

Введение

Документ содержит краткие рекомендации по использованию пакета *Doxygen* совместно с *Microsoft Visual Studio* и *Eclipse IDE for C/C++ Developers* в ОС *Microsoft Windows* для документирования разрабатываемых проектов на языке *C++*.

Doxygen – система документирования исходных текстов, поддерживающая языки *C++*, *C*, *Java*, *Objective-C*, *Python*, *IDL (Corba and Microsoft flavors)*, *Fortran*, *VHDL*, *PHP*, *C#*, частично *D*.

Вики:

<http://ru.wikipedia.org/wiki/Doxygen>

Сайт проекта:

<http://www.stack.nl/~dimitri/doxygen/>

Цель

Ознакомление со средствами автоматического документирования проектов на языке *C++* в среде *Microsoft Visual Studio* с помощью пакета *Doxygen*.

Требования

1) Пакет *Microsoft Visual Studio* – предварительно установлен и настроен для разработки проектов на языке *C++*.

2) Пакет *Eclipse IDE for C/C++ Developers* – предварительно установлен и настроен для разработки проектов на языке *C++*.

3) Пакет *Doxygen* – установка, настройка и использование рассматриваются ниже. Для полноценной работы пакета необходимо установить дополнительные программные средства.

Graphviz – пакет утилит для автоматической визуализации графов, заданных в виде описания на языке *dot*.

Вики:

<http://ru.wikipedia.org/wiki/Graphviz>

Сайт проекта:

<http://www.graphviz.org/>

Mscgen – программное средство для построения диаграмм последовательности (в *UML* – диаграммы взаимодействия объектов, упорядоченные по времени их проявления).

Вики:

http://ru.wikipedia.org/wiki/Диаграмма_последовательности

<http://ru.wikipedia.org/wiki/UML>

<http://en.wikipedia.org/wiki/MscGen>

Сайт проекта:

<http://www.mcternan.me.uk/mscgen/>

Дистрибутивы (стабильные версии на 7.10.2015):

1) *doxygen-1.8.10-setup.exe*

<http://ftp.stack.nl/pub/users/dimitri/doxygen-1.8.10-setup.exe>

2) *graphviz-2.38.msi*

<http://www.graphviz.org/pub/graphviz/stable/windows/graphviz-2.38.msi>

3) *mscgen_0.20.exe*

http://www.mcternan.me.uk/mscgen/software/mscgen_0.20.exe

Подготовка

Установка ПО

Установить перечисленные дистрибутивы в разделе "Требования" в указанном порядке. Рекомендуемые пути для установки пакетов *Doxygen*, *Graphviz* и *Mscgen* соответственно:

c:\Program Files\doxygen

c:\Program Files\doxygen\graphviz

c:\Program Files\doxygen\mscgen

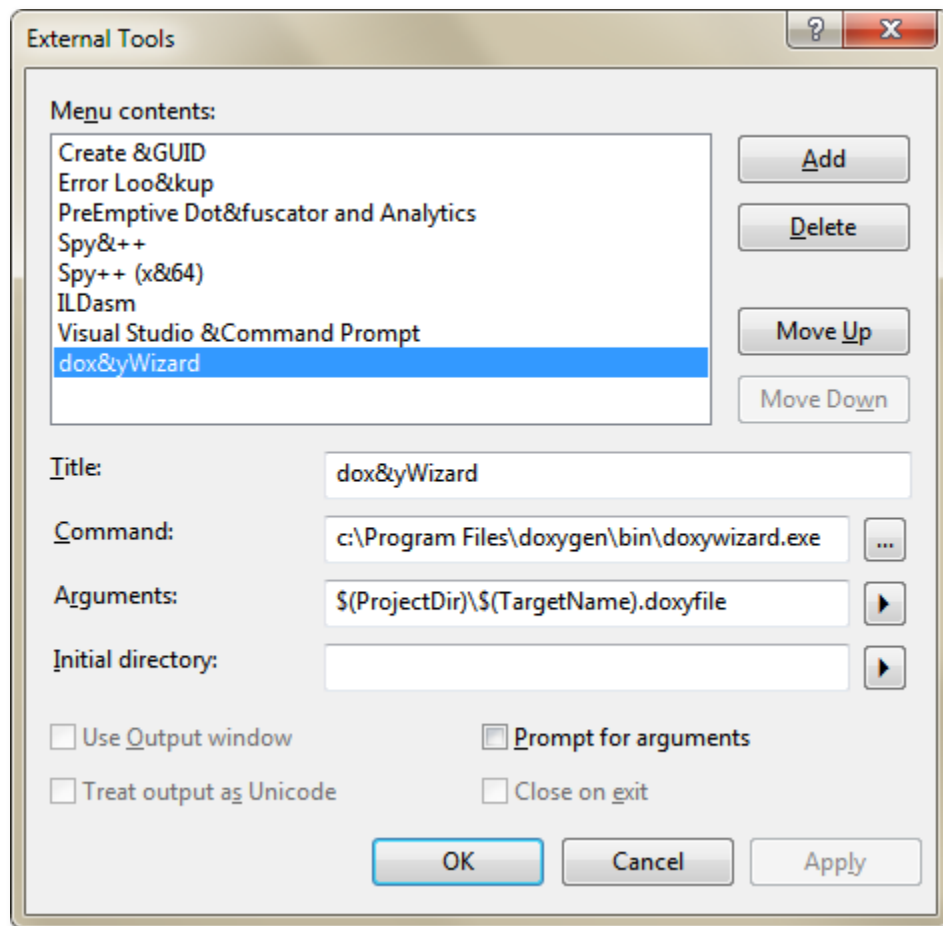
Настройка Visual Studio для использования Doxygen

- 1) Создать простейший проект *Visual C++*.
- 2) В основном меню выбрать пункт *Tools/External Tools*.
- 3) С помощью кнопки *Add* добавить пункт *doxyWizard* со следующими параметрами:

