

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/346447041>

Doxygen и Graphviz : ДОКУМЕНТИРОВАНИЕ ПРОЕКТОВ НА C#

Article · May 2006

CITATIONS

0

READS

16

1 author:



Alexei G. Piskunov

National Aviation University

35 PUBLICATIONS 9 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Aerial Photo [View project](#)



Разработка десктопных приложений. Гексалогия. [View project](#)

Doxygen и Graphviz : ДОКУМЕНТИРОВАНИЕ ПРОЕКТОВ НА C#

ПИСКУНОВ А.Г., ГОРБАНЬ А.С.

26 июня 2015 г.

АННОТАЦИЯ

Данный документ обсуждает использование open source [4] утилит Doxygen [3] и dot из пакета Graphviz [1] для генерации документации из кодов C.

Дата создания документа: 26.06.2015

1 БЫСТРОЕ ВВЕДЕНИЕ В DOXYGEN

Doxygen [3] - это утилита создания документации для программ, написанных на C++, C, Java, Objective-C, Python, IDL (Corba и Microsoft) и некоторых версий PHP, C, и D. Выходная документация представляет собой файлы в html-формате, в формате LaTeX (в частности, для дальнейшего создания pdf - файлов), rtf, справочного руководства в стиле Unix - man, справочного руководства в стиле кроссплатформенной библиотеки QT, xml и так далее.

Даже, если не оформлять специальным образом исходные тексты, Doxygen сгенерирует из них замечательные справочники программного интерфейса приложения. В выходной документ могут быть включены разделы с описанием используемых namespaces-ов, классов, файлов и их указатели на соответствующие разделы. В случае использования пакета визуализации графов - Graphviz [1], описание классов может сопровождаться симпатичными диаграммами. Пример диаграммы достаточно сложного класса из реального проекта представлен на рис. 1.

Кириллизация Doxygen -а выполнена Александром Челпановым (русский) и Алексем Ткаченко (Olexij Tkatchenko, украинский).

1.1 Doxygen license

Permission to use, copy, modify, and distribute this software and its documentation under the terms of the GNU General Public License is hereby granted. No representations are made about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty. See the GNU General Public License for more details.

Documents produced by Doxygen are derivative works derived from the input used in their production; they are not affected by this license.

Действие этой лицензии не распространяется на документы, полученные при помощи Doxygen -а . Так как они считаются производным от ввода, использованного при создании документов.

1.2 A BINARY DISTRIBUTION Doxygen 1.4.6 FOR WINDOWS

Дистрибутив Doxygen версии 1.4.6 можно скачать по адресу:

<ftp://ftp.stack.nl/pub/users/dimitri/doxygen-1.4.6-setup.exe> - 5466Кбайт
либо

<ftp://ftp.stack.nl/pub/users/dimitri/doxygen-1.4.6.windows.bin.zip>
- 1171 Кбайт.

В комплект поставки входят две утилиты, которые должны быть доступны через переменную среды PATH:

```
-rwxrwxrwx  1 user      group      2781184 May  8  2004 doxygen.exe
-rwxrwxrwx  1 user      group      172032 May  8  2004 doxytag.exe
```

На этом инсталляция закончивается.

1.3 GENERATION OF HTML DOCS

1.3.1 Get started

Создадим справочник программиста в формате html по проекту Hello, представленному в разделе 2 . Для этого необходимо выполнить следующие шаги

