ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра ПИ им. Л.П. Фельдмана

Факультет ИСП

Лабораторная работа №4

Курс: «Профессиональная практика программной инженерии»

Тема: «Создание самодокументирующегося кода»

Выполнил

ст. гр. ПИ-19а

Харламов Д.В.

Проверил

Григорьев А.В.

Рычка О.В.

Донецк – 2022

**Цель работы:** научиться добавлять в программный код специальным образом оформление докблок-комментарии, для последующей автоматической генерации API reference, а также познакомиться с форматом  
оформления документации DocBook.

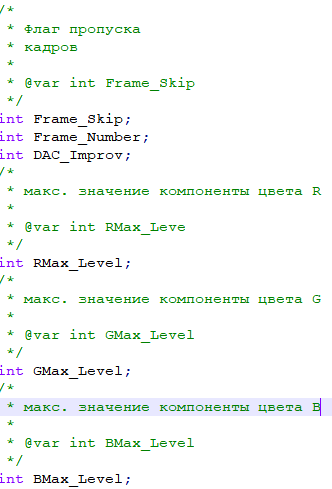


Рисунок 1 – Пример докблока для свойства

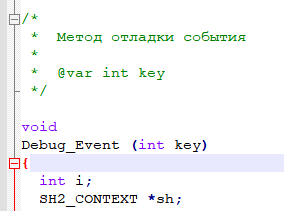


Рисунок 2 – Пример докблока для функции

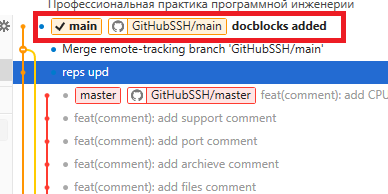


Рисунок 3 – commit изменений с докблоками

Cодержимое файла docbook.xml:

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook XML V4.5//EN"

"http://www.oasis-open.org/docbook/xml/4.5/docbookx.dtd">

<book>

<bookinfo>

<title>Sega BS</title>

<author>

<firstname>Danil</firstname>

<surname>Kharlamov</surname>

<affiliation>

<orgname>jjharlcontact@mail.ru</orgname>

</affiliation>

</author>

<pubdate>07-02-2023</pubdate>

</bookinfo>

<chapter>

<title>Документация к эмулятору Sega BS</title>

<section>

<title>Общее описание</title>

<para><emphasis>Отличный мультиплатформенный эмулятор для

Windows</emphasis></para>

<para>Достоинства - высокая точность эмуляции, отсутствие game-hack'ов и

великолепная совместимость с играми. Для получения звука максимально

близкого к оригиналу, используйте опцию: Sound-&gt;SuperHQ. Главный же

недостаток эмулятора, это продолжение его достоинств - высокие системные

требования (для эмулятора Genesis).</para>

</section>

</chapter>

<chapter>

<title>Работа с эмулятором</title>

<section>

<title>Работа с эмулятором</title>

<para><emphasis>Отличный</emphasis> мультиплатформенный эмулятор для

Windows</para>

<mediaobject>

<imageobject>

<imagedata fileref="image.jpg"/>

</imageobject>

<textobject>

<phrase>Скриншот – вариант использования №1 – игра</phrase>

</textobject>

</mediaobject>

<programlisting>

/\*

\* Change\_File\_S - сохранение

\*

\* @param char \*Dest

\* @param char \*Dir

\*/

int

Change\_File\_S (char \*Dest, char \*Dir)

{

gchar \*filename = NULL;

GtkWidget \*widget;

gint res;

widget =

create\_file\_chooser\_dialog ("Save state", GTK\_FILE\_CHOOSER\_ACTION\_SAVE);

// fileselection\_set\_dir (fd.filesel, Rom\_Dir);

addStateFilter (widget);

res = gtk\_dialog\_run (GTK\_DIALOG (widget));

if (res == GTK\_RESPONSE\_OK)

{

filename = gtk\_file\_chooser\_get\_filename (GTK\_FILE\_CHOOSER (widget));

}

gtk\_widget\_destroy (widget);

if (filename != NULL)

{

strcpy (Dest, filename);

g\_free (filename);

return 1;

}

return 0;

}

/\*

\* Change\_File\_L - загрузка

\*

\* @param char \*Dest

\* @param char \*Dir

\*/

int

Change\_File\_L (char \*Dest, char \*Dir)

{

gchar \*filename = NULL;