Title: dox&yWizard

Command: C:\Program Files\doxygen\bin\doxywizard.exe

Arguments: \$(ProjectDir)\\$(TargetName).doxyfile

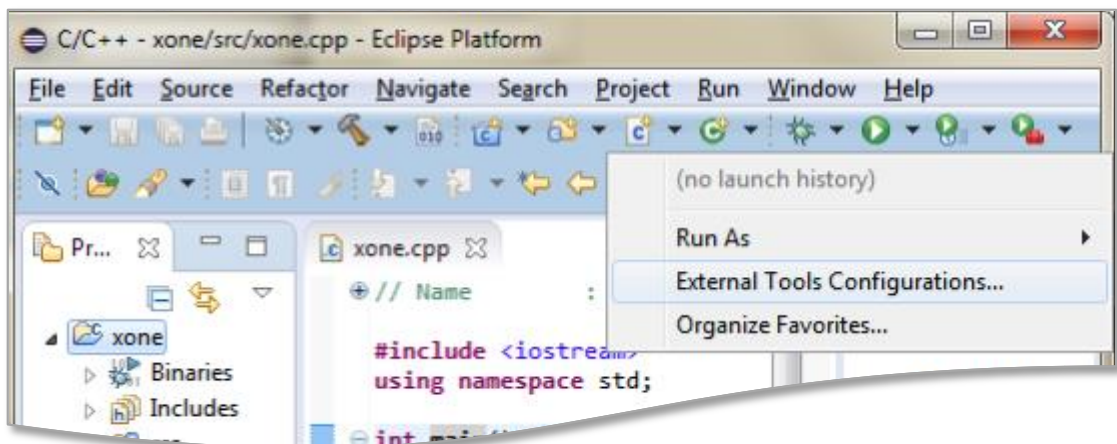
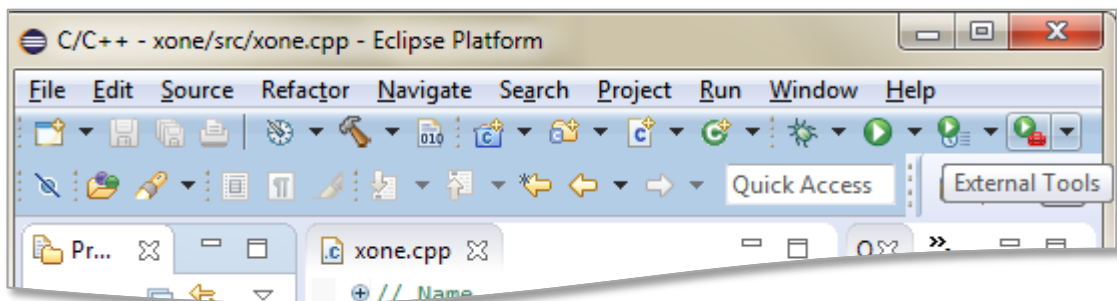


- 4) Нажав кнопку *Ok*, сохранить изменения.

Теперь мастер пакета *Doxygen* для обработки файлов текущей директории проекта доступен по команде меню *Tools/doxyWizard* или с помощью комбинации клавиш *Alt+T, Z*.

Настройка Eclipse C/C++ Development Tools для использования Doxygen

- 1) Создать C++ проект.
- 2) В панели инструментов выбрать пункт *External Tools/External Tools Configuration*.

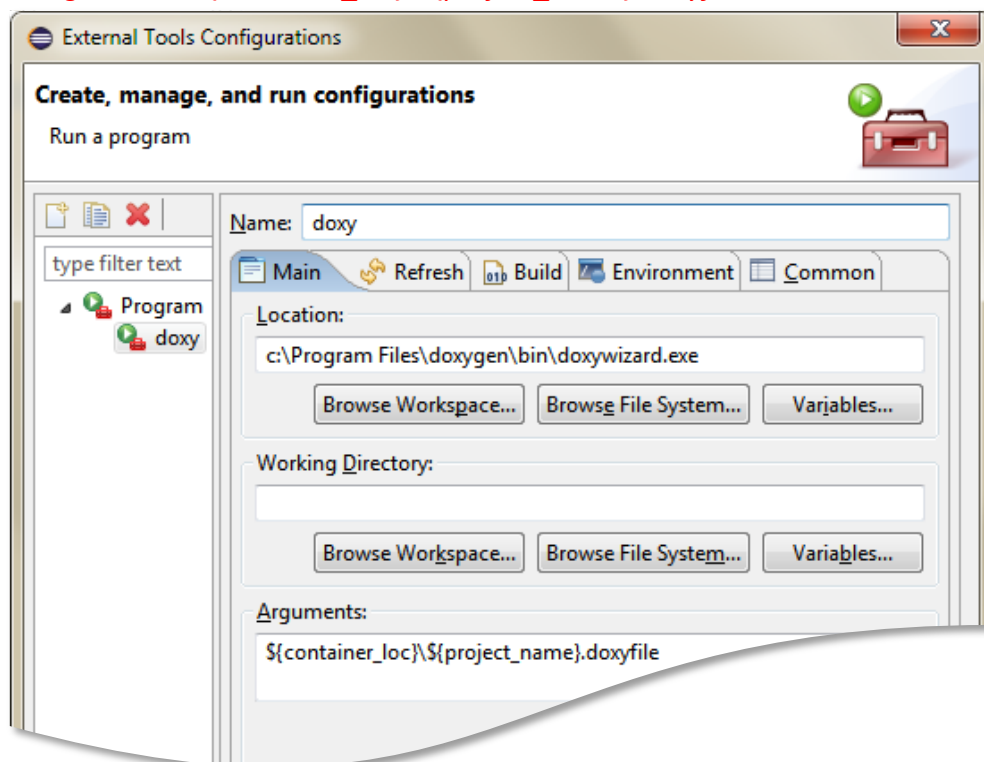


- 3) Создать новую конфигурацию со следующими параметрами:

Name: doxy

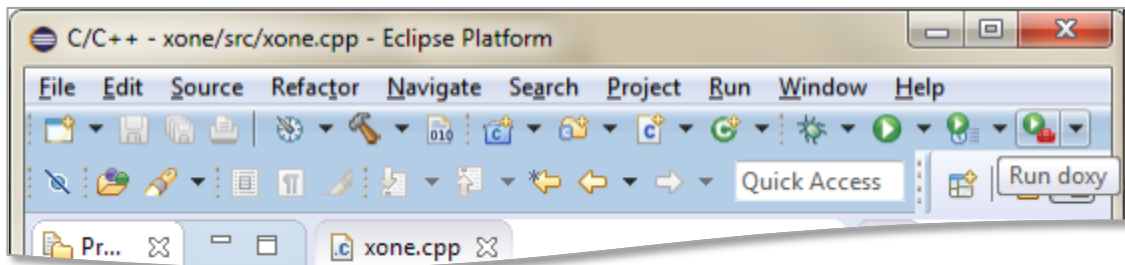
Location: C:\Program Files\doxygen\bin\doxywizard.exe

Arguments: \${container_loc}\\${project_name}.doxyfile



- 4) Нажав кнопку [Apply](#), сохранить изменения.
- 5) Нажав кнопку [Run](#), запустить мастер пакета Doxygen.