- создать файл конфигурации Doxygen -а

```
doxygen -g hello
```

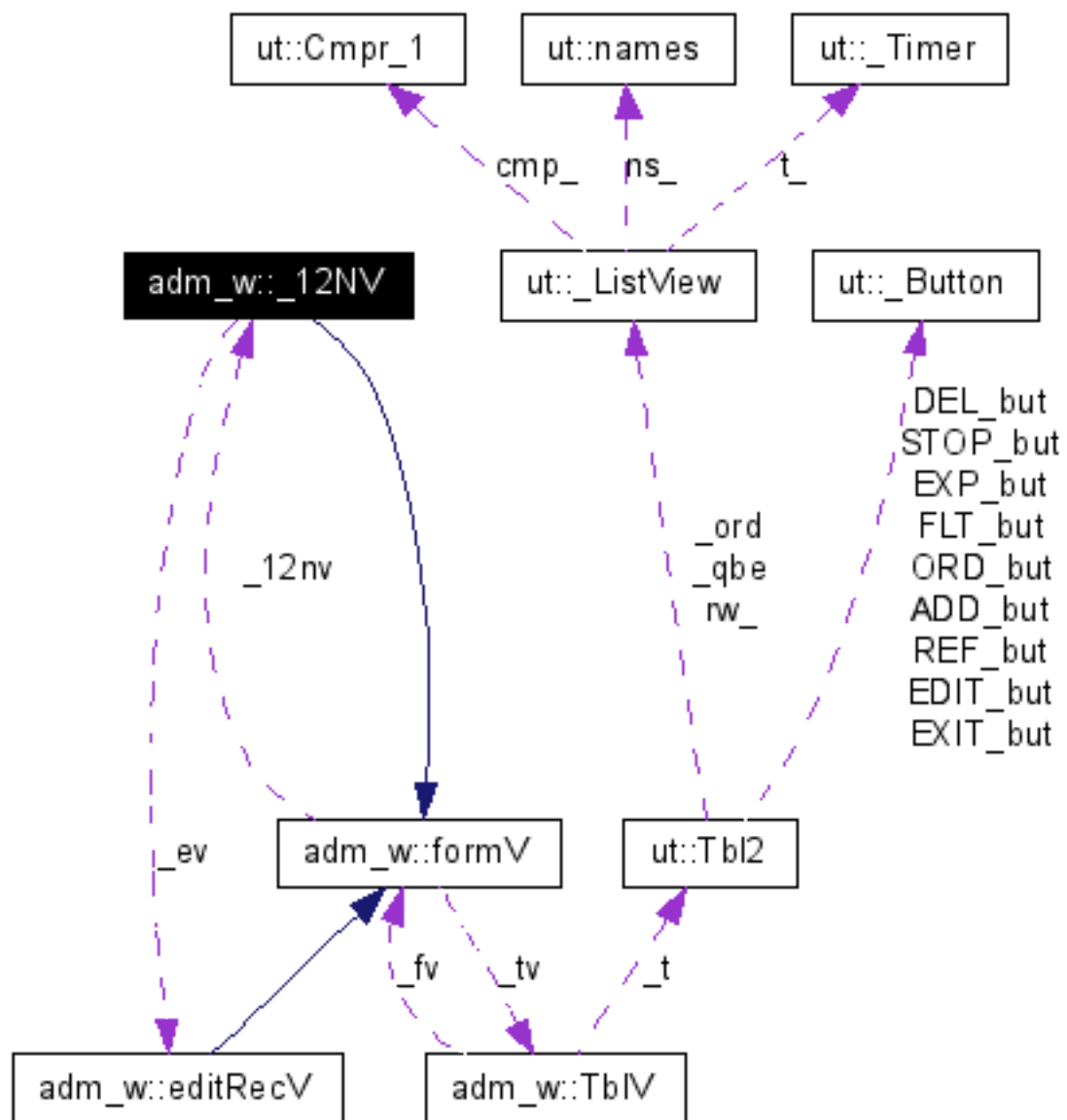


Рис. 1: Пример диаграммы ссылок на классы

- получив файл конфигурации ./hello., являющийся текстовым файлом, задайте в нем следующие параметры:
 - PROJECT_NAME = hello
 - OUTPUT_DIRECTORY = _bld
 - OUTPUT_LANGUAGE = Russian
 - EXTRACT_ALL = YES
по умолчанию, в документацию попадают только специально откомментированные части кода. Здесь задано включение в документ всех классов, файлов и т.д., так как в нашем проекте еще нет таких, специально откомментированных частей кода.
 - INPUT = ./csharp/
 - FILE_PATTERNS = *.cs
 - EXTRACT_STATIC = YES
Этот параметр используется, чтобы включить в документ статические члены файла;
- Постройте документацию командой

```
doxygen hello
```

Примечание

Подразумевается, что должна была получиться следующая структура рабочего каталога

```
---- File:lslr.
```

```
---- End Of File:lslr.
```