GtkWidget \*widget;

gint res;

widget =

create\_file\_chooser\_dialog ("Load state", GTK\_FILE\_CHOOSER\_ACTION\_OPEN);

// fileselection\_set\_dir (fd.filesel, Rom\_Dir);

addStateFilter (widget);

res = gtk\_dialog\_run (GTK\_DIALOG (widget));

if (res == GTK\_RESPONSE\_OK)

{

filename = gtk\_file\_chooser\_get\_filename (GTK\_FILE\_CHOOSER (widget));

}

gtk\_widget\_destroy (widget);

if (filename)

{

strcpy (Dest, filename);

g\_free (filename);

return 1;

}

return 0;

}</programlisting>

<para>Вверху представлена строка меню. Есть возможность

сохранения/загрузки в меню File. Можно задать регион, опции, настроить

управление.</para>

<para>Поддерживаются ROMы формата .bin, .gen; также запакованные в

.zip.</para>

<mediaobject>

<imageobject>

<imagedata fileref="netplay.jpg"/>

</imageobject>

<textobject>

<phrase>Скриншот – вариант использования №2 – игра по сети</phrase>

</textobject>

</mediaobject>

<programlisting>

int

Init\_Network (void)

{

Kaillera\_HDLL = LoadLibrary ("kailleraclient.dll");

if (Kaillera\_HDLL != NULL)

{

Kaillera\_Get\_Version =

(int (WINAPI \*) (char \*version)) GetProcAddress (Kaillera\_HDLL,

"\_kailleraGetVersion@4");

Kaillera\_Init =

(int (WINAPI \*) ()) GetProcAddress (Kaillera\_HDLL, "\_kailleraInit@0");

Kaillera\_Shutdown =

(int (WINAPI \*) ()) GetProcAddress (Kaillera\_HDLL,

"\_kailleraShutdown@0");

Kaillera\_Set\_Infos =

(int (WINAPI \*) (kailleraInfos \* infos))

GetProcAddress (Kaillera\_HDLL, "\_kailleraSetInfos@4");

Kaillera\_Select\_Server\_Dialog =

(int (WINAPI \*) (HWND parent)) GetProcAddress (Kaillera\_HDLL,

"\_kailleraSelectServerDialog@4");

Kaillera\_Modify\_Play\_Values =

(int (WINAPI \*) (void \*values, int size))

GetProcAddress (Kaillera\_HDLL, "\_kailleraModifyPlayValues@8");

Kaillera\_Chat\_Send =

(int (WINAPI \*) (char \*text)) GetProcAddress (Kaillera\_HDLL,

"\_kailleraChatSend@4");

Kaillera\_End\_Game =

(int (WINAPI \*) ()) GetProcAddress (Kaillera\_HDLL,

"\_kailleraEndGame@0");

if ((Kaillera\_Get\_Version != NULL) and (Kaillera\_Init != NULL)

and (Kaillera\_Shutdown != NULL) and (Kaillera\_Set\_Infos != NULL)

and (Kaillera\_Select\_Server\_Dialog != NULL)

and (Kaillera\_Modify\_Play\_Values != NULL)

and (Kaillera\_Chat\_Send != NULL) and (Kaillera\_End\_Game != NULL))

{

Kaillera\_Init ();

Kaillera\_Initialised = 1;

return 0;

}

FreeLibrary (Kaillera\_HDLL);

}

Kaillera\_Get\_Version = Empty\_Kaillera\_Get\_Version;

Kaillera\_Init = Empty\_Kaillera\_Init;

Kaillera\_Shutdown = Empty\_Kaillera\_Shutdown;

Kaillera\_Set\_Infos = Empty\_Kaillera\_Set\_Infos;

Kaillera\_Select\_Server\_Dialog = Empty\_Kaillera\_Select\_Server\_Dialog;

Kaillera\_Modify\_Play\_Values = Empty\_Kaillera\_Modify\_Play\_Values;

Kaillera\_Chat\_Send = Empty\_Kaillera\_Chat\_Send;

Kaillera\_End\_Game = Empty\_Kaillera\_End\_Game;

Kaillera\_Initialised = 0;

return 1;