Теперь мастер пакета Doxygen для обработки файлов директории выбранного в *Project Explorer* объекта доступен из панели инструментов по кнопке [Run doxy](#).

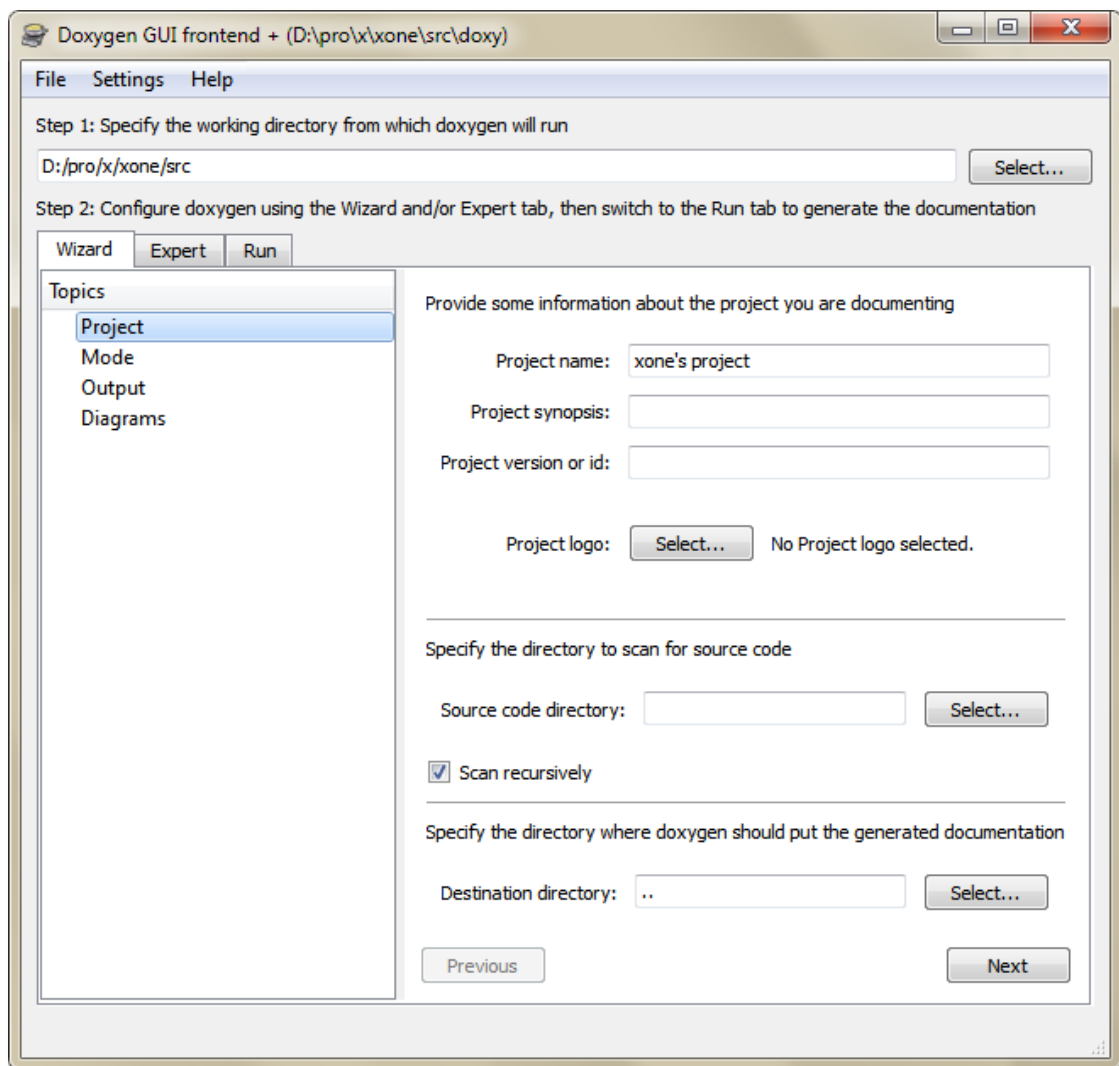


Примечание: после запуска Doxygen обратите внимание на поле

[Step 1: Specify the working directory from which doxygen will run](#)

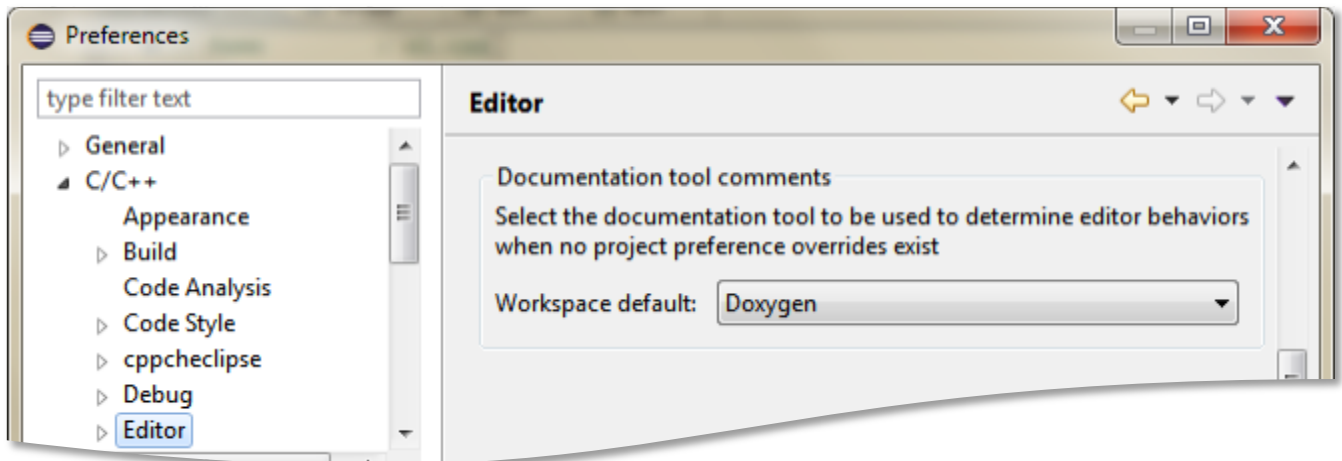
Там должен быть задан путь к директории, содержащей исходные файлы проекта. Чтобы не замусоривать эту директорию результатами работы Doxygen, рекомендуется в поле

[Destination directory](#)
ввести ".."