- откройте браузером файл ./_bld/html/index.html

Вполне вероятно, что Ваш браузер покажет вполне читабельную страницу, похожую на страницу, представленную на рис. [2](#)

Кликнув на гиперссылку Классы, а затем, adm_w::b увидим страницу [3](#)

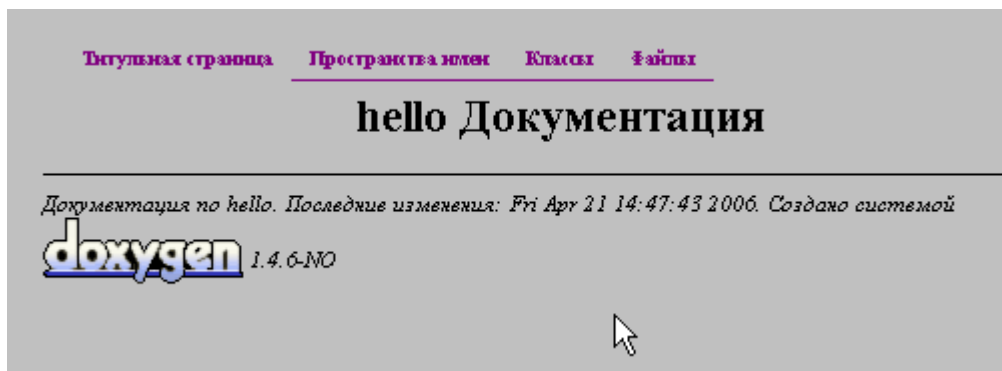


Рис. 2: стартовая страница html-документа по проекту Hello



Рис. 3: страница html-документа с описанием класса adm_w::b

1.3.2 Improvement of the graphs of document

Можно задать еще некоторые параметры:

- **HAVE_DOT = YES**
Означает, что на компьютере установлен пакет визуализации графов Graphviz [1], и Doxygen будет пользоваться этим пакетом для генерации диаграмм. Много опций Doxygen имеет смысл только тогда, когда используется Graphviz. Установка Graphviz обсуждается в разделе 3;
- **CLASS_DIAGRAMS = NO**
Убирает из описания каждого класса диаграмму наследования класса. На этой диаграмме изображены все предки и потомки класса;
- **HIDE_UNDOC_RELATIONS = YES**
Убирает из документа связи между документированными классами. В нашем примере классы a, b, c, и т.д. документированные, а класс int - не документирован;
- **CLASS_GRAPH = YES**
В описании класса появляется диаграмма наследования класса (вида отличного от вида CLASS_DIAGRAMS) и список классов потомков (наследуемые классы) и классов предков (базовые классы). Параметр CLASS_DIAGRAMS ставится в значение NO;
- **COLLABORATION_GRAPH = YES**
Появляется граф связей. На диаграмме отображаются предки, дети и члены - объекты класса. Если выключен параметр HIDE_UNDOC_RELATIONS = NO на диаграммах нашего проекта будут отображены все члены класса включая поля dummy и dummy2;
- **UML_LOOK = YES**
любители UML могут использовать UML стиль диаграмм;
- **GRAPHICAL_HIERARCHY = YES**
Появляется граф с общей иерархией всех классов приложения. См. рис 4;
- **GENERATE_TREEVIEW = YES**
Приводит к появлению окна с деревом всех объектов проекта как на рис. 5;