}</programlisting>

<para>Для игры по сети используется сервер Kaillera.</para>

</section>

<section>

<title>Работа с Sega CD</title>

<mediaobject>

<imageobject>

<imagedata fileref="image.jpg"/>

</imageobject>

</mediaobject>

<mediaobject>

<imageobject>

<imagedata fileref="cd.png"/>

</imageobject>

<textobject>

<phrase>Скриншот – вариант использования №3 – Sega CD</phrase>

</textobject>

</mediaobject>

<programlisting>

void

Check\_CD\_Command (void)

{

#ifdef DEBUG\_CD

fprintf (debug\_SCD\_file, "CHECK CD COMMAND\n");

#endif

// Check CDD

if (CDD\_Complete)

{

CDD\_Complete = 0;

#ifdef DEBUG\_CD

fprintf (debug\_SCD\_file, "CDD exporting status\n");

fprintf (debug\_SCD\_file,

"Status=%.4X, Minute=%.4X, Seconde=%.4X, Frame=%.4X, Ext=%.4X\n",

CDD.Status, CDD.Minute, CDD.Seconde, CDD.Frame, CDD.Ext);

#endif

CDD\_Export\_Status ();

if (Int\_Mask\_S68K and 0x10)

sub68k\_interrupt (4, -1);

}

// Check CDC

if (SCD.Status\_CDC and 1) // CDC is reading data ...

{

#ifdef DEBUG\_CD

fprintf (debug\_SCD\_file, "Send a read command\n");

#endif

// DATA ?

if (SCD.TOC.Tracks[SCD.Cur\_Track - SCD.TOC.First\_Track].Type)

CDD.Control |= 0x0100;

else

CDD.Control and= ~0x0100; // AUDIO

if (File\_Add\_Delay == 0)

{

if (CD\_Load\_System == CDROM\_)

ASPI\_Read\_One\_LBA\_CDC ();

else

FILE\_Read\_One\_LBA\_CDC ();

}

else

File\_Add\_Delay--;

}

if (SCD.Status\_CDD == FAST\_FOW)

{

SCD.Cur\_LBA += 10;

CDC\_Update\_Header ();