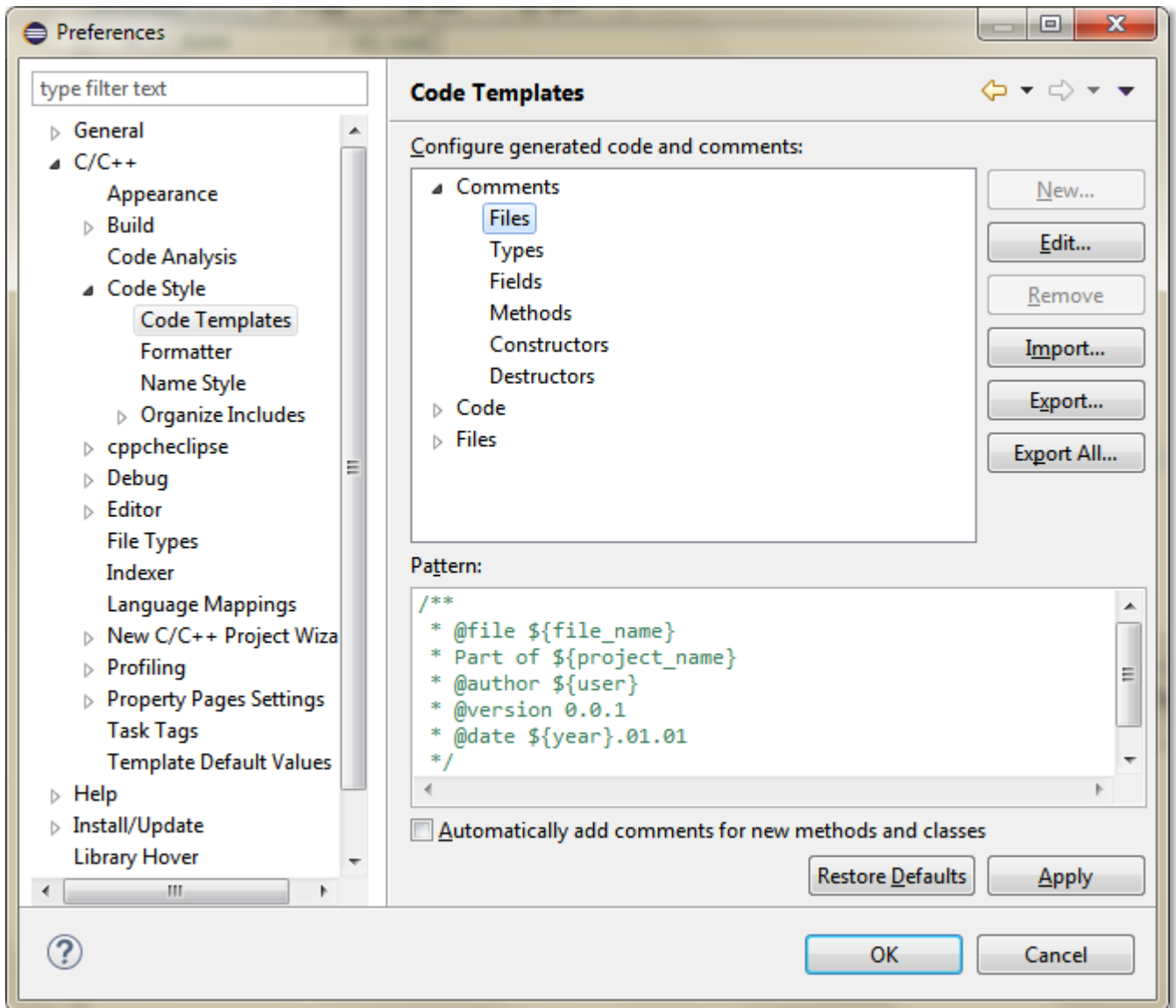


6) В основном меню выбрать пункт *Window/Preferences*, открыть раздел *C/C++ / Editor* и установить:

Workspace default: Doxygen



7) В разделе *C/C++ / Code Style / Code Templates* рекомендуется отредактировать параметры подраздела *Comments*:



Настройка мастера *Doxywizard* пакета *Doxygen*

Выполнить запуск мастера *Doxywizard* и проверить следующие опции.

- 1) Закладка *Wizard*, меню *Topics* пункт *Project*, должно быть установлено:

Project name: <название_проекта>

Scan recursively

The screenshot shows the 'Wizard' tab of the Doxywizard dialog. On the left, the 'Topics' list has 'Project' selected. The main area is titled 'Provide some information about the project you are documenting'. It contains the following fields and controls:

- Project name:** A text box containing 'task01'.
- Project synopsis:** An empty text box.
- Project version or id:** An empty text box.
- Project logo:** A 'Select...' button and the text 'No Project logo selected.'
- Specify the directory to scan for source code:**
 - Source code directory:** An empty text box and a 'Select...' button.
 - ☒ **Scan recursively**
- Specify the directory where doxygen should put the generated documentation:**
 - Destination directory:** An empty text box and a 'Select...' button.
- Navigation:** 'Previous' and 'Next' buttons at the bottom.