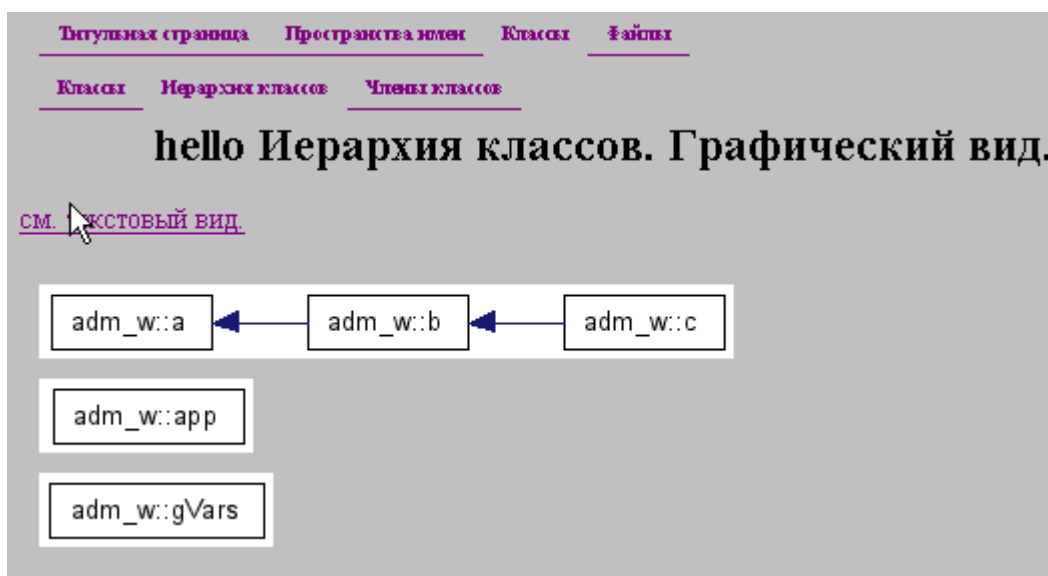


Рис. 4: Построенная иерархия классов

- Не удалось увидеть изменения при использовании параметров `CALL_GRAPH` - граф вызовов функций, `INCLUDE_GRAPH` и `INCLUDED_BY_GRAPH` - графы включения файлов.

1.3.3 Improvement of the text of document

1.3.3.1 Comments of code. Специальные комментарии, расположенные в коде программы попадают в документацию. Комментарии бывают короткие или подробные. Допускается не более одного короткого и одного детального комментария перед каждым документируемым элементом (файл, класс, переменная, функция, перечисление и т.д.) проекта. Объект файл представляет собой исключение и будет обсужден позже. Есть много разных возможностей для коротких и подробных комментариев, остановимся на двух

- дополнительный символ слэш - `///` создает короткий комментарий;
- дополнительный символ звездочка - `/**` создает подробный комментарий.

