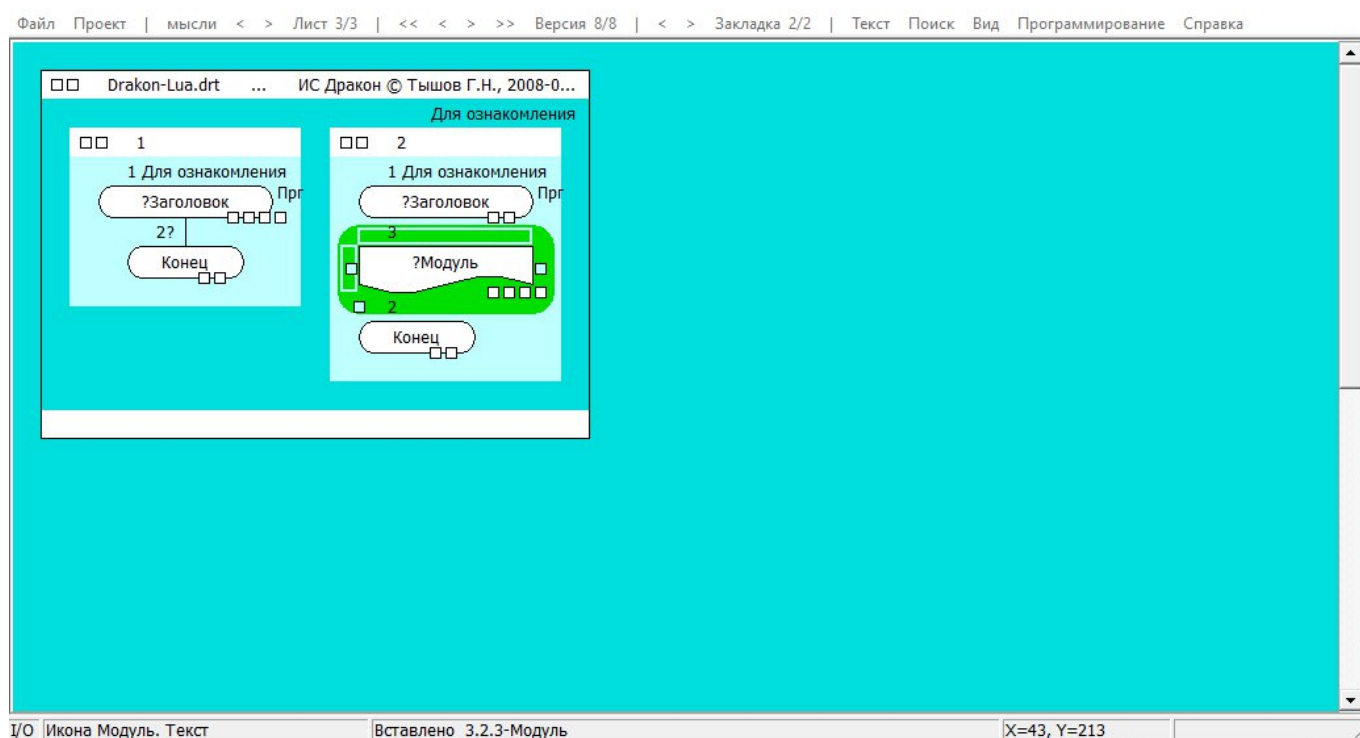


Рис. 11 Работа с иконкой «Модуль»