- 2) Закладка *Wizard*, меню *Topics* пункт *Mode*, должно быть установлено:

All Entities

Optimize for C++ output

The screenshot shows the 'Wizard' tab of the Doxywizard dialog. On the left, the 'Topics' list has 'Mode' selected. The main area is titled 'Select the desired extraction mode:'. It contains the following controls:

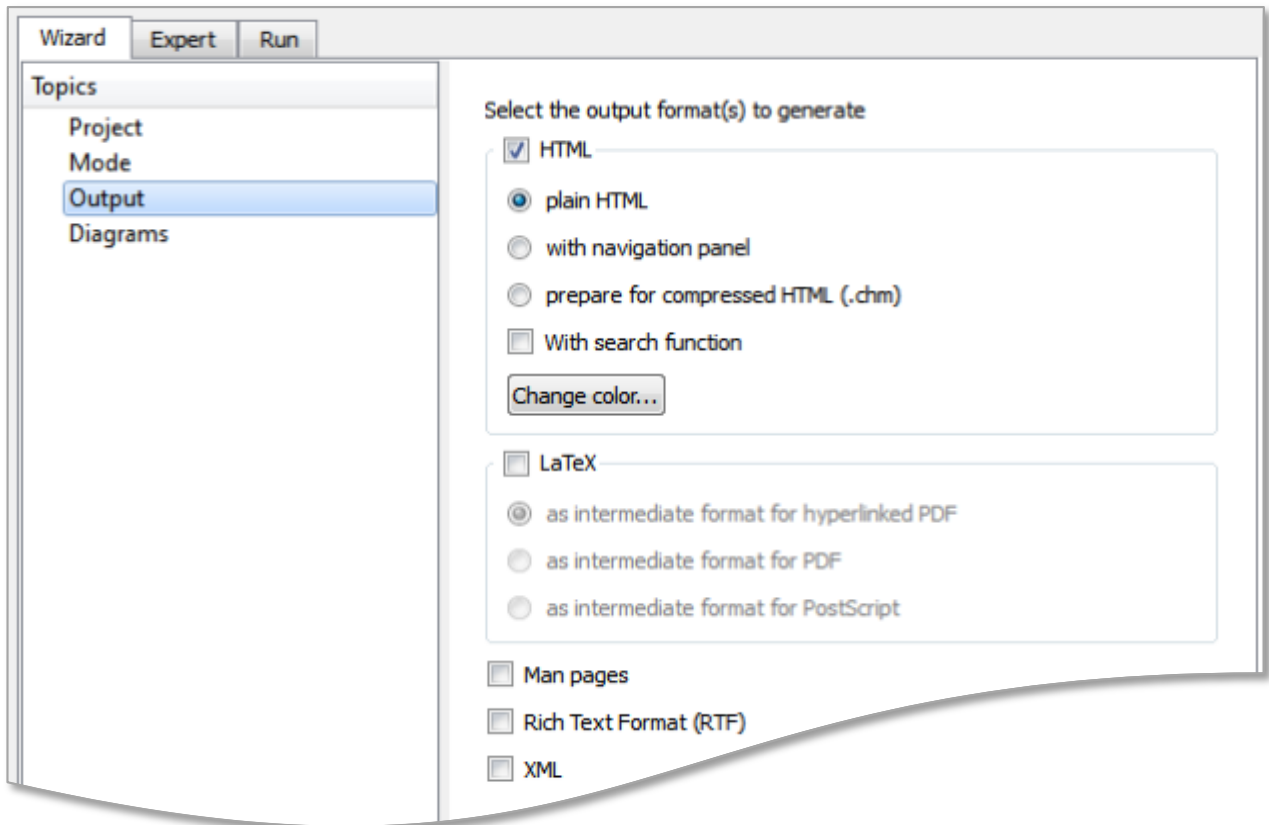
- ☐ Documented entities only
- ☒ **All Entities**
- ☐ Include cross-referenced source code in the output
- Select programming language to optimize the results for:**
 - ☒ **Optimize for C++ output**
 - ☐ Optimize for C++/CLI output
 - ☐ Optimize for C

Пункт *Output*. Должно быть установлено:

HTML

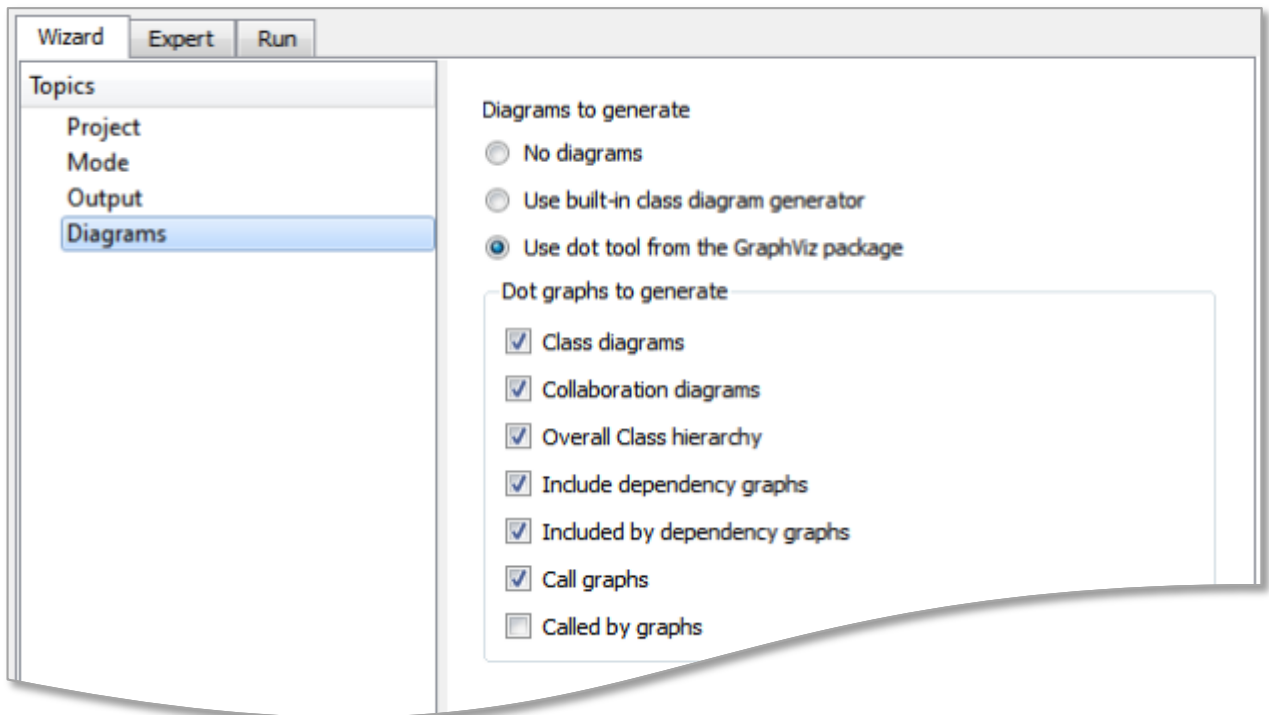
plain HTML

Остальные опции этого раздела должны быть отключены (отметки сняты).