Следующий код

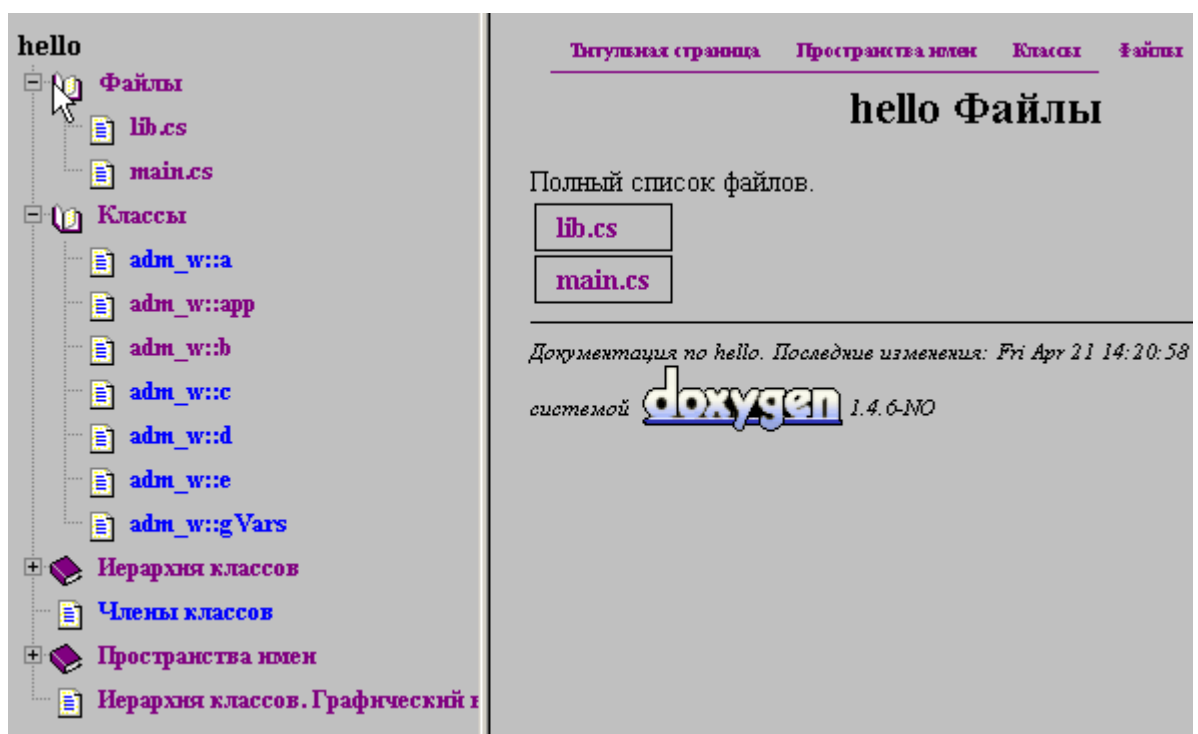


Рис. 5: Дерево объектов проекта

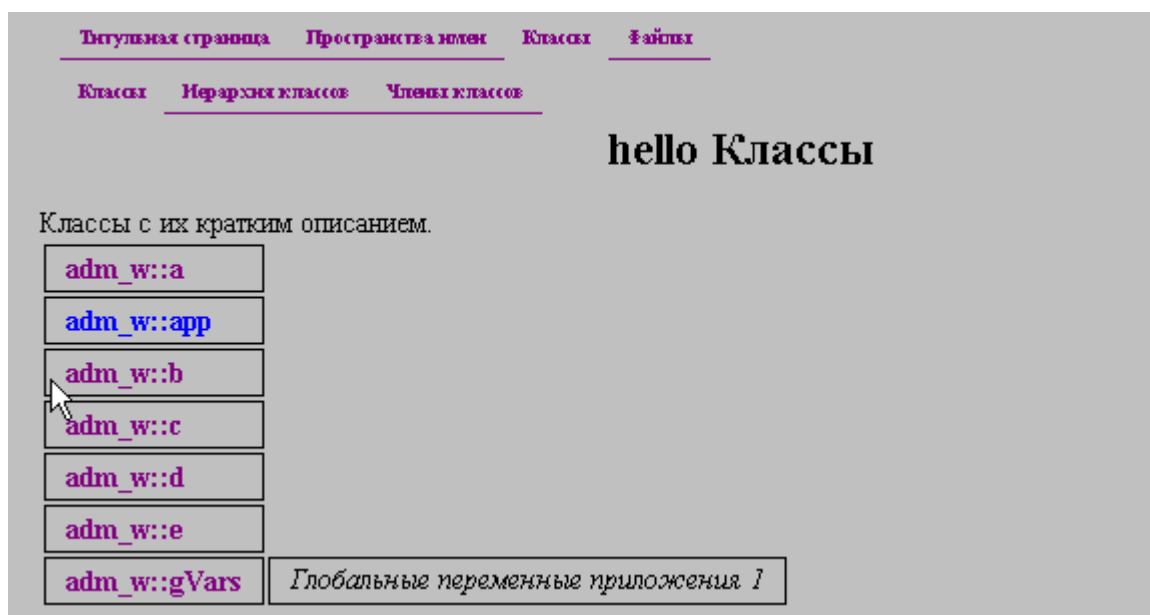


Рис. 6: Пример короткого комментария

```

/// Глобальные переменные приложения 1
public class
    gVars
        /**
            Эти переменные могут использоваться в каждом классе или каждой функции
            приложения.
        */

```

приводит к изменениям в документе, показанных на рис. 6 и 7. Напоминаем, что комментарий должен быть перед комментируемым элементом проекта, по крайней мере, в данном случае, перед открывающей скобкой класса. Символ `<` может изменить это правило.

```

public const string    path = "./var";        ///< каталог временных файлов.
public const string    nm   = "adm_w.log";

```

Символ больше `<` говорит Doxygen, что комментарий относится не к `nm`, а к `path`.

1.3.3.2 Special commands. Существует набор специальных команд для улучшения читабельности текста и позволяющий держать комментарии в дополнительных файлах.

