

# Визуальное программирование на языке ДАЛВЯЗ 2

Барановский Дмитрий Владимирович  
г. Москва, 2023 г.

## Введение

В 2016 году я выложил на форуме сайта [oberoncore.ru](http://oberoncore.ru) в теме «Разное > Визуальное программирование > Алгоритмический язык ДАЛВЯЗ 2 Силуэтное программирование» материал с описанием языка ДАЛВЯЗ 2 и моих попыток использовать ДАЛВЯЗ 2 для программирования на языках высокого уровня.

Те опыты (в смысле использования) оказались неудачными – так программировать можно, но неудобно, интерфейс явно недоработан.

И вот в этом году мои размышления на тему «как приспособить блок-схемы для нужд программистов» привели к появлению новой версии программы `dalvjaz2: v.0.9.3.1`.

Текстовое описание языка ДАЛВЯЗ 2, приведенное в файлах `dalvjaz2_part1.pdf` и `dalvjaz2_part2.pdf`, выложенных в вышеуказанной теме, остается неизменным за исключением расширения функций определителя рисунка и стиля, который теперь получил название «определитель рисунка и комментария».

Программа `dalvjaz2 v.0.9.3.1` – это по-прежнему макетный вариант, хотя и более продвинутый, чем версия от 2016 года.

Наиболее существенные изменения по сравнению с предыдущей версией `dalvjaz2`:

- 1) программа `dalvjaz2` перенесена под Delphi 2010;
- 2) переработан интерфейс программы: теперь главное окно программы включает три постоянно находящихся на экране окна:
  - окно текстового редактирования записи;
  - окно редактирования схемы;
  - окно комментария / файла комментариев;

3) введены и поддерживаются одновременно три основных типа служебных комментариев:

- служебные комментарии для языка Паскаль Delphi;
- служебные комментарии для второго алгоритмического языка (сейчас в dal2.cfg заданы служебные комментарии для КП BlackBox, а комментарии для C/C++ закомментированы);
- служебные комментарии для псевдокода; такой формат комментариев для псевдокода задан для совместимости с псевдокодом файла morozko.txt (который я вместе с рисунками к нему выложил в архиве rk2.rar), но понятно, что псевдокод может использовать любой набор служебных комментариев, например псевдокод файлов kniga\_\*.txt (см. описание режима «читалка») использует паскалевские служебные комментарии;

4) реализованы гиперссылки вида <имя файла>#<номер записи> на другие записи файлов исходного кода и электронной документации, в том числе и на записи с другими типами служебных комментариев; для работы с вызываемой по гиперссылке записью запускается еще один экземпляр программы dalvjaz2;

5) комментарии для разрабатываемого исходного кода могут вводиться в файле комментариев проекта;

6) т.к. развитие программы dalvjaz2 происходит под влиянием моих размышлений на тему «видеоэкология чтения», то в новой версии dalvjaz2 реализован режим «читалка», реализующий функции читалки «СДЧ», выложенной мной в теме «Видеоэкология чтения 2023».

Далее по тексту материала рассматриваются вопросы создания в программе dalvjaz2 v.0.9.3.1 процедуры Sozdatx\_Knigu на языке КП BlackBox, находящейся в файле kp.txt в подкаталоге \rk рабочего каталога программы dalvjaz2.

## **1. Главное окно программы dalvjaz2 v.0.9.3.1**

Главное окно программы dalvjaz2 v.0.9.3.1 приведено на рис. 1.

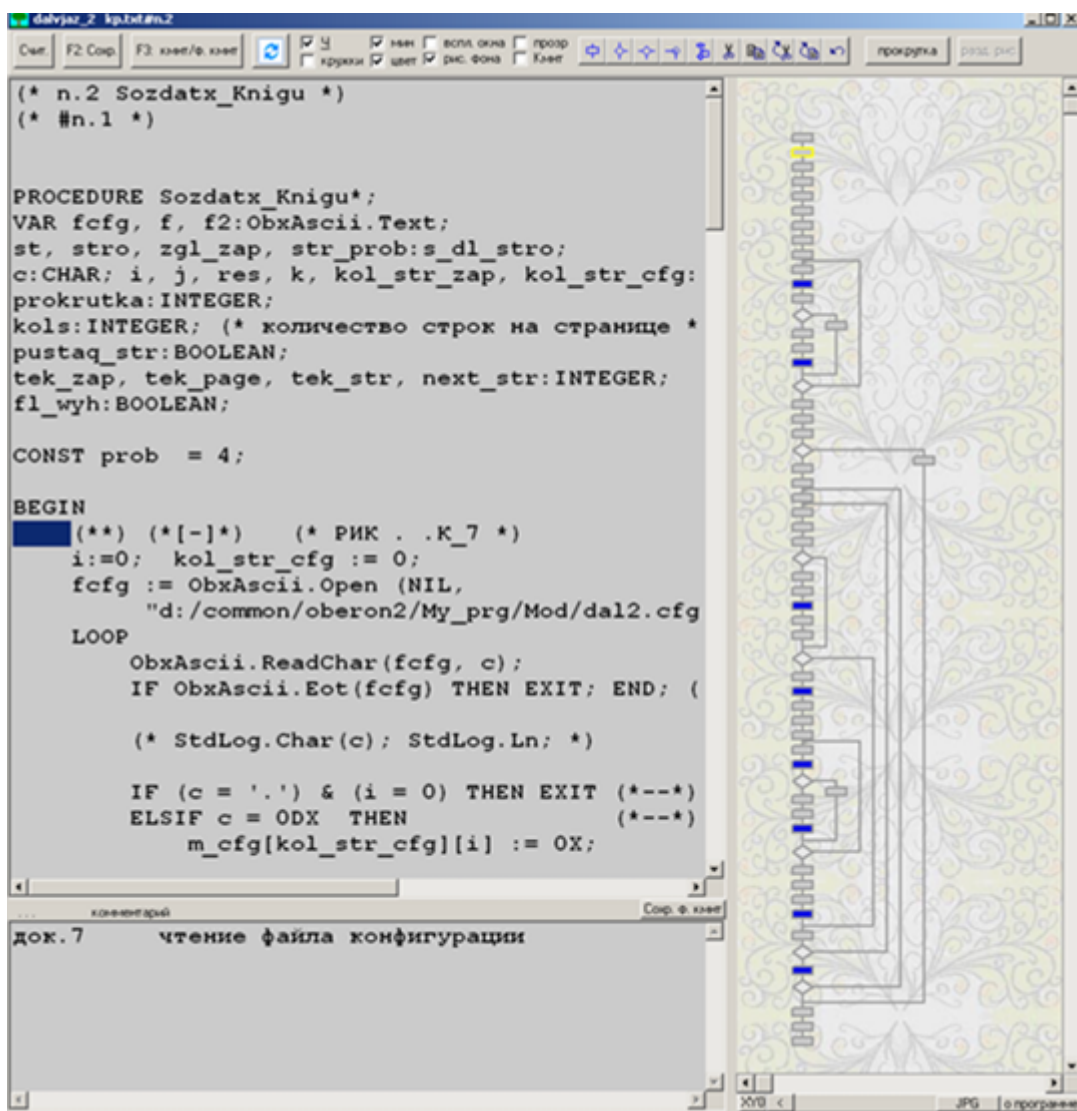


Рис. 1 – Главное окно программы dalvjaz2 v.0.9.3.1

Из рис. 1 видно, что клик мышью по элементу схемы выделяет этот элемент желтым, а в тексте записи синим выделяются начальные символы строки, соответствующей началу этого элемента.