}</programlisting>

<para>Запуск CD-дисков.</para>

</section>

</chapter>

</book>

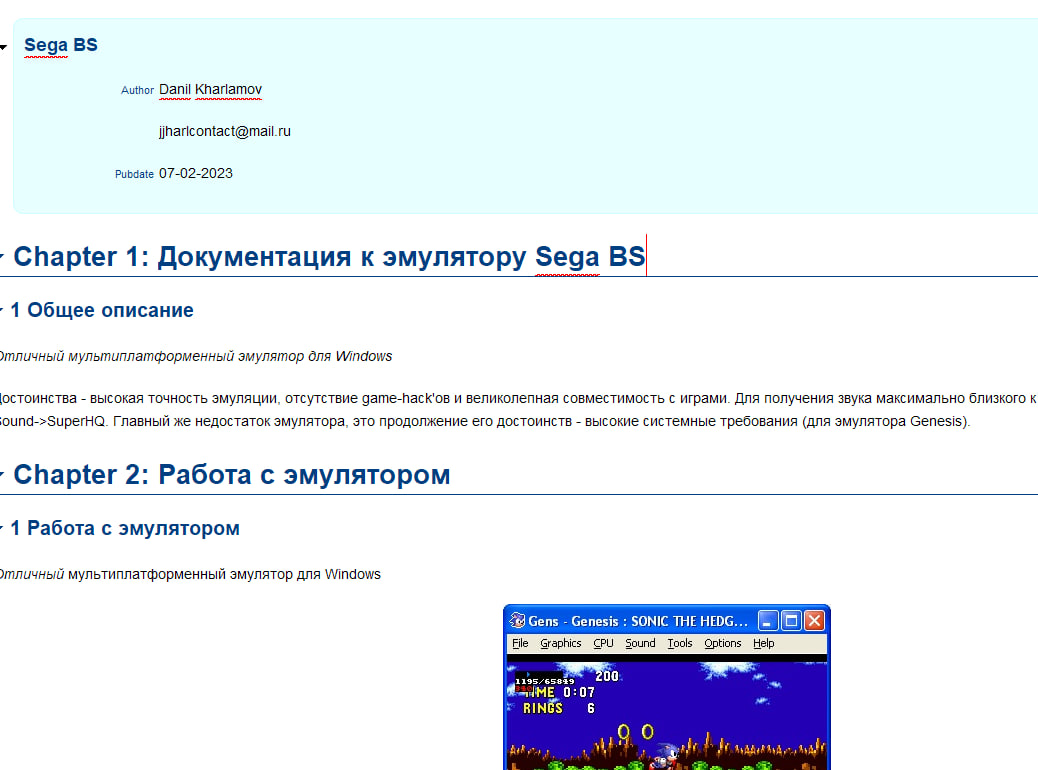


Рисунок 4 – Фрагмент DocBook



Рисунок 5 – Фрагмент DocBook

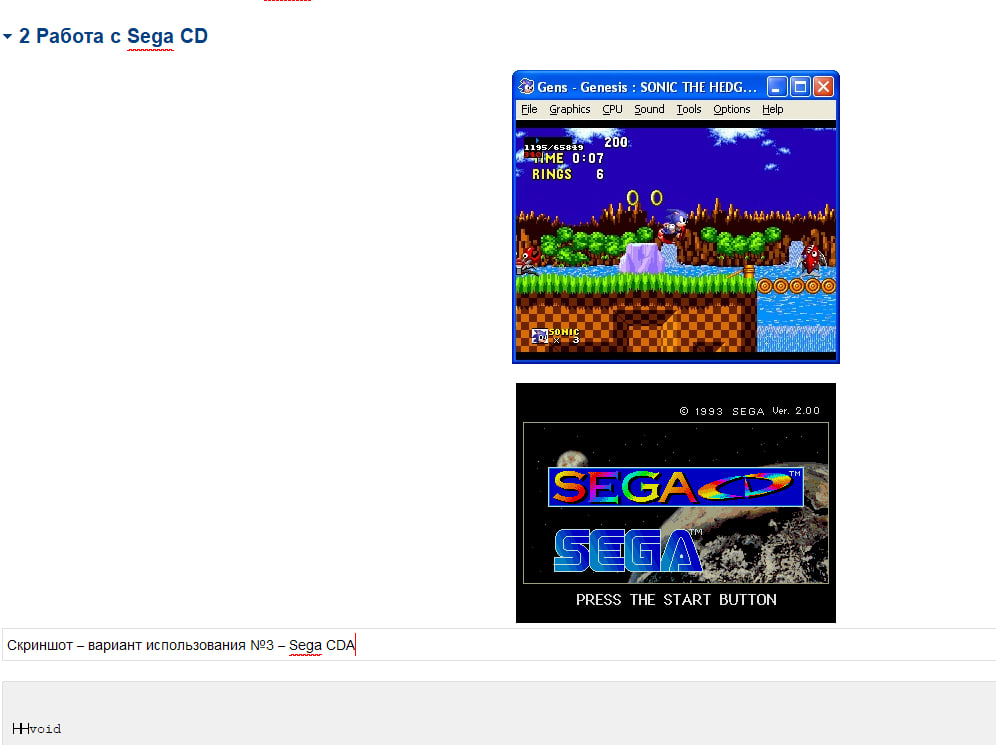


Рисунок 6 – Фрагмент DocBook

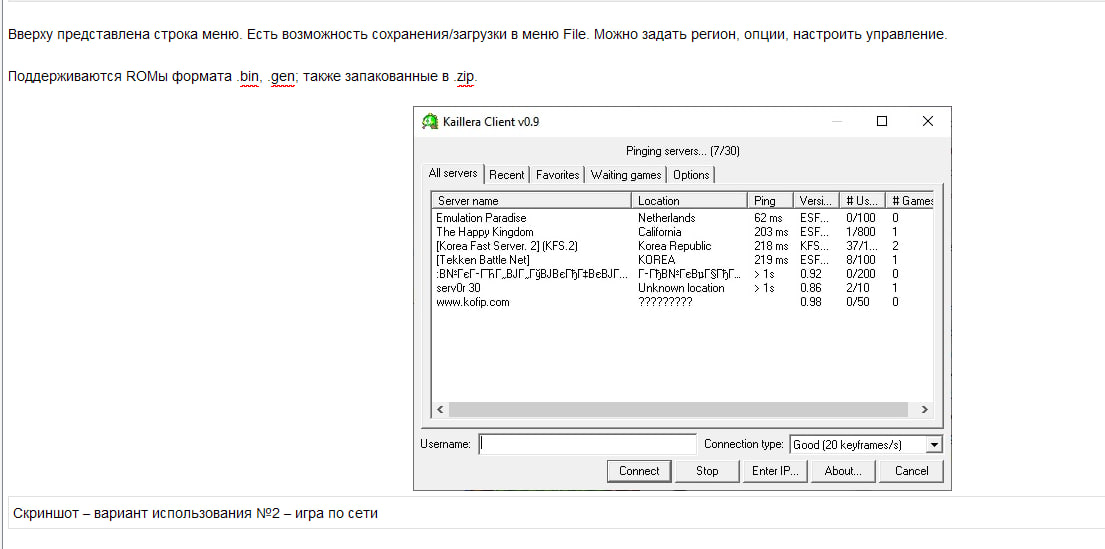