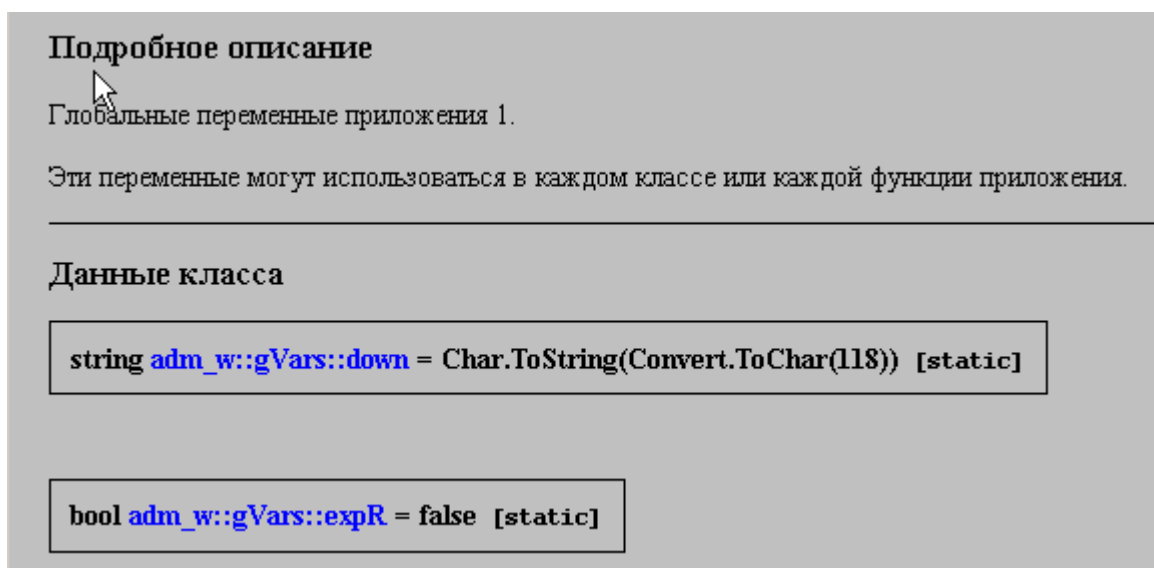


Рис. 7: Пример подробного комментария

В частности, команда `\file` используется для комментирования файла, как показано в файле `lib.cs`:

```
/** \file
    \brief файл с библиотекой
 */
```

Мы не будем останавливаться на всех командах, а приведем пример с

- использованием списков (элемент списка отмечается символом -);
- комментированием файла (команда `\file`) внутри и вне файла;
- команды автор (`\author`);
- команды дата (`\date`);
- команды замечание (`\note`);
- команды короткий комментарий (`\brief`);
- команды новый параграф (`\par`);

В каталог `./csharp` поместим файл `main.txt` со следующим содержанием:

---- File:./csharp/main.txt

```
/**
 \file main.cs
 \brief главный файл приложения

 очень подробное описание файла main.cs
 \author Вася Пупкин
 \date 2006
 \note
  это приложение было разработано для рассылки hDrummer-a,
  члена sql.ru с 2002 года.

 \par
  Цель проекта заключалась в
  - демонстрации возможностей Doxygen-a;
  - введению в Doxygen.
*/
```

---- End Of File:./csharp/main.txt

Изменим параметр файла конфигурации Doxygen

```
FILE\_PATTERNS          = main.txt *.cs
```

Тогда в выходном документе появятся изменения, показанные на рис.
[8](#) .

2 CODE OF HELLO PROJECT

Для генерации документов использовался следующий пример - проект на C-е, состоящий из двух файлов

- main.cs;
- lib.cs;

Файл main.cs

главный файл приложения [Подробнее...](#)

Пространства имен

namespace **adm_w**

Классы

class	adm_w::gVars	<i>Глобальные переменные приложения 1. Подробнее...</i>
class	adm_w::app	

Подробное описание

главный файл приложения

очень подробное описание файла **main.cs**

Автор:
Вася Пупкин

Дата:
2006

Заметки:
это приложение было разработано для рассылки hDrummer-a, члена sql.ru с 2002 года.

Цель проекта заключалась в

- ♦ демонстрации возможностей Doxygen;
- ♦ введению в изучение Doxygen.