Если элементу схемы присвоен комментарий, то он отображается в окне комментария.

Клик в окне текста записи на начальной строке элемента схемы приводит к выделению соответствующего элемента в окне «схема».


Результаты графического редактирования схемы немедленно отображаются в окне текста записи.


Для отображения результатов редактирования текста записи в окне «схема»

необходимо нажать кнопку  «Обновить схему».


Вообще говоря, нажимать кнопку «Обновить схему» рекомендуется перед выполнением любой операции графического редактирования, чтобы в текстовую запись не попал ее измененный старый вариант, неявно содержащийся в структуре данных схемы.

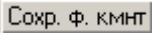
## 2. Управляющие элементы диалога

- кнопка  «Считать»; по ее нажатию выполняется открытие стандартного диалога чтения файла в каталоге «\_\_каталог\_исх\_кода» (задается в файле dal2.cfg), используемого для хранения файлов пользователя;

- кнопка  «Сохранить»; по ее нажатию (или по нажатию клавиши «F2» на клавиатуре) выполняется сохранение в файл редактируемой текстовой записи;


- поиск по текстовой записи вызывается нажатием сочетания клавиш «Ctrl+F» на клавиатуре.


- кнопка  «комментарий / файл комментариев»; по ее нажатию (или по нажатию клавиши «F3» на клавиатуре) в окне комментария отображается либо текущий комментарий, либо содержимое файла комментариев.

- кнопка  «сохранить файл комментариев»; по ее нажатию (или по нажатию клавиши «F8» на клавиатуре) выполняется сохранение файла комментариев.

Кроме того, сохранение файла комментариев выполняется при двойном клике на первой строке редактируемого текста комментария.

Если после двойного клика на первой строке редактируемого текста комментария кликнуть на начальной строке элемента схемы в окне «текст записи», то отредактированный комментарий будет присвоен этому элементу схемы, а в конце начальной строки элемента схемы появится определитель рисунка и комментария, соответствующий введенному комментарию.

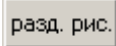
- кнопка  «Обновить схему»; по ее нажатию (или по нажатию клавиши «F4» на клавиатуре) выполняется обновление схемы в соответствии с текущим состоянием текста записи;


- флажок  «подчеркивание»: задание / сброс подчеркивания текста схемы;


- флажок ☐ кружки : обозначение кружками элементов «else» и «end» схемы;
- флажок ☒ мин : переход от минимизированного отображения схемы к полному и обратно (также по клавише «F5» клавиатуры);
- флажок ☒ цвет : переключение между отображением схемы в цвете и в оттенках серого;
- флажок ☐ вспл. окна : отображение для минимизированной схемы всплывающих окон с текстами (или комментариями) блоков схемы; для временного подавления режима всплывающих окон при движении указателя мыши по схеме следует нажать клавишу «Shift» на клавиатуре;
- флажок ☒ рис. фона : отображение для схемы рисунка фона; рисунок фона bg1.jpg находится в подкаталоге \img\_bg рабочего каталога программы dalvjaz2;
- флажок ☐ прозр : задание / отмена прозрачности блоков схемы; при взведенном флажке «прозр» текст элементов схемы отображается на фоновом рисунке или на фоне схемы;
- флажок ☐ Кмнт : задание / отмена отображения комментария вместо текста элемента схемы во всплывающем окне элемента схемы;
- кнопка  : задание более или менее плавной прокрутки при перемещении по схеме; перемещение по схеме выполняется:


- 1) с помощью горизонтальной и вертикальной полос прокрутки схемы;
- 2) путем прокрутки колесика мыши;
- 3) путем наведения указателя мыши на свободный от элементов участок схемы с последующим нажатием правой кнопки мыши и перетаскиванием схемы в нужном направлении; рекомендуется сначала сдвинуть схему по горизонтали или вертикали примерно на сантиметр, а затем использовать колесико мыши.


Плавность прокрутки задается параметрами «\_\_диапазон\_прокрутки» и «\_\_диап\_прокрутки\_книга» в файле dal2.cfg. Как правило, для просмотра книги требуется более плавная прокрутка, чем при работе со схемой, хотя и здесь возможны варианты. При загрузке книги более плавная прокрутка задается автоматически.

- кнопка  (для режима «читалка»): изменение разделительного рисунка для читалки; разделительные рисунки находятся в подкаталоге \img\_bg, начиная с bg3.jpg (их название становится понятным при чтении книги с вертикальной прокруткой);

- кнопка  : установка координат схемы в 0, используется для быстрого возврата к началу координат схемы;

- кнопка  : используется для возврата к предыдущей ветке силуэта после того, как указатель мыши был наведен на блок перехода к другой ветке с последующим нажатием правой кнопки мыши;

- кнопка  : используется для создания jpg-файла с изображением схемы; при использовании этой возможности для больших схем рекомендуется отключать рисунок фона, т.к. он не охватывает всей площади схемы (плиточное копирование рисунка фона по всей площади большой схемы пока не реализовано);


- кнопка  : вывод информации о программе.

### 3. Кнопки графического редактирования схемы

Работа с кнопками графического редактирования схемы



подробно описана в файле dalvjaz2\_part1.pdf.

К имеющемуся там описанию можно добавить, что для удобства вставки элемента «\* иначе» (кнопка ) следует предварительно установить флажок «кружки», чтобы убедиться, что элемент «\* иначе» вставлен в схему.

### 4. Создание процедуры Sozdatx\_Knigu

В результате выполнения процедуры Sozdatx\_Knigu должен быть сформирован файл следующего вида (книга с вертикальной прокруткой):

```
(* n_1 1-28 *)  
(**1*)
```

"Важным при чтении является движение взгляда не по самим буквам, а по

белым полоскам, разделяющим строки. Эти полосы называются БЕЛЫМИ

СТРОКАМИ. Движение взгляда по белой строке позволяет избежать ненужного напряжения зрения, появляющегося при попытке разглядеть буквы. Ее использование при чтении способствует излечению пресбиопии (старческого

```
(** [-] // РИК .bg3 *)
```

```
. . .
```

```
(**6*)
```

только представить белую строку, но и расширить ее до границ страницы.

Как только это будет достигнуто, зрение улучшится. Тогда процесс чтения

пойдет быстро, легко и без каких-либо усилий . . . ."

```
(** выход *)
```

```
(** КОНЕЦ *)
```

В вышеприведенном тексте есть следующие служебные строки:

- «(\* n\_1 1-28 \*)» - заголовок записи с указанием входящих в нее строк текстового файла. Для примера я привел текст книги для маленького текстового файла. Для больших текстовых файлов книга состоит из нескольких записей по 1000 строк и одной записи размером меньше 1000 строк с окончанием файла;

- «(\*\*1\*)» и т.п. - задание номеров страниц для книги;

- «(\*\* [-] // РИК .bg3 \*)» - задание разделительного рисунка для текста книги;

- «(\*\* выход \*)» - действие «\* выход»;

- «(\*\* КОНЕЦ \*)» - действие «\* конец записи».