Пункт *Diagrams*. Должно быть установлено:

Use dot tool from the GraphViz package



3) Зкладка *Expert*, меню *Topics* пункт *Project*, должно быть установлено:

OUTPUT_LANGUAGE: Ukrainian

JAVADOC_AUTOBRIEF

Пункт *Build*. Должно быть установлено:

EXTRACT_ALL

EXTRACT_PRIVATE

EXTRACT_STATIC

Пункт *Input*. Должно быть установлено:

INPUT_ENCODING: CP1251

FILE_PATTERNS: в поле ввести "*.md" и нажать кнопку "+"

EXAMPLE_PATH: в поле ввести "." и нажать кнопку "+"

Пункт *Dot*. Должно быть установлено:

MSCGEN_PATH: Путь_к_mscgen.exe

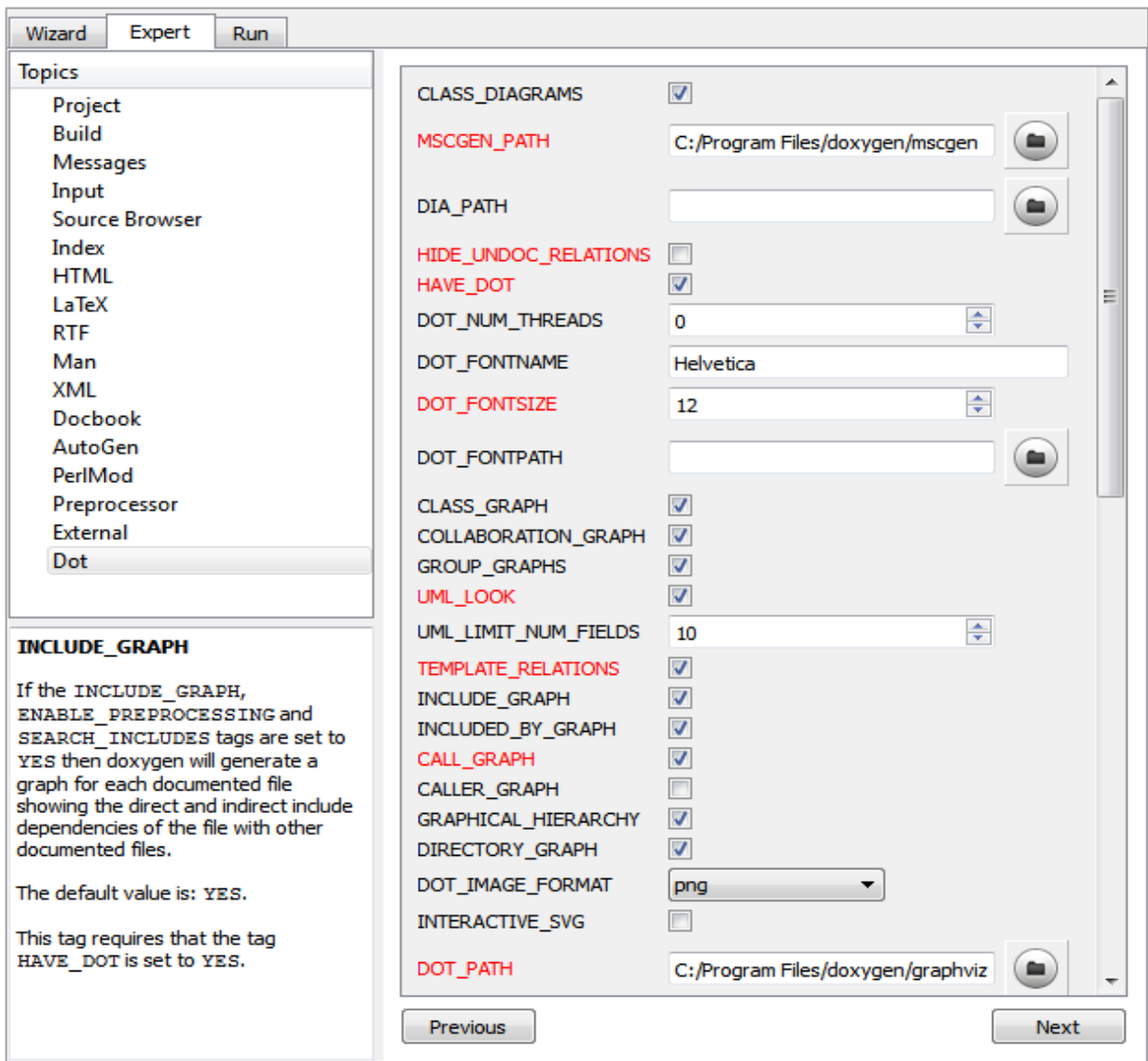
HAVE_DOT

UML_LOOK

TEMPLATE_RELATIONS

CALL_GRAPH

DOT_PATH: Путь_к_dot.exe



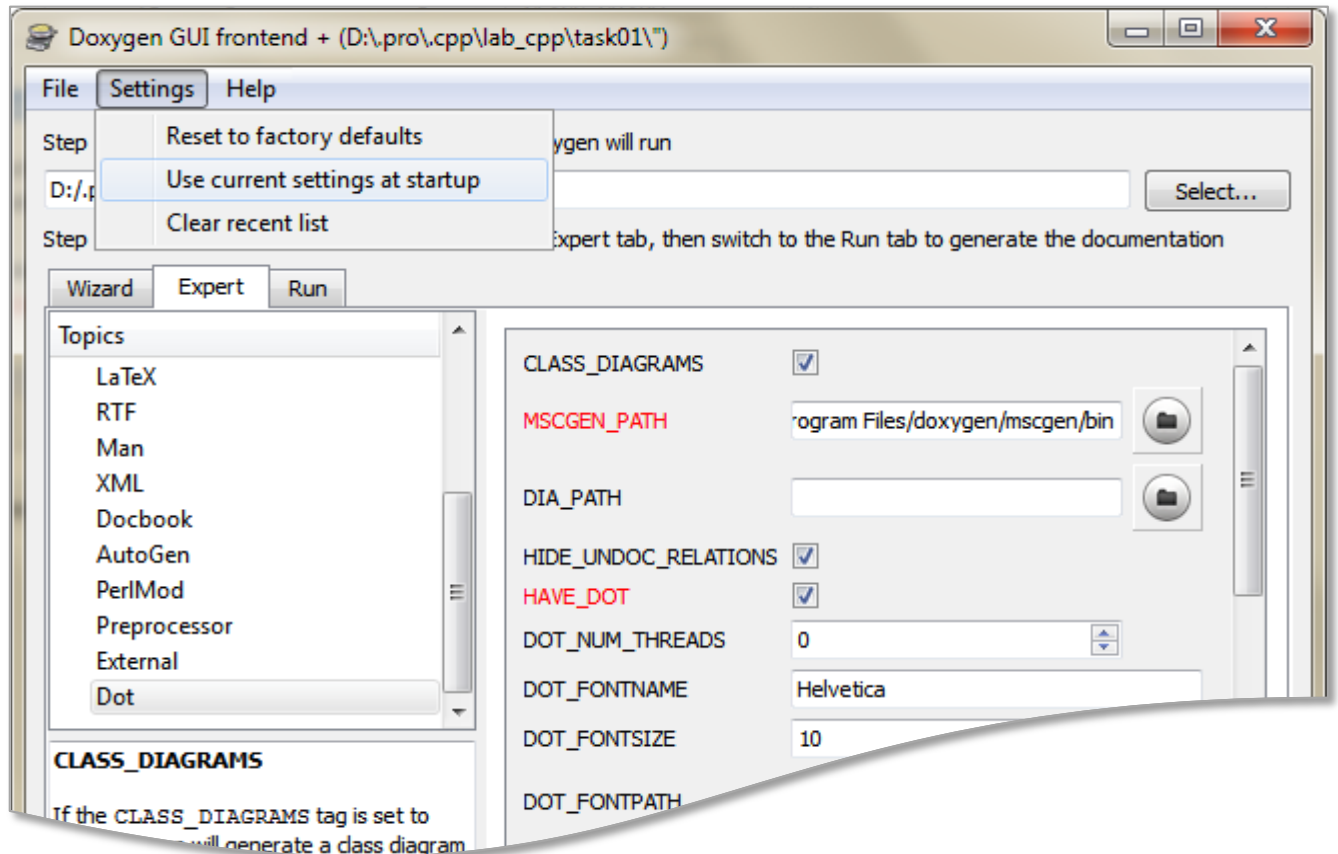
Остальные опции – по умолчанию.

Примечание: указанные пути к утилитам пакетов *Mscgen* и *Graphviz* должны соответствовать п. "Установка ПО" (см. выше).

Сохранение настроек

Сохранить настройку пакета *Doxygen* можно с помощью меню

Settings/User current settings at startup



Использование

- 1) Создать проект *Visual C++*.
- 2) Подготовить и сохранить исходный код в файлах *.cpp* и *.h*. При подготовке исходного кода использовать специальный вид комментариев (см. приложение А).
- 3) Выполнить обработку файлов с исходным кодом пакетом *Doxygen*:
 - вызвать мастер *Doxywizard* (В *Visual Studio*: меню *Tools / doxyWizard* или клавиши *Alt+T, Z*. В *Eclipse*: меню *Run / External Tools / doxy*);
 - активировать *Doxygen* (закладка *Run*, кнопка *Run doxygen*);
 - дождаться сообщения "*Doxygen has finished*" в окне "*Output produced by doxygen*".
- 4) Проверить результат – кнопка *Show HTML output*.

Примечание: Результат работы *Doxygen* сохраняется в папке *html* текущего каталога проекта (использовать файл *index.html*).

Рекомендации по комментированию исходного кода

1) Комментарии для *Doxygen* располагают непосредственно перед введением имён – объявлением классов, их полей, методов, а так же функций, переменных, констант и т.д. Комментарии в *JavaDoc* стиле:

```
/**
 * Краткий комментарий.
 * Подробное описание
 */
```

или

```