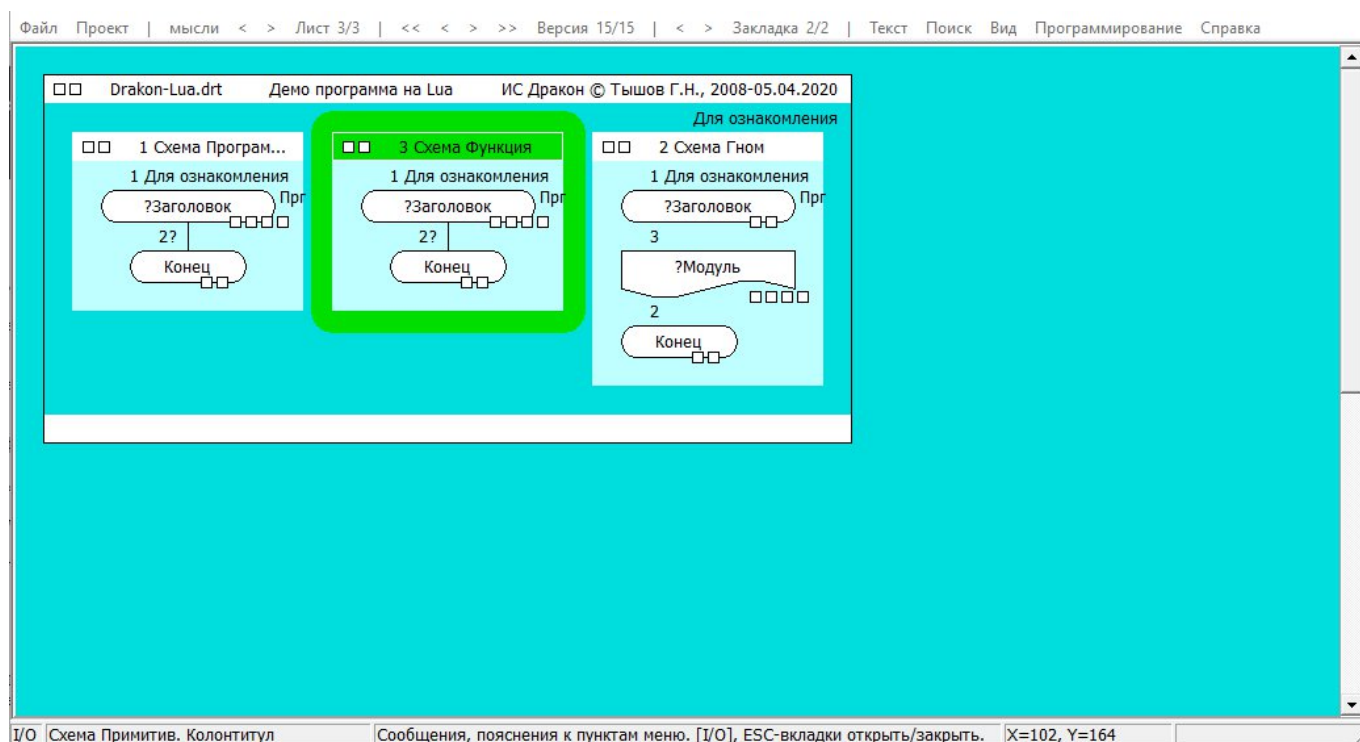


Рис. 12 Дополнительная схема для определения функции

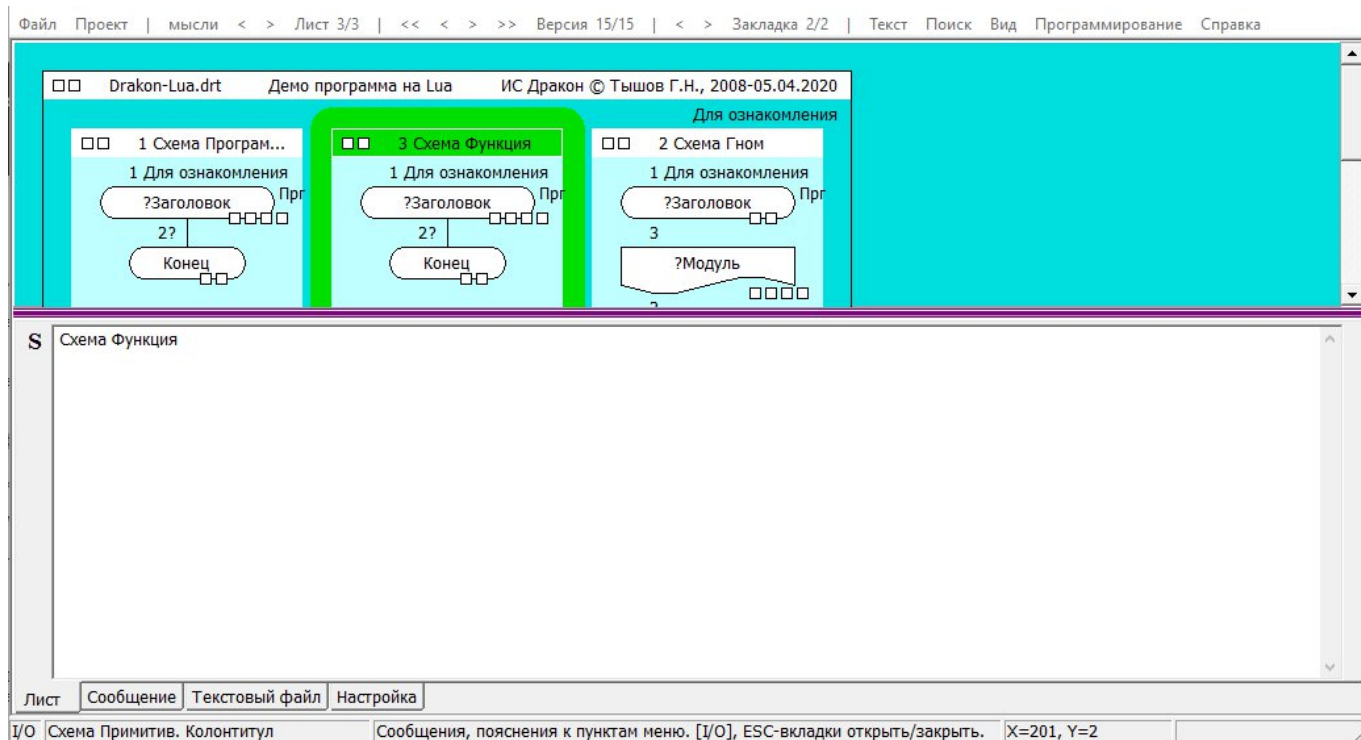


Рис. 13 Определение заголовка окна