В файле dal2.cfg для читалки выделена отдельная секция (рис. 2).

```
;      параметры читалки:
;
__количество_строк_на_странице "5"      ; от 5 до 10
__пустая_строка_между_строками "ДА"     ; "ДА"/"НЕТ"
__прокрутка "1"                        ; 1 - по горизонтали 2 - по вертикали
```

Рис. 2 – Параметры читалки в файле dal2.cfg

Создадим для процедуры Sozdatx\_Knigu упрощенный файл dal2.cfg:

```
; список считываемых строк:
```

```
; 1. заголовок записи
```

```

; 2. строка основного условия для __прокрутка = 1
; 3. строка дополнительного условия для __прокрутка = 1
; 4. строка действия для __прокрутка = 1
; 5. строка окончания условия для __прокрутка = 1
; 6. строка рисунка
; 7. действие "выход"
; 8. действие "КОНЕЦ"
; 9. начало блока действия
; 10. закрывающий комментарий
; 11. количество строк на странице: от 5 до 10
; 12. пустая строка между строками текста Y/N
; 13. прокрутка: 1 - по горизонтали 2 - по вертикали
;
(* n_
if [-]
end else if [-]
(** [-] *)
end; //
(** [-] // РИК .bg3 *)
(** ВЫХОД*)
(** КОНЕЦ*)
(**
*)
5
Y
2
.

```

С заделом на будущее этот файл включает и параметры для книги с горизонтальной прокруткой.

При вводе текста процедуры будем использовать графическое редактирование для создания структуры процедуры, а текстовое редактирование будем использовать для наполнения созданной структуры реальным содержанием.

Итак, сначала следует считать файл конфигурации (разместим все действия по чтению конфигурации в одном элементе схемы):

```

(**) (*[-]*)    (* РИК . .K_7 *)
i:=0; kol_str_cfg := 0;
fcfg := ObxAscii.Open (NIL,

```



```

        "d:/common/oberon2/My_prg/Mod/dal2.cfg");
LOOP
    ObxAscii.ReadChar(fcfg, c);
    IF ObxAscii.Eot(fcfg) THEN EXIT; END; (*--*)

    (* StdLog.Char(c); StdLog.Ln; *)

    IF (c = '.') & (i = 0) THEN EXIT (*--*)
    ELIF c = 0DX THEN (*--*)
        m_cfg[kol_str_cfg][i] := 0X;
        INC(kol_str_cfg); i := 0;
    ELSE (*--*)
        m_cfg[kol_str_cfg][i] := c; INC(i);
    END ;
END ;

```

В вышеприведенном тексте есть следующие служебные строки:

- «(\*\*)» - начало элемента «\* действие»;
- «(\*[-]\*)» - вывод строки элемента маскируется;
- «(\* РИК . .K\_7 \*)» - определитель рисунка и комментария для элемента;
- «(\*--\*)» - признак подавления ключевого слова;
- «END ;» - подавление служебного комментария «END».

На рис. 3 – рис. 8 приведен алгоритм процедуры Sozdatx\_Knigu, полученный копированием содержимого окна «схема» в режиме полного отображения схемы с установленным флажком «Кмнт».

Но реальное графическое редактирование процедуры удобнее выполнять для минимизированной схемы, как это показано на рис. 1.