/// Краткий комментарий.
```

Можно приводить комментарии в той же строке, что и комментируемое выражение:

```
int var; /**< Текст комментария. */
```

или

```
int var; ///  
Текст комментария.
```

2) Код реализации методов и функций комментируется в обычном *C++* стиле.

3) Каждый файл проекта начинается с заголовка:

```
/**
 * @file FileName.ext
 * Назначение данного файла.
 * @author имя
 * @version x.y.z
 * @date yyyy.mm.dd
 */
```

Примечание: возможные варианты комментирования исходного кода, обрабатываемые системой *Doxygen*, подробно рассмотрены в <C:\Program Files\doxygen\html\docblocks.html>

Поддержка Markdown

Начиная с версии 1.8.0 *Doxygen* поддерживает *Markdown* – простой язык разметки.

Вики:

<https://ru.wikipedia.org/wiki/Markdown>

В папку с исходным кодом проекта рекомендуется добавить файл *readme.md* и разместить в нём с использованием разметки *Markdown* следующую информацию для отображения на главной странице:

- тема;
- цель;
- задание;
- заключение.

Примечание: список *Expert/Input/FILE_PATTERNS* должен содержать пункт *"*.md"*. Полная справка по поддержке *Markdown* в пакете *Doxygen* находится в <C:\Program Files\doxygen\html\markdown.html>

Пример комментариев для объявления класса

Файл *One.h*

```
/**
 * @file One.h
 * Объявление класса One.
 * @author xone
 * @version 0.0.1
 * @date 2015.09.01
 */

#ifndef ONE_H_
#define ONE_H_

/**
 * Предназначен для демонстрации автодокументирования класса.
 * Используются комментарии в стиле Javadoc для обработки пакетом Doxygen
 */
class One {
public:
    /**
     * Инициализация полей.
     * Используется список инициализации, тело - пустое
     */
    One();

    /**
     * Пустой деструктор.
     */
    virtual ~One();

    /**
     * Установка поля One::x.
     * @param x присваивается полю One::x
     */
    void setX(int x);

    /**
     * Чтение поля One::x.
     * @return текущее значение поля One::x
     */
    int getX() const;

private:
    /** Используется для демонстрации краткого комментария атрибута. */
    int x;
};

#endif /* ONE_H_ */
```