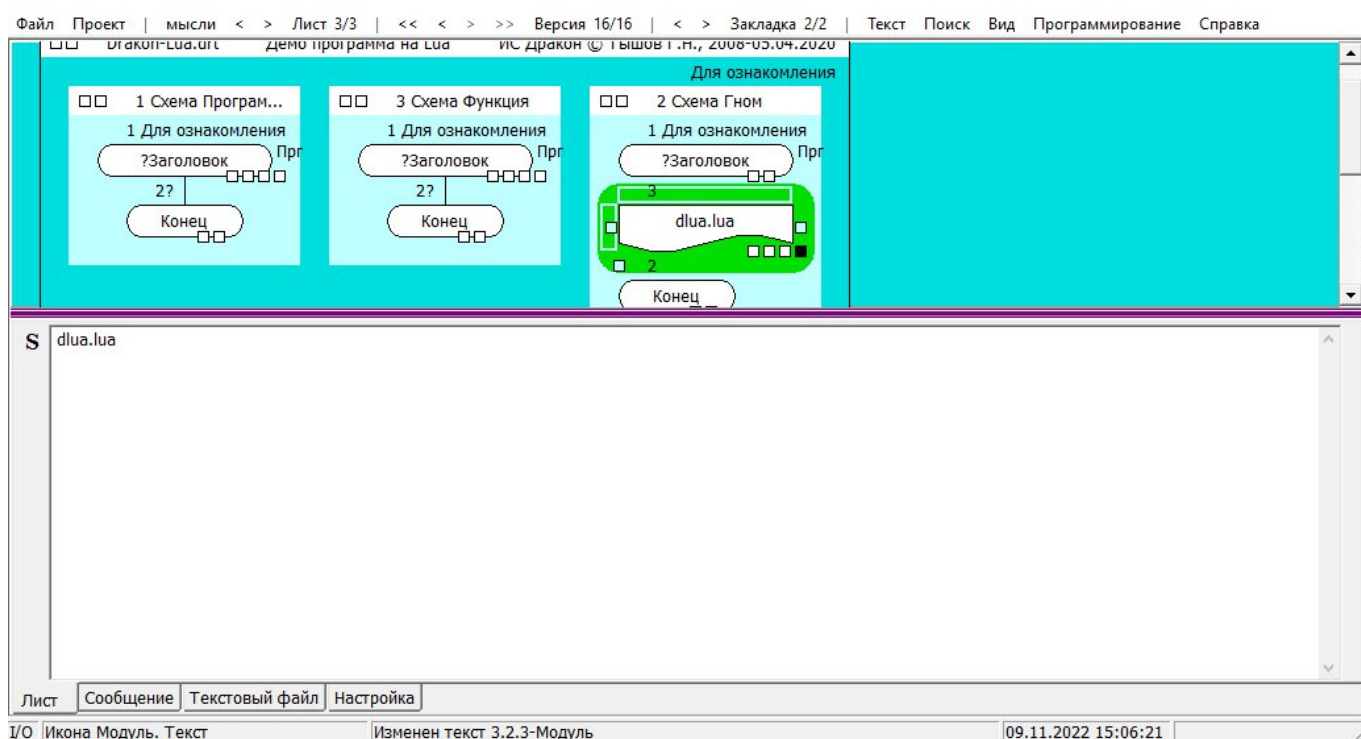


Рис. 14 Определение параметров выходного файла

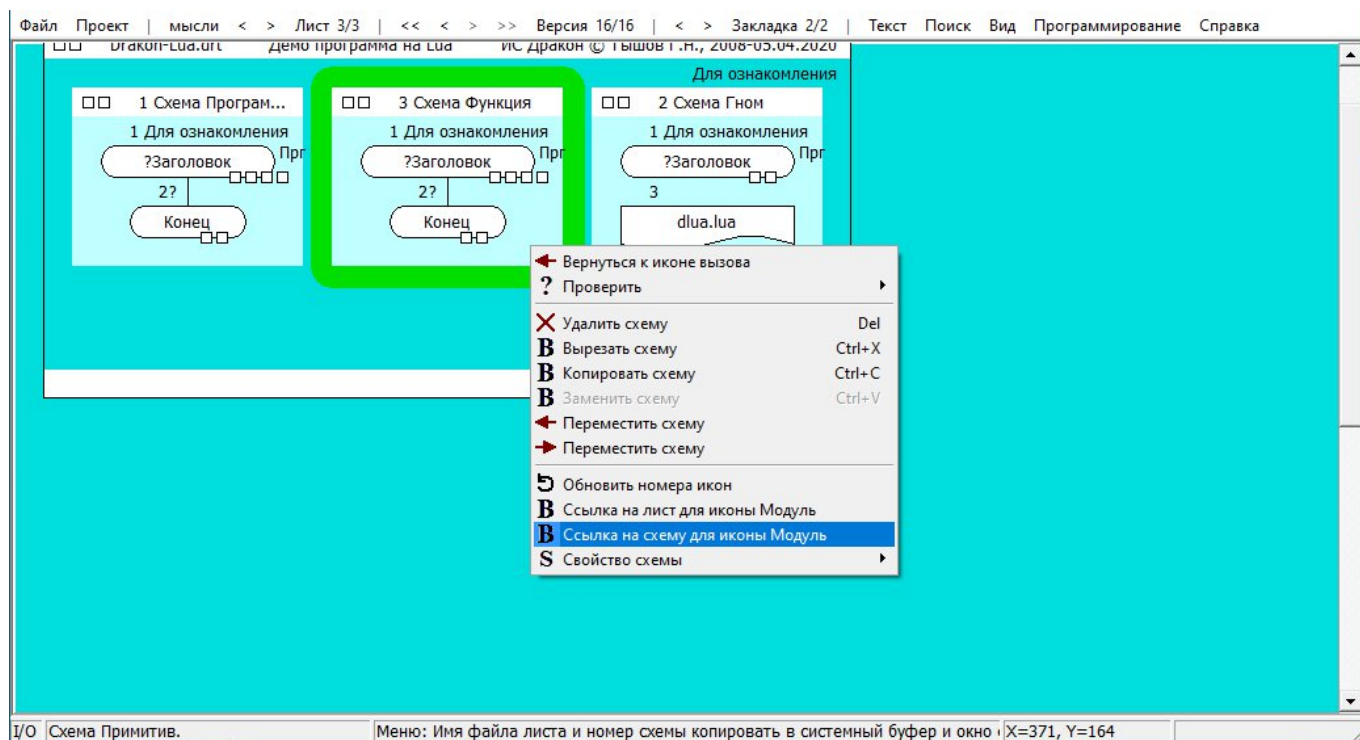


Рис. 15 Ссылка на схему для иконки «Модуль»