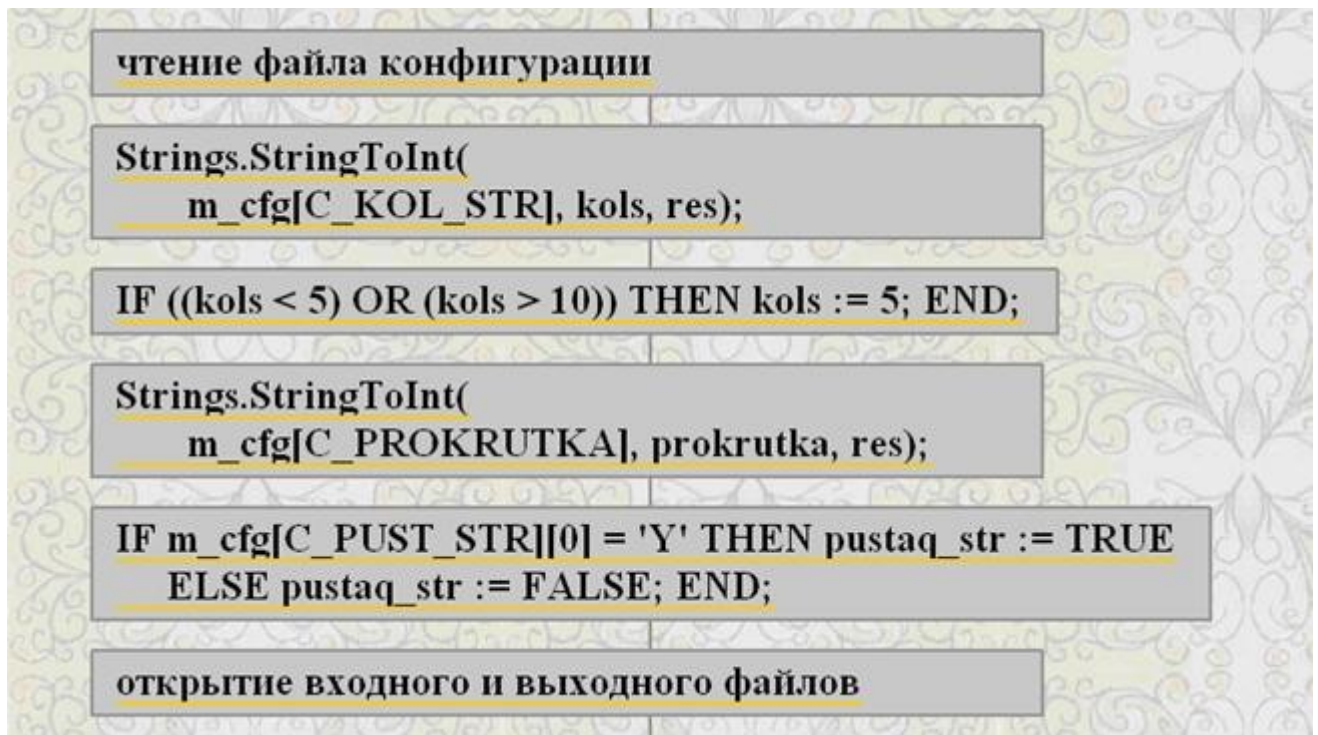


Рис. 3 – Начало алгоритма процедуры Sozdatx\_Knigu

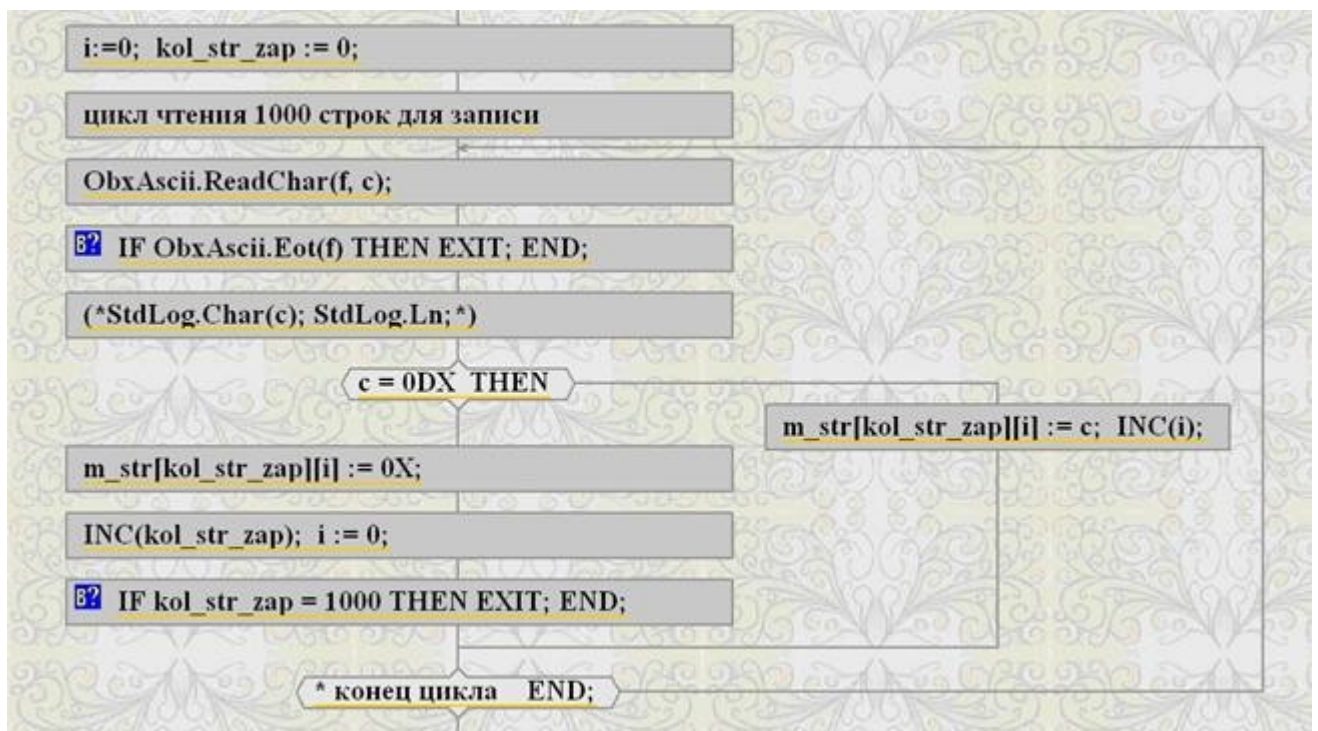


Рис. 4 – Продолжение алгоритма процедуры Sozdatx\_Knigu

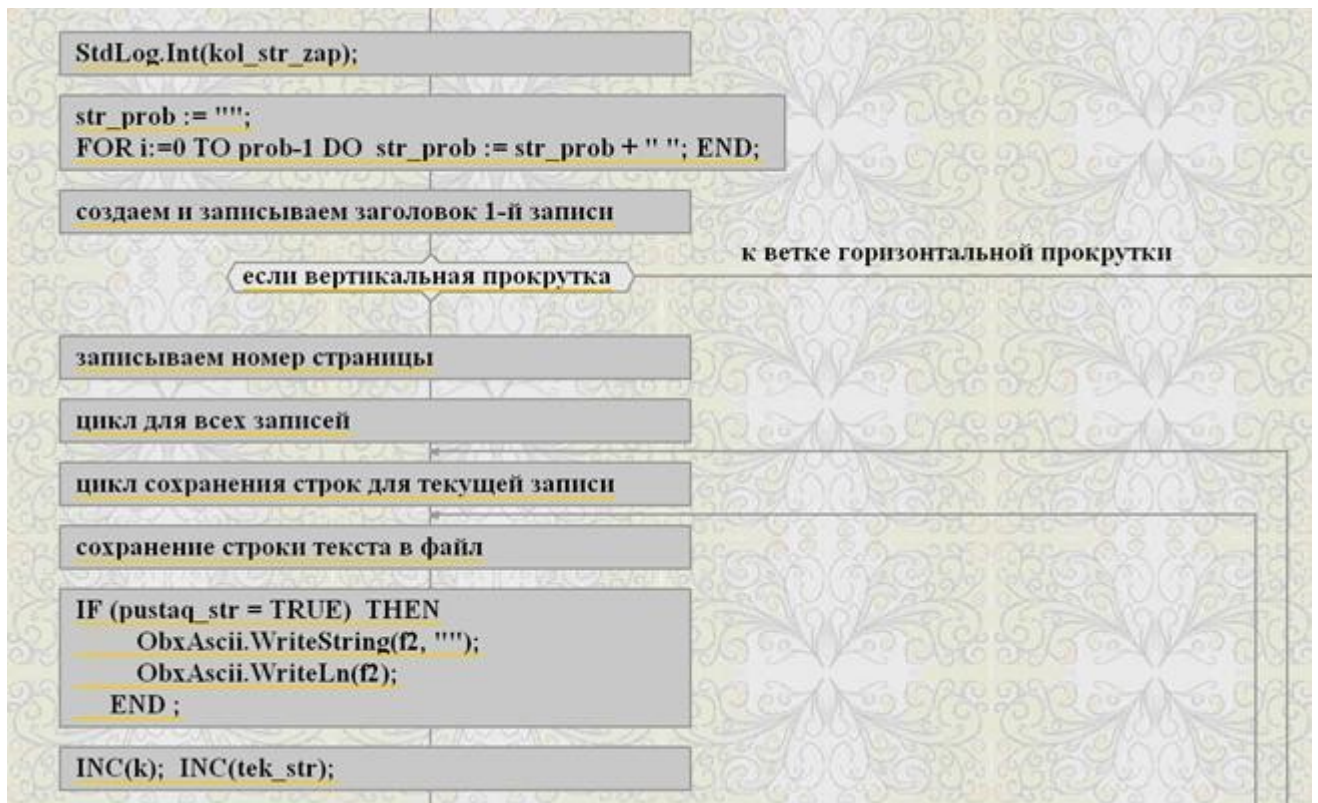


Рис. 5 – Продолжение алгоритма процедуры Sozdatx\_Knigu

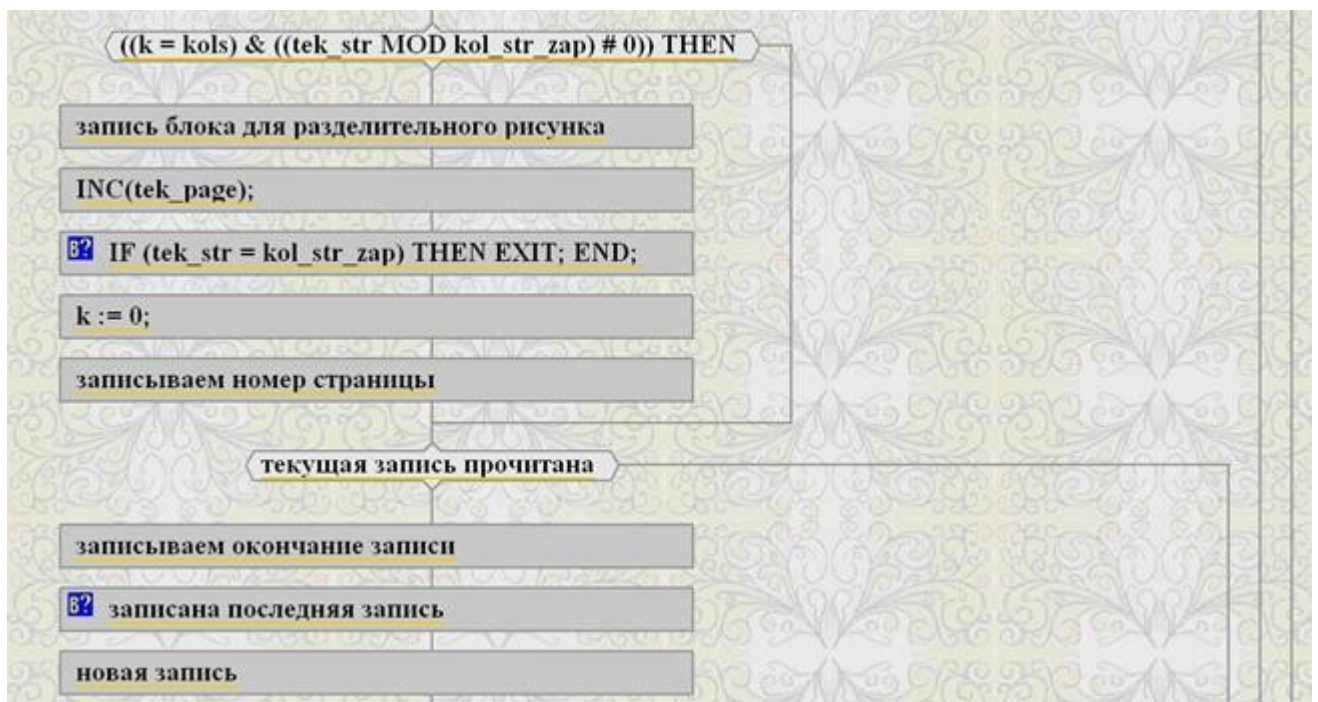