Рис. 8: Пример документирования файла main.cs

2.1 main.cs file

---- File:./csharp/main.cs

```
using System;
using System.IO;
namespace adm_w{
public class gVars
    /// Глобальные переменные приложения 1
    /**
        Эти переменные могут использоваться в каждом классе или каждой функции
        приложения.
    */
{
    public const string version = "0.92.1";
    public const string      name = "DBManager";
    public const string      path = "./var";      ///< каталог временных файлов.
    public const string      nm   = "adm_w.log";
    public static string uuser   = "";
    public static string upwd    = "";
}
class app {
    static bool isArgs(string s){
        bool r = false;
        switch (s[0]) {
            case '-': r = true; break;
            case '/': r = true; break;
        }
        return r;
    }
    static bool isArgs(string arg, params string[] vals) {
        bool r = false;
        for (int i = 0; i < vals.Length; i++){
            if (arg.Substring(1).ToLower() == vals[i]) {
                r = true;
                break;
            }
        }
        return r;
    }
    static void Main(string[] args) {
        for (int i = 0; i < args.Length; i++){
            if (isArgs(args[i])) {
```

```

        if (isArgs (args[i], "?","h","help"))
            goto error;
        else if (isArgs (args[i], "u")) {
            i++;
            if (i >= args.Length ) goto error;
            gVars.uuser= args[i];
        }
        else if (isArgs (args[i], "p")) {
            i++;
            if (i >= args.Length ) goto error;
            gVars.upwd= args[i];
        }
    }
}
c.wrLn ("Hello, world2!"); b.wrLn ("Hello, world!");
return;
error:
    string msg = "where:\n"+ "\t -u unisay user\n" + "\t -p unisay pwd\n";
    a.wrLn ("usage "+ gVars.name +": mnr [-u xx -p yy ]" ); b.wrLn (msg );
}
}}
```

---- End Of File:./csharp/main.cs

2.2 lib.cs file

---- File:./csharp/lib.cs

```

/** \file
    \brief файл с библиотекой
 */
using System;
using System.IO;
namespace adm_w{
    class a{
        static public void wrLn ( string s ) {          /// \callgraph
            Console.WriteLine (s);
        }
    }
    class b: a { int dummy; }
    class c: b { int dummy2; }
    class d { int dummy; }
    class e {
```

```
public int dummy;  
public d    foo;  
public void x (string s) { a.wrLn(s); }  
}  
}
```

---- End Of File:./csharp/lib.cs

3 A BINARY DISTRIBUTION GRAPHVIZ FOR WINDOWS

Дистрибутив версии 2.8 можно скачать по адресу:

<http://www.graphviz.org/pub/graphviz/ARCHIVE/graphviz-2.8.exe> - 5551Кбайт

Для инсталляции необходимо всего лишь ответить на вопрос куда инсталлировать пакет. В дистрибутив входит

- значительное количество утилит. В Doxygen из них используется утилита dot.exe. Это утилита, которая из текстового описания графа, сделанного на достаточно понятном языке описания графов - DOT [2] , строит файлы популярных форматов, в частности png, jpg, ps, ...;
- Также, самостоятельный интерес представляет утилита для визуального редактирования графов - dotty.exe;
- Утилита для редактирования картинок - lefty.exe;
- Набора руководств пользователя по всем утилитам;
- Большого количества разнообразных примеров графов, сделанных на языке DOT;
- API интерфейс для использования возможностей Graphviz в приложениях, написанных на C;
- Описание API интерфейса находится по адресу <http://www.graphviz.org/pub/graphviz/CURRENT/doxygen/html/>.

Graphviz можно использовать по обычной лицензии GPL. Заметить поддержку кириллицы не удалось.

ССЫЛКИ

- [1] AT&T and Lucent Bell Labs. Graphviz - Graph Visualization Software. <http://www.graphviz.org/>.
- [2] AT&T and Lucent Bell Labs. The DOT Language. <http://www.graphviz.org/doc/info/lang.html>.
- [3] Dimitri van Heesch. Doxygen - Documentation System. <http://www.stack.nl/~dimitri/doxygen/index.html>.
- [4] OpenSourceCommunity. SourceForge. <http://sourceforge.net/>.