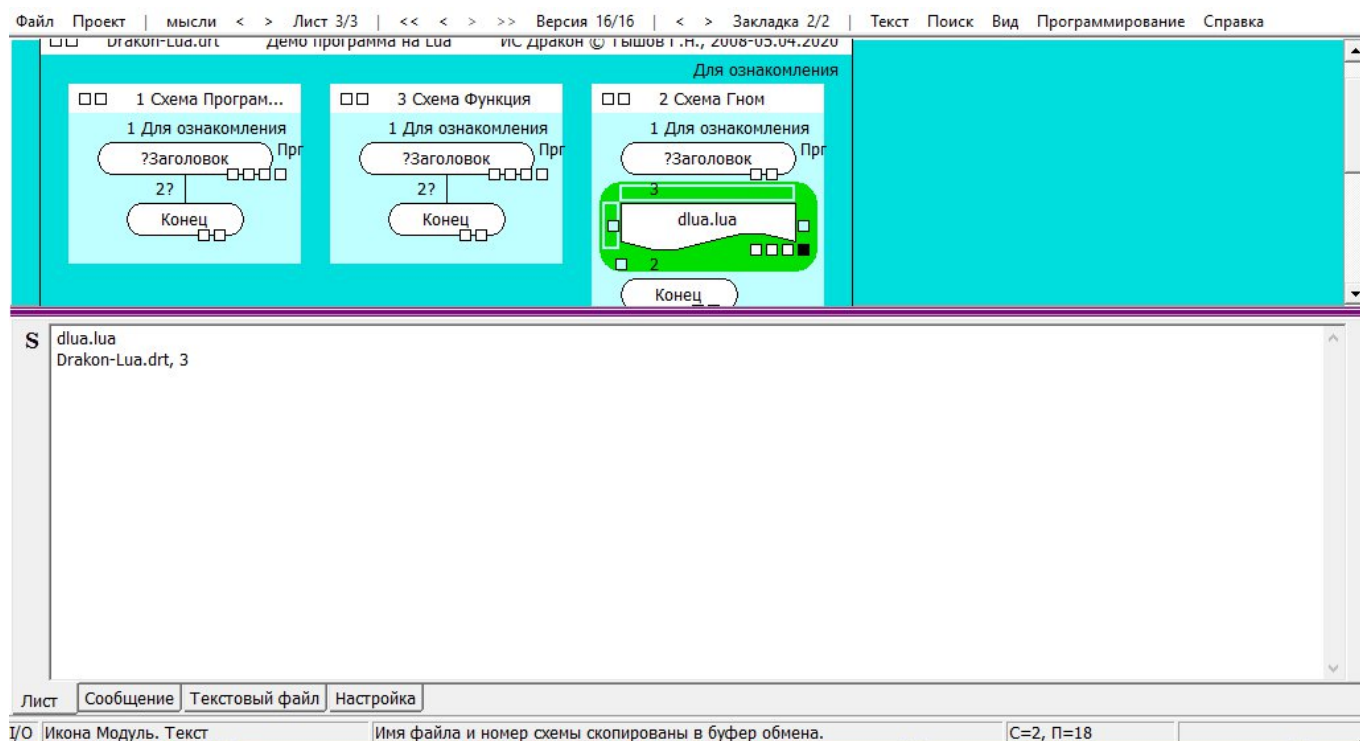


Рис. 16 Редактирование параметров иконки «Модуль»