Рис. 6 – Продолжение алгоритма процедуры Sozdatx\_Knigu



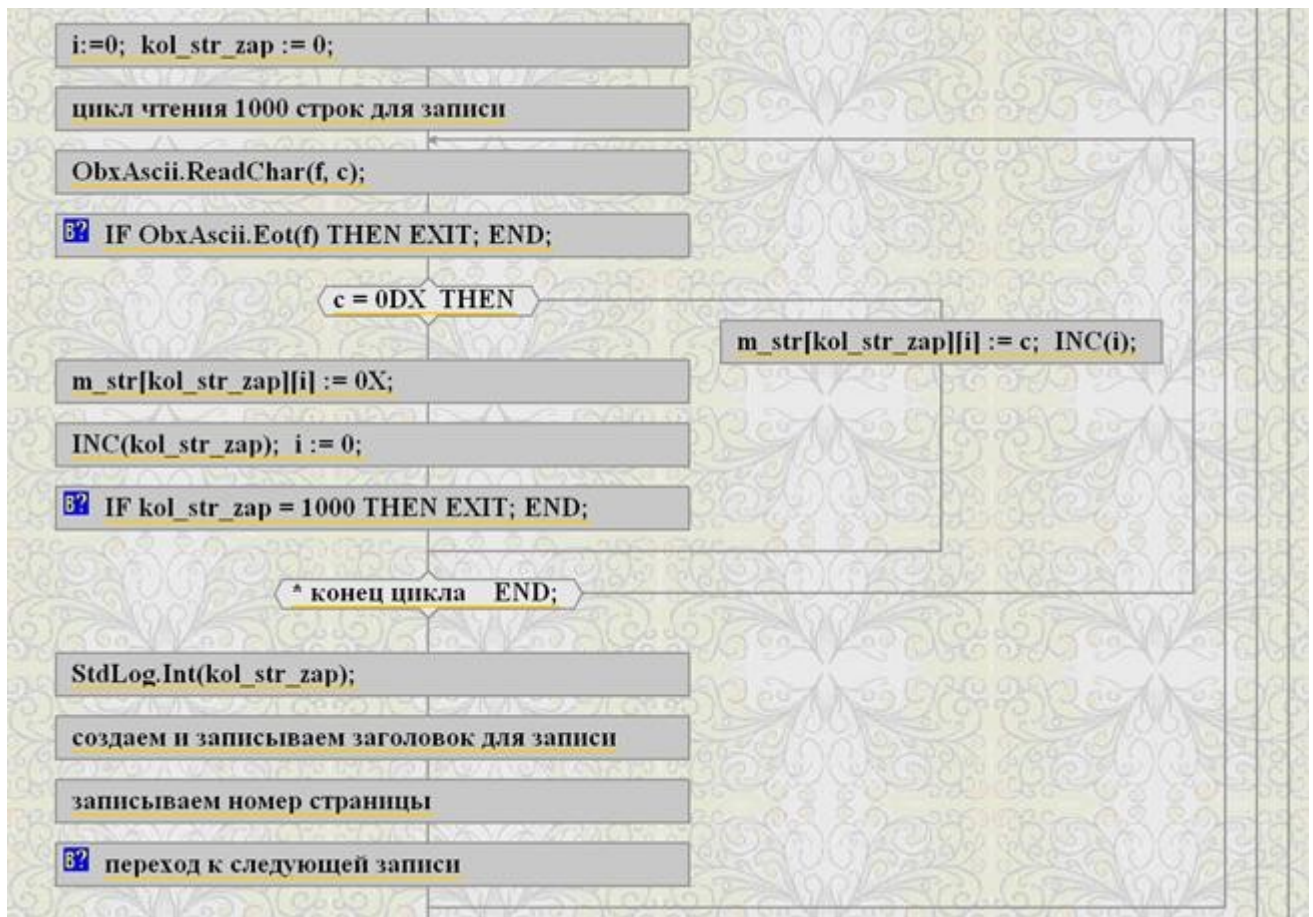


Рис. 7 – Продолжение алгоритма процедуры Sozdatx\_Knigu

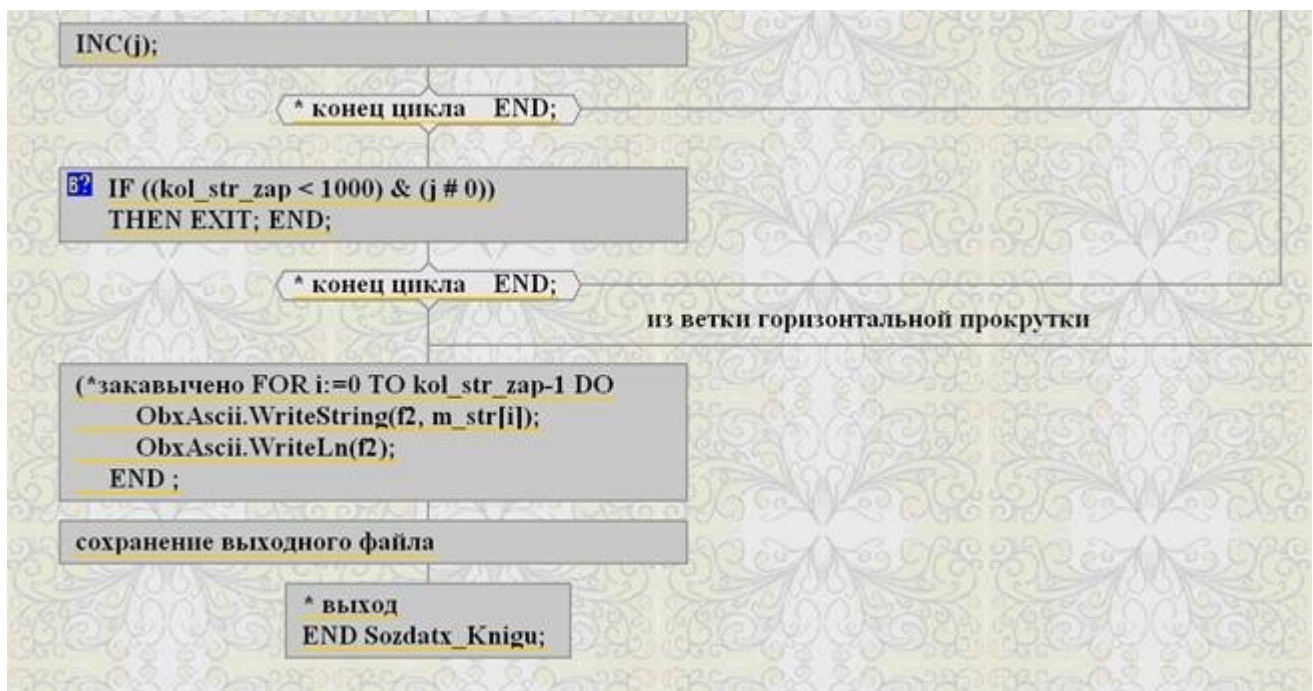


Рис. 8 – Окончание алгоритма процедуры Sozdatx\_Knigu

На рисунке 9 приведено копирование цикла чтения 1000 строк для записи.