Рисунок 7 – Фрагмент DocBook

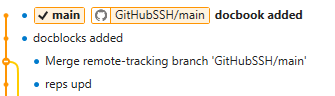


Рисунок 8 – коммит docbook.xml

Т.к. окно ошибок редактора пустое, то валидация считается пройденной успешно (рисунок 9).

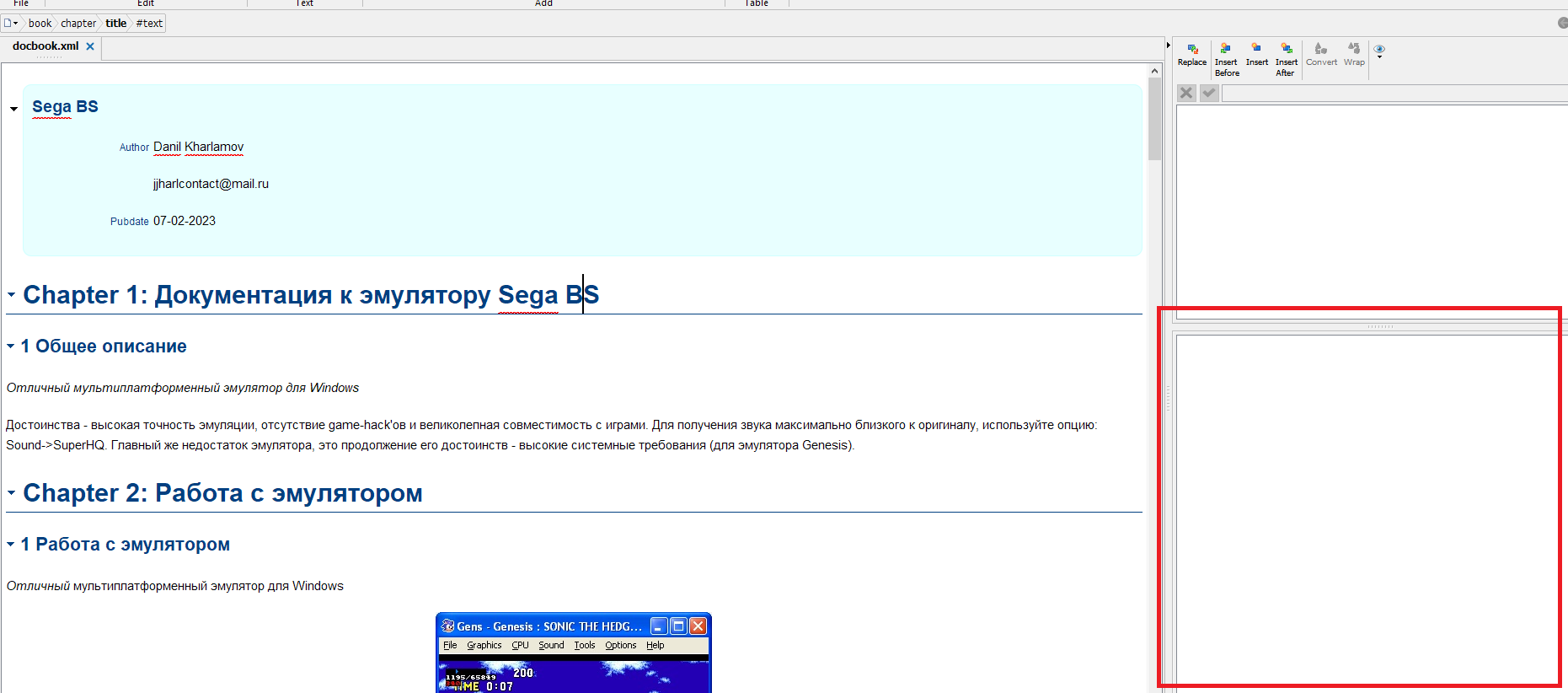


Рисунок 9 – Отсутствие ошибок