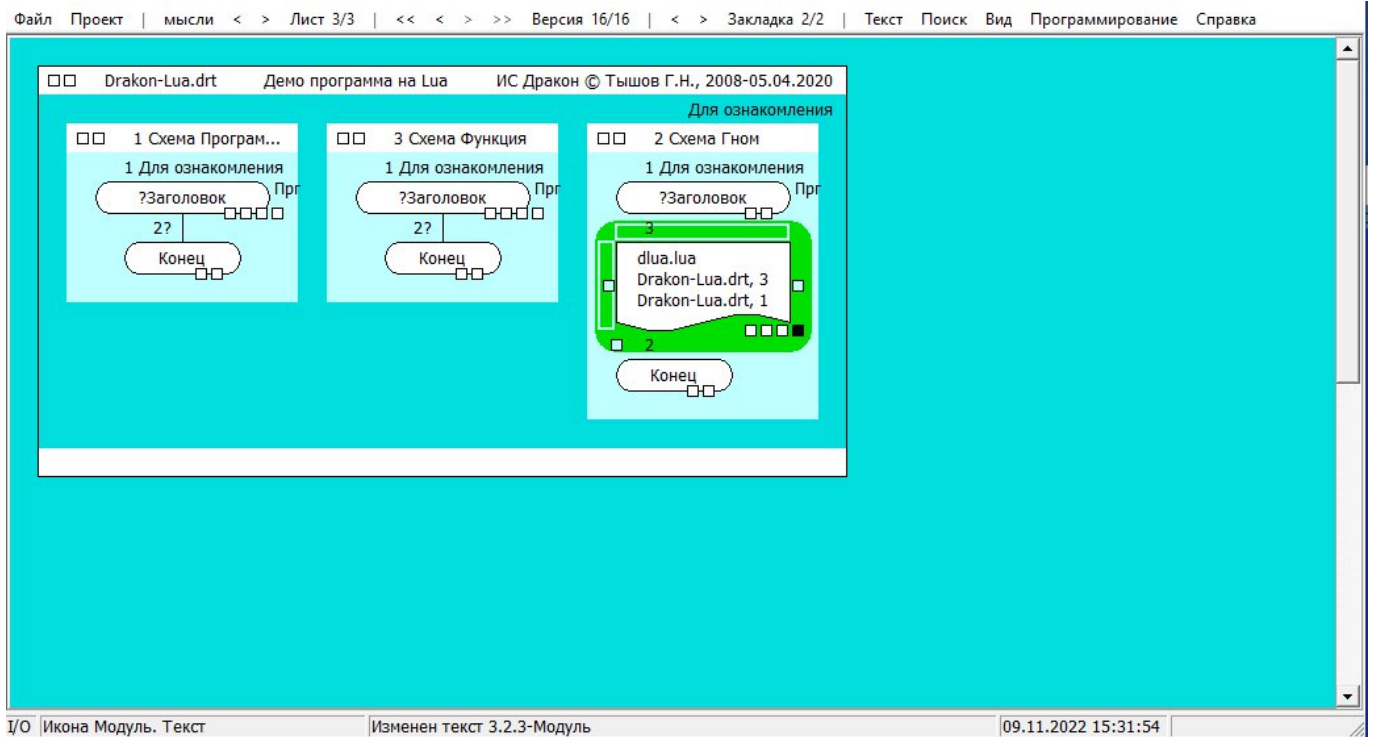


Рис. 17 Параметры схемы «Гном»