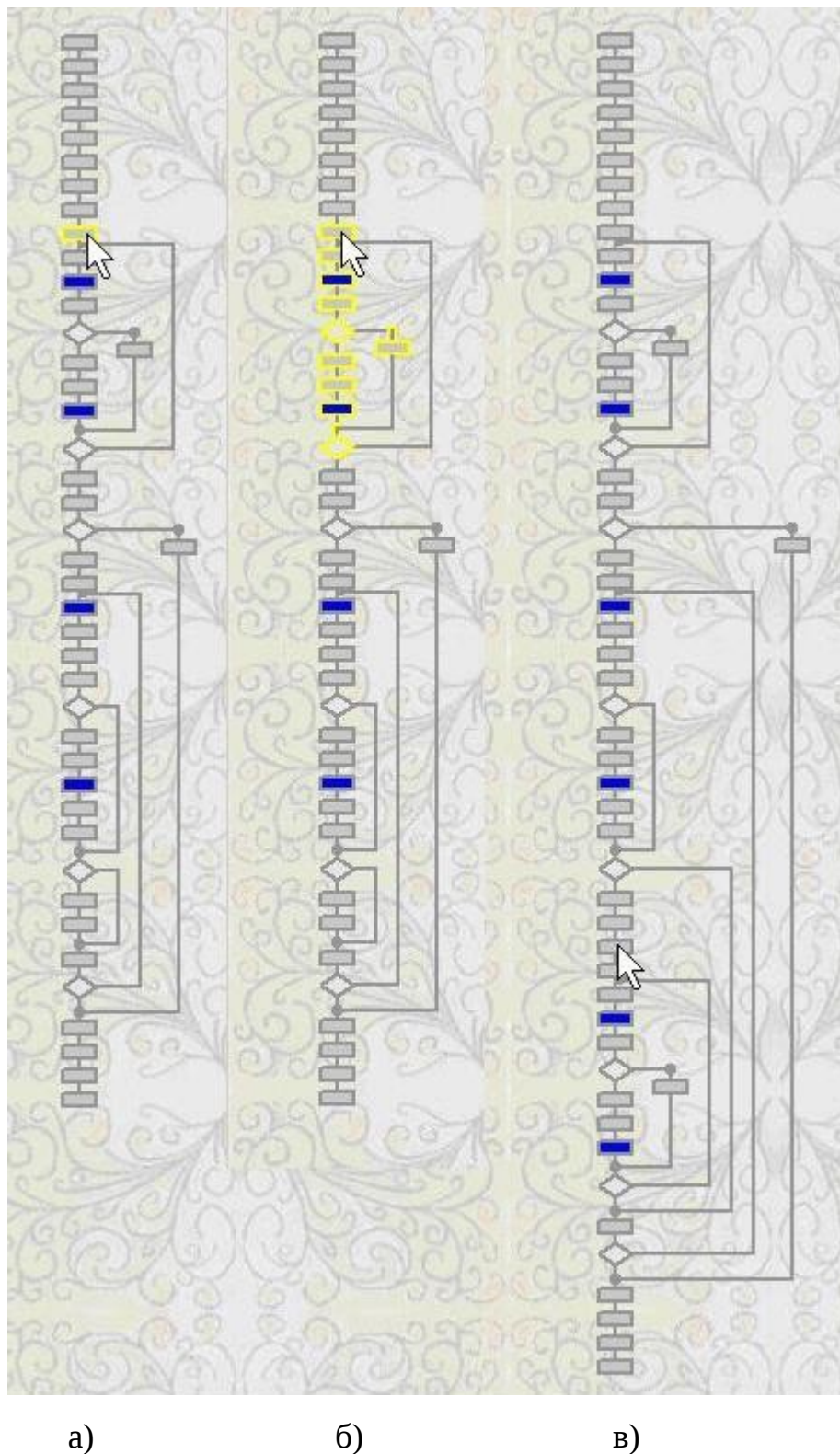


Рис. 9 – Копирование цикла чтения 1000 строк для записи:

- а) нажимаем «Ctrl» и наводим указатель мыши на заголовок цикла;
- б) после отжатия «Ctrl»;
- в) вставка скопированного цикла из буфера копирования.



На рис. 10 приведена вставка цикла для всех записей.

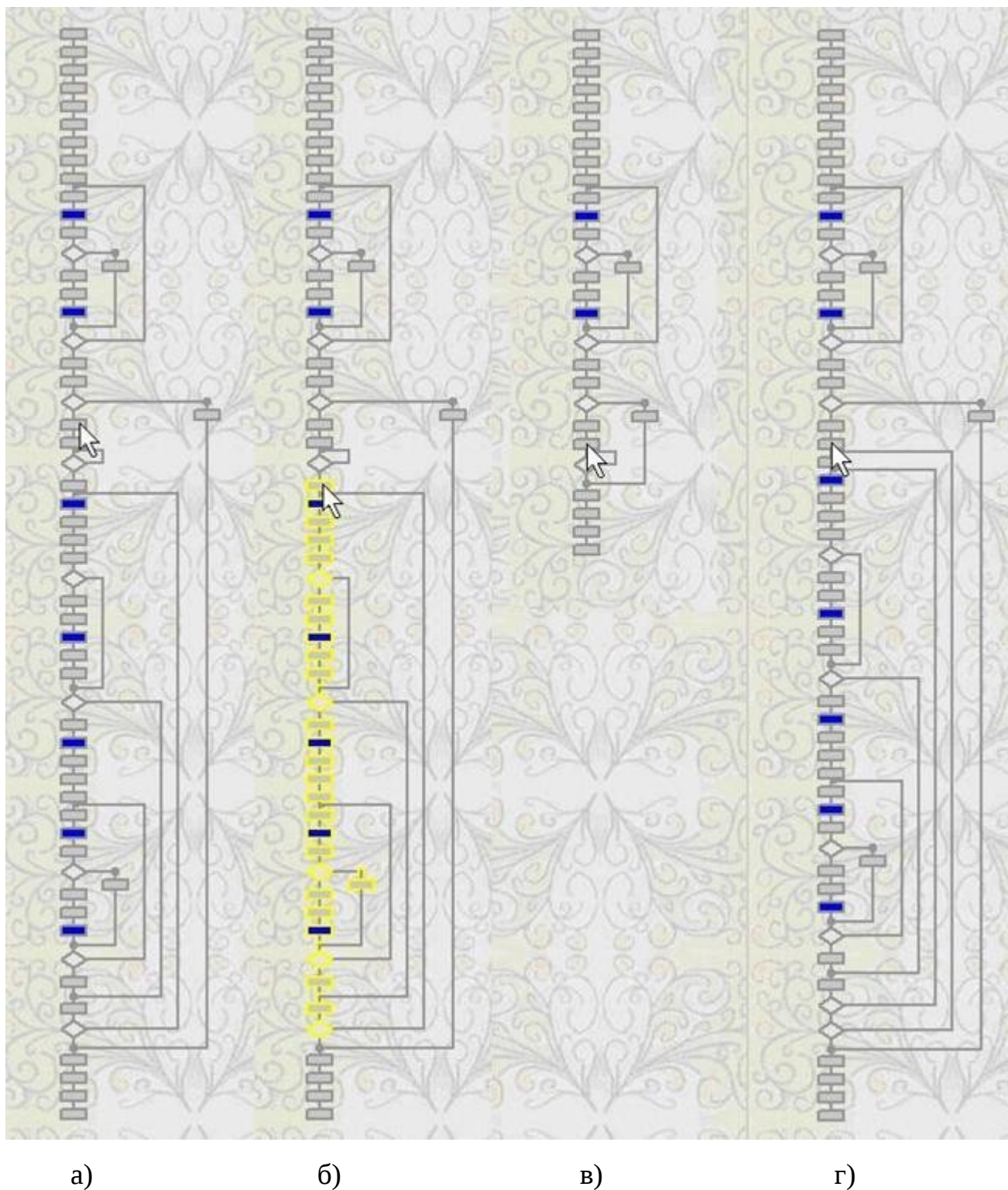


Рис. 10 – Вставка цикла для всех записей:

- а) вставляем цикл;
- б) выделяем цикл для одной записи;
- в) вырезаем цикл для одной записи;
- г) вставляем цикл для одной записи внутрь цикла для всех записей.

На рис. 1 приведена окончательная схема процедуры Sozdatx\_Knigu.

Ниже приведен фрагмент исходного кода процедуры Sozdatx\_Knigu.

```
. . .
(**) Strings.StringToInt(
    m_cfg[C_KOL_STR], kols, res);
(**) IF ((kols < 5) OR (kols > 10)) THEN kols := 5; END;
(**) Strings.StringToInt(
    m_cfg[C_PROKRUTKA], prokrutka, res);
(**) IF m_cfg[C_PUST_STR][0] = 'Y' THEN pustaq_str := TRUE
    ELSE pustaq_str := FALSE; END;  (*--*)
(**) (*[-]*)    (* РИК . .K_8 *)
f := ObxAscii.Open (NIL,
(*      "d:/common/oberon2/My_prg/Mod/chtenie.txt"); *)
    "d:/common/oberon2/My_prg/Mod/bejts1.txt");
f2 := ObxAscii.NewText();
(**) i:=0; kol_str_zap := 0;
(** цикл *) LOOP    (* РИК . .K_9 *)
    (**) ObxAscii.ReadChar(f, c);
    (**) IF ObxAscii.Eot(f) THEN EXIT; END; (* РИК .b? *)
    (**) (*StdLog.Char(c); StdLog.Ln;*)
    IF c = 0DX THEN
        (**) m_str[kol_str_zap][i] := 0X;
        (**) INC(kol_str_zap); i := 0;
        (**) IF kol_str_zap = 1000 THEN EXIT; END; (* РИК .b? *)
    ELSE
        (**) m_str[kol_str_zap][i] := c; INC(i);
    END;
(** конец цикла *) END;
(**) StdLog.Int(kol_str_zap);
. . .
```

Сразу обращают на себя внимание служебные комментарии действия «(\*\*)», определители рисунка и комментария (например, «(\* РИК . .K\_8 \*)», «(\* РИК .b? \*)») и служебные комментарии начала и конца цикла.

Проблема ли это?

По моему мнению, служебные комментарии действия соответствуют пунктам списка в текстовом документе, а служебный комментарий конца цикла поясняет, что это оператор «END;» цикла, а не условия, т.е. оба этих типа служебных комментариев способствуют повышению читабельности текста программы.

Интереснее вопрос с определителями рисунка и комментария. Для чего они?

Определитель рисунка «(\* РИК .b? \*)» показывает наличие в блоке оператора «EXIT» (соответствует оператору «break» в Паскале и C/C++) и этот блок на минимизированной схеме отображается синим цветом, показывая программисту, где у него находятся места выхода из цикла, что представляется достаточно разумным решением (рис. 11).



Рис. 11 – Блок с оператором «EXIT» на минимизированной схеме

Что касается определителя комментария, то во время написания программы подробные комментарии пишутся редко и как правило не по собственному желанию, если внутри компании есть жесткие стандарты написания исходного кода и без подробных комментариев к исходникам работу у программиста просто не примут.

А потом уже как-то психологически напряжно писать комментарии, внося их в уже отлаженный и работающий код, что чревато внесением в него ошибок.

Поэтому введение определителя комментария представляется мне компромиссным решением, которое к тому же при необходимости позволяет редактировать комментарий, не влезая в отлаженный исходный код.

**Кроме того, при вводе исходного текста программы можно вставлять номера комментариев, отсутствующих в файле комментариев проекта. Такая возможность реализована для того, чтобы при написании исходного текста программы не заморачиваться с его комментированием, а уже впоследствии добавить комментарии к исходникам, не трогая при этом уже отлаженный исходный код.**

Для реализации вышеуказанной функции перед началом редактирования исходного кода следует по нажатию клавиши «F6» на клавиатуре вызвать диалог задания текущего номера комментария. А при вводе исходного кода для добавления определителя отсутствующего комментария к первой строке



элемента схемы следует кликнуть по ней мышью, после чего нажать клавишу «F7» на клавиатуре.

В дальнейшем, при вводе этого комментария в файле комментариев, следует перед началом ввода текста комментария скопировать его номер из текста записи над окном файла комментариев, после чего сохранить файл комментариев. В данном случае не требуется устанавливать соответствие между введенным комментарием и соответствующей ему строкой текстовой записи (такое соответствие задается путем двойного клика на первой строке введенного комментария с последующим кликом на строке текстовой записи, к которой требуется присоединить введенный комментарий).

## 5. Гиперссылки

В текущей версии программы dalvjaz2 связи между текстовыми записями (процедурами или их описаниями, определениями типов, классов, переменных и т.п.) формируются при помощи гиперссылок.

Гиперссылки на записи файлов исходного кода имеют вид:

<имя файла с расширением>#<номер записи>, например:

«unit1.pas#n\_30» – гиперссылка на запись n\_30 файла unit1.pas .

Гиперссылки вида #<номер записи> указывают на другие записи файла, в который входит текущая редактируемая запись.

Гиперссылка вида <имя файла с расширением># приведет к открытию в новом экземпляре программы dalvjaz2 списка записей указанного в гиперссылке файла.