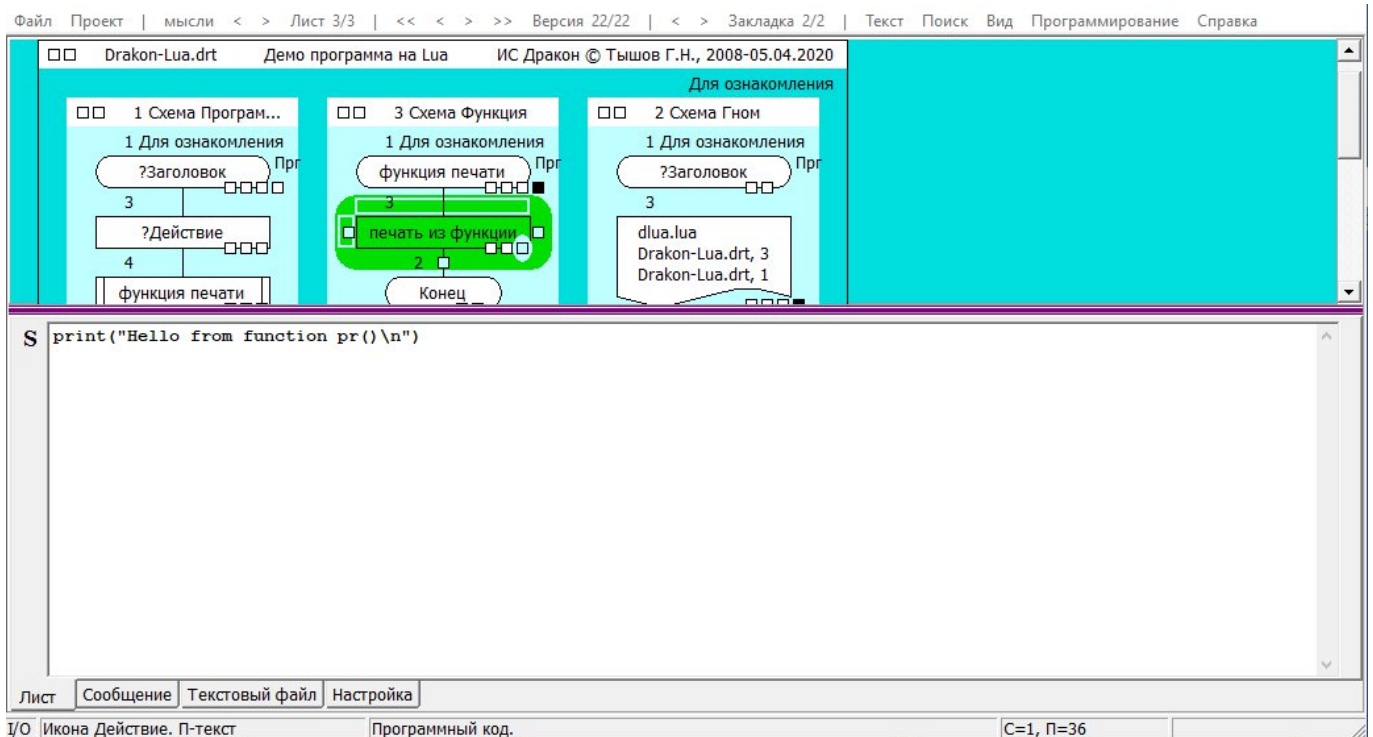


Рис. 18 Добавление программного кода в функции

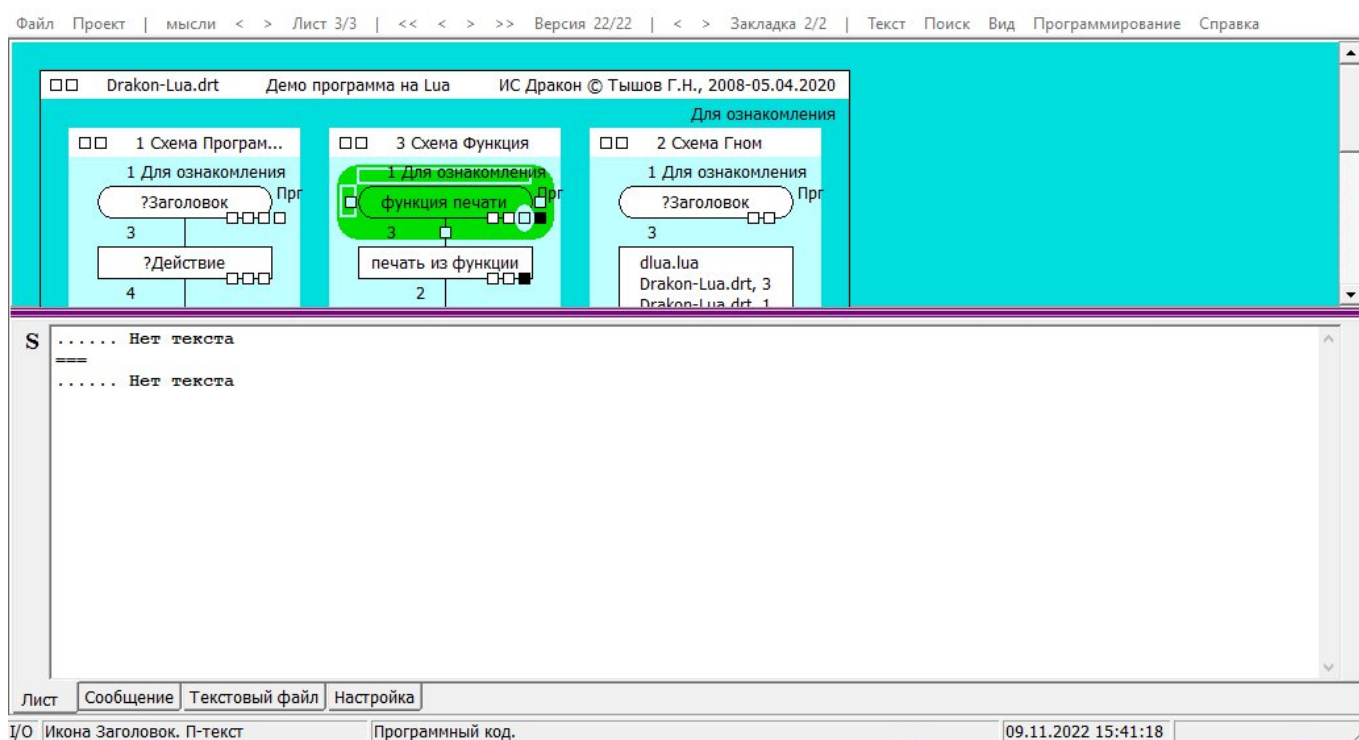


Рис. 19 Определение функции в соответствии с требованиями Lua

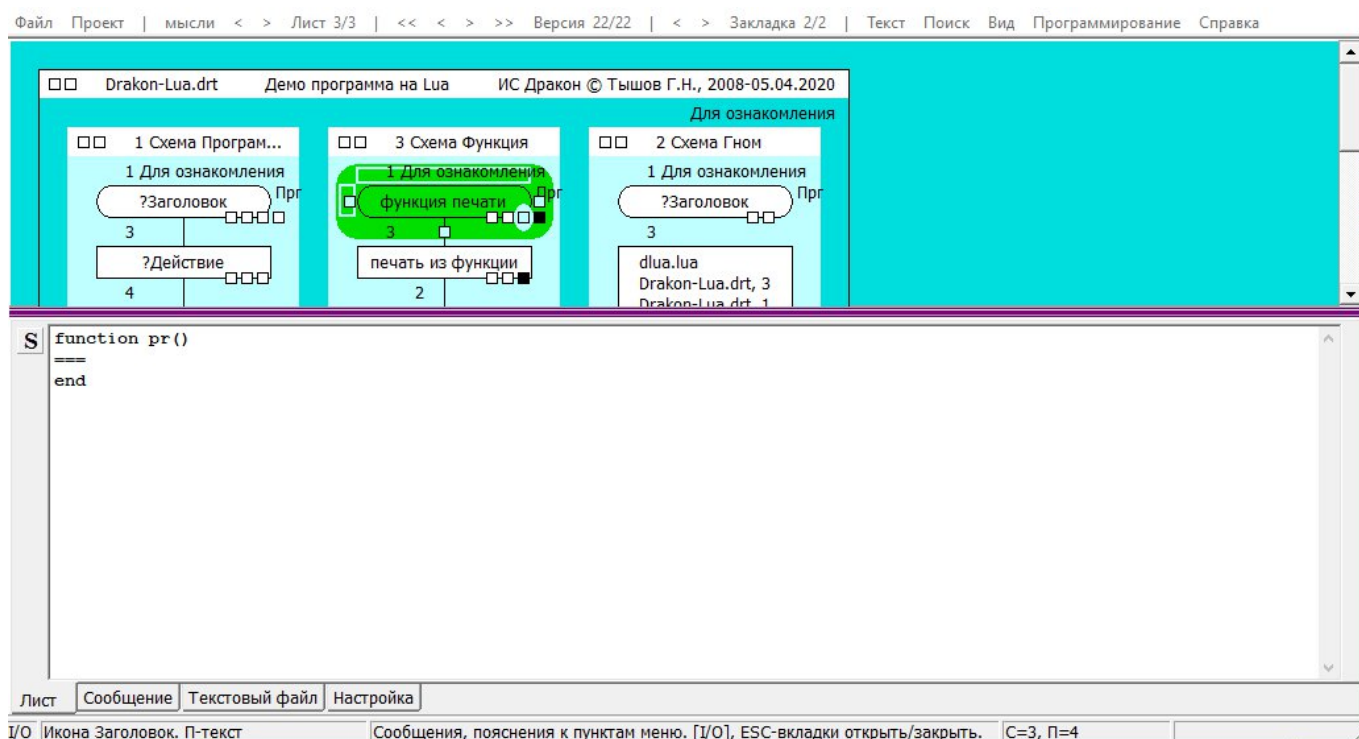


Рис. 20 Параметры функции pr()

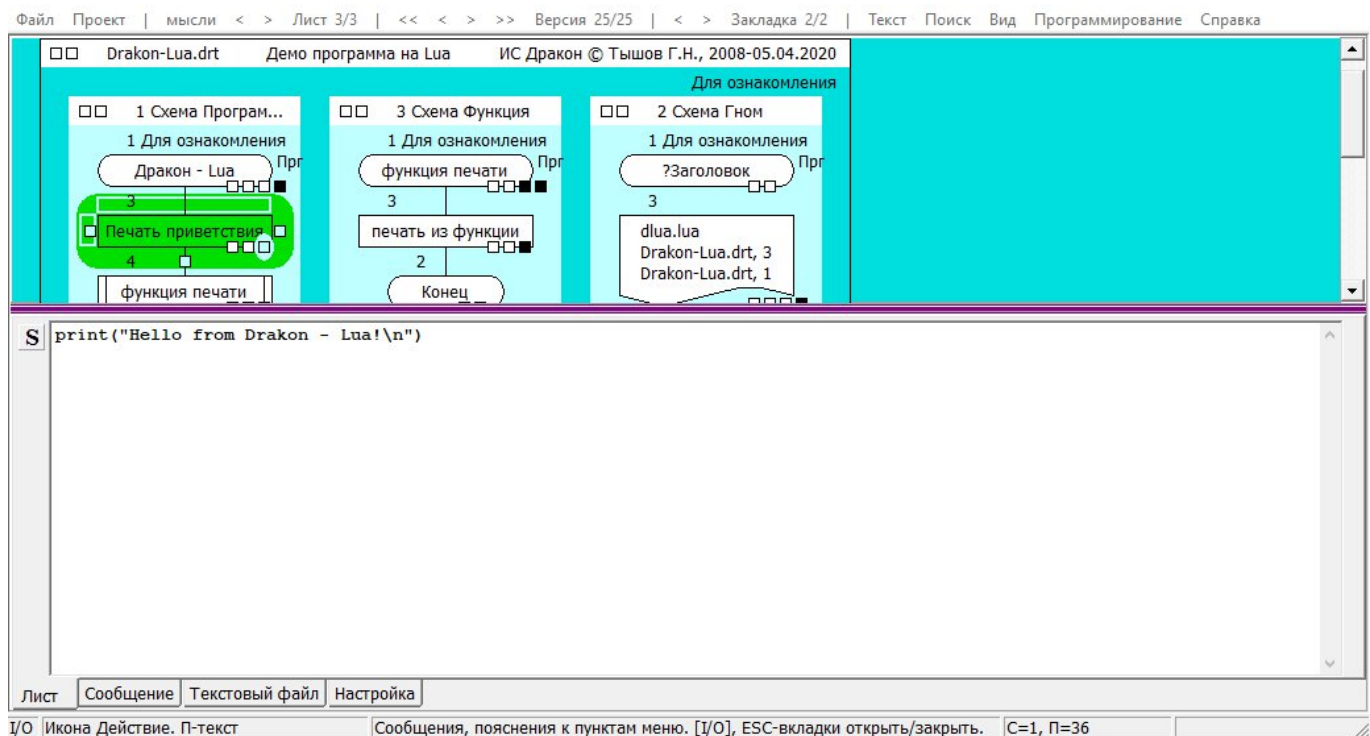


Рис. 21 Оператор Lua в основном модуле

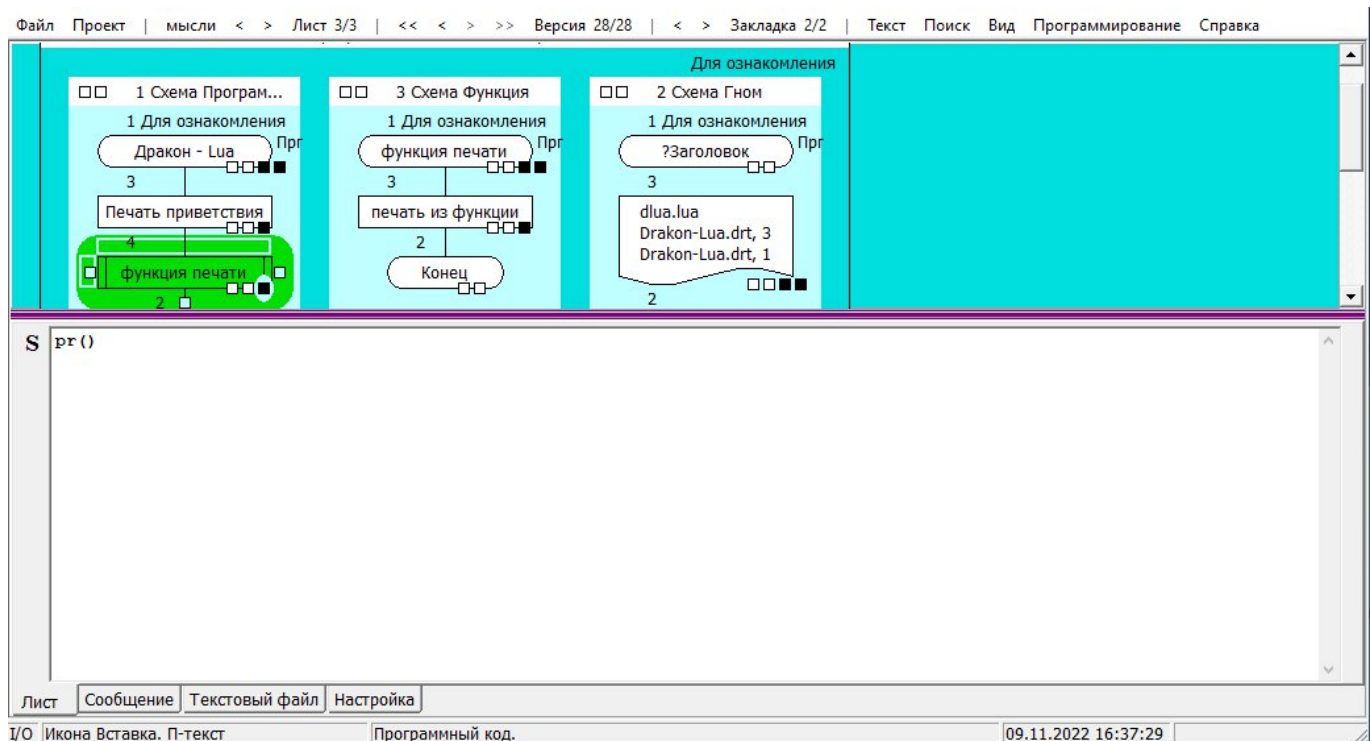