Гиперссылка вида <имя файла.jpg># (например: ris\dalvjaz2.jpg#) приведет к открытию в новом экземпляре программы dalvjaz2 jpg-файла, указанного в гиперссылке.

Гиперссылки на записи htm-файлов имеют вид:

<имя htm-файла без расширения>#<номер записи>, например:

prgdoc#8 - ссылка на запись «8» файла prgdoc.htm.

Для вызова htm-файлов из программы dalvjaz2 необходимо в файле dal2.cfg правильно указать путь к браузеру.

Для перехода по гиперссылке из редактируемой текстовой записи необходимо дважды кликнуть по ссылке.

Для перехода по гиперссылке из схемы (рис. 12) достаточно один раз кликнуть по ссылке.

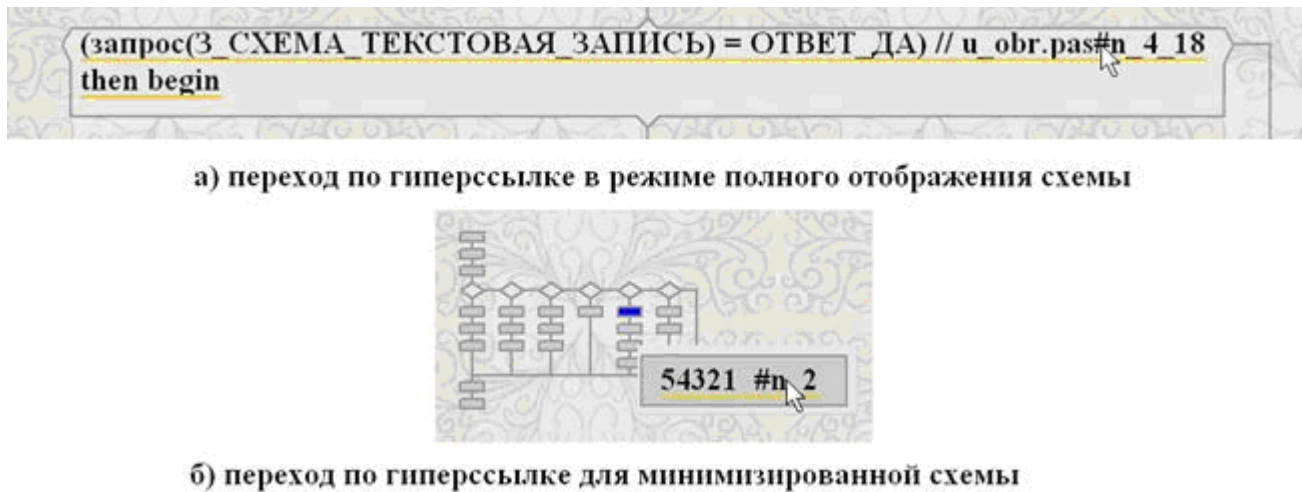


Рис. 12 – переход по гиперссылке из схемы

Кстати, для упомянутой в файлах dalvjaz2\_part1.pdf и dalvjaz2\_part2.pdf процедуры «вывод\_схемы» (в текущей версии программы dalvjaz2 она перенесена на Паскаль и ее исходный код находится в файле u\_lvu.pas) гиперссылки решают вопрос задания быстрых логических связей между процедурой логики верхнего уровня и участками обработчика, связанными с обработкой конкретных команд (запросов) процедуры логики верхнего уровня. Достаточно разбить обработчик на короткие текстовые записи, содержащие обработку для конкретной команды.

Так, при переходе по гиперссылке из фрагмента процедуры логики верхнего уровня, приведенного на рис. 12а в новом экземпляре dalvjaz2 будет открыт фрагмент обработчика (файл u\_obr.pas), выполняющий обработку запроса «3\_СХЕМА\_ТЕКСТОВАЯ\_ЗАПИСЬ».

## 6. Режим «читалка»

Этот пункт написан прежде всего для читателей, которые захотят использовать программу dalvjaz2 как читалку, поэтому здесь я не стану вдаваться в излишние подробности программной реализации режима «читалка», скажу об этом лишь самый минимум в следующем абзаце.

В текущей версии программы dalvjaz2 я решил включить в dalvjaz2 функции читалки «СДЧ», выложенной мной в теме «Видеоэкология чтения 2023», т.к. dalvjaz2 более продвинута, чем «СДЧ», при работе с графикой, а реализованная в dalvjaz2 концепция силуэта (и входящего в него сложного условия) позволяет без особых усилий воспроизвести в dalvjaz2 читалку с горизонтальной прокруткой, да и с вертикальной прокруткой тоже. Надо только переформатировать исходный текстовый файл в соответствующий псевдокод (сложное условие для горизонтальной прокрутки и цепочка действий для вертикальной прокрутки). Этим занимается процедура b\_sozd\_kniguClick (файл u\_ls.pas).

Для использования dalvjaz2 в качестве читалки необходимо выполнить следующие действия:

- скачайте архив dal20931.rar с каталогом исходного кода и файлами конфигурации dal2.cfg и styles.cfg программы dalvjaz2;
- скачайте архив dal20931\_exe.rar с исполняемым файлом программы;
- разверните архивы в каталог dalvjaz2;
- в файле конфигурации dal2.cfg в параметре “\_\_каталог\_исх\_кода” задайте каталог, в котором находятся ваши текстовые файлы;
- запустите программу dalvjaz2, после чего на экране появится главное окно программы dalvjaz2;
- нажмите кнопку “Счит.” в левом верхнем углу окна программы;
- в открывшемся стандартном диалоге открытия файла выберите нужный текстовый файл, после чего на экране появится окно выбора записи открытого текстового файла (оно будет пустым);
- в открывшемся окне нажмите кнопку “Создать книгу”. Кнопка залипнет на несколько секунд, после чего в заданном вами в файле dal2.cfg каталоге текстовых файлов появится файл “kniga\_<имя текстового файла>”;
- нажмите кнопку “Выбрать другой файл” и выберите файл “kniga\_<имя текстового файла>”, после чего на экране появится окно выбора записей этого файла; в каждой записи содержится 1000 строк текста, отформатированного для комфортного чтения;
- выберите одну из записей (например n\_3 2001-3000), после чего запись будет загружена в окно “схема”;

- при необходимости сдвиньте влево границу окон текста записи и схемы и раскройте на полный экран окно программы;

- загруженная запись просматривается в режиме горизонтальной прокрутки с использованием горизонтальной полосы прокрутки и/или колесика мыши; если текущее направление прокрутки по умолчанию – вертикальное (параметр «\_\_прокрутка\_по\_умолчанию» в файле dal2.cfg), то нажмите правую кнопку мыши и перетащите текст примерно на сантиметр по горизонтали, после чего отпустите правую кнопку мыши, а затем используйте колесико мыши для пролистывания текста;

- аналогично загружается книга с вертикальной прокруткой за исключением того, что при необходимости следует настроить колесико мыши для прокрутки по вертикали;

- подробное описание элементов диалога программы приведено в п. 2 «Управляющие элементы диалога».

## **Заключение**

Наконец-то перед Новым Годом у меня появились время и, главное, свежие мысли, чтобы привести в какой-то порядок мои неоконченные наработки по языку ДАЛВЯЗ 2 и реализующей работу с ДАЛВЯЗ 2 программе dalvjaz2.

Возможно, этот материал заинтересует кого-нибудь и подвигнет на усилия в области развития визуального программирования.