Рис. 22 Вызов функции pr() из основного модуля

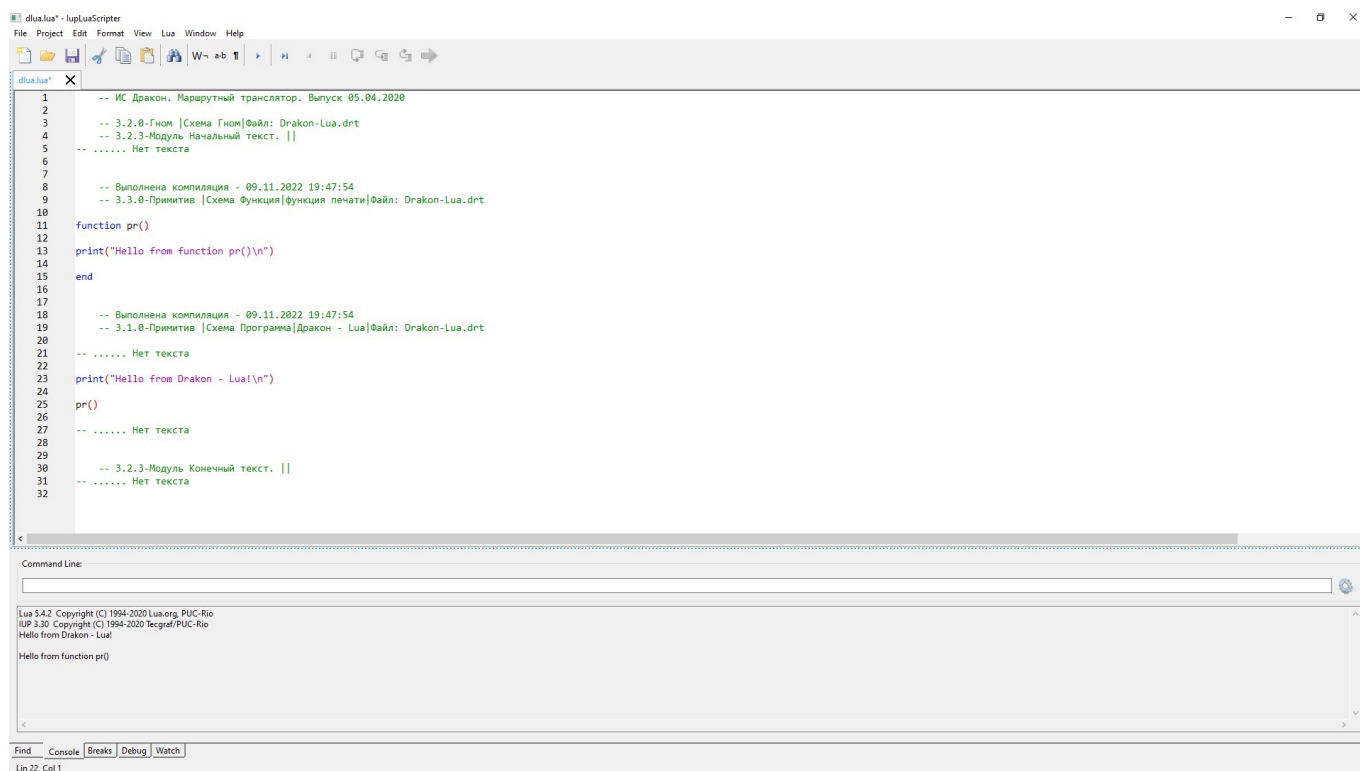


Рис. 23 Результат выполнения программы в lrpLuaScripter