Пример комментариев для реализации класса

Файл *One.cpp*

```
/**
 * @file One.cpp
 * Реализация класса One.
 * @author xone
 * @version 0.0.1
 * @date 2015.09.01
 */

#include "One.h"

One::One() : x(-1) {
    // Пустое тело.
}

One::~One() {
    // Пустое тело.
}

int One::getX() const {
    return x;
}

void One::setX(int x) {
    // Используем this для доступа к атрибуту.
    this->x = x;
}
```

Пример комментариев для объявления и реализации функции

Файл *main.cpp*

```
/**
 * @file main.cpp
 * Реализация функции main().
 * @author xone
 * @version 0.0.1
 * @date 2015.09.01
 */

#include "One.h"

/**
 * Точка входа.
 * @param argc количество параметров командной строки
 * @param argv массив параметров командной строки
 * @return код завершения
 */
int main(int argc, char** argv) {
    // Обычные комментарии C++.
    // ...
    return 0;
}
```

Пример комментариев для объявления перегруженного оператора

```
/**
 * Вывод экземпляров One и его потомков в стандартный поток.
 * Для вывода в поток используется виртуальный метод,
 * вызываемый через ссылку на базовый класс (полиморфизм)
 * @param os ссылка на поток вывода
 * @param one ссылка на выводимый объект
 * @return ссылка на поток вывода
 * @remark Пример перегрузки оператора для базового класса,
 * который используется и для его потомков
 * @see One
 */
ostream& operator<<(ostream& os, One& one);
```

Пример разметки *Markdown* для отображения на главной странице

файла *readme.md*

```
# Документирование исходного кода C++      {#mainpage}

## Цель

Ознакомление со средствами автоматического документирования
проектов на языке C++ с помощью пакета
[Doxygen] (http://www.stack.nl/~dimitri/doxygen/) .

## Задание

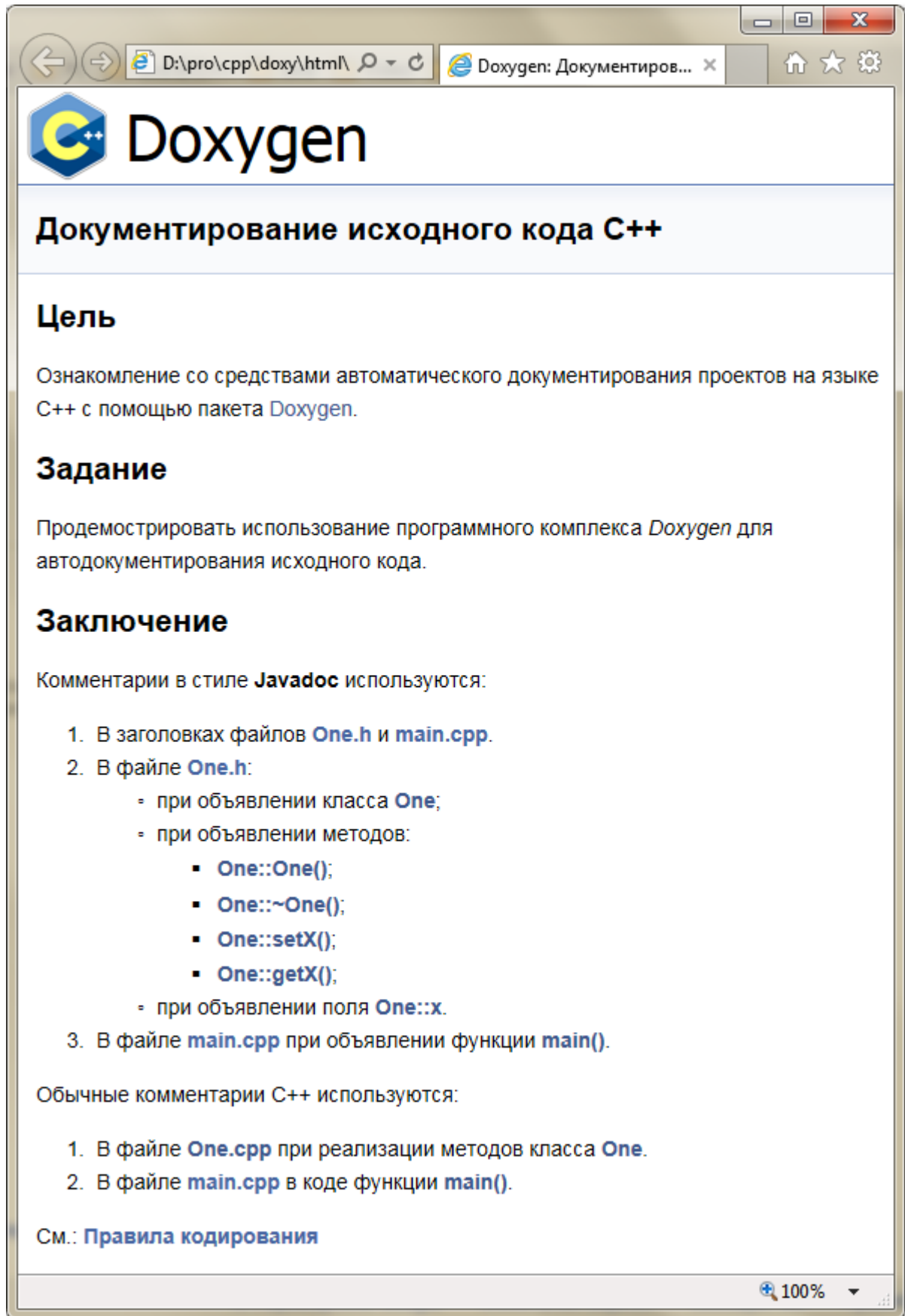
Продемонстрировать использование программного комплекса _Doxygen_
для автодокументирования исходного кода.

## Заключение

Комментарии в стиле __Javadoc__ используются:
1. В заголовках файлов One.h и main.cpp.
2. В файле One.h:
    - при объявлении класса One;
    - при объявлении методов:
        - One::One();
        - One::~~One();
        - One::setX();
        - One::getX();
    - при объявлении поля One::x.
3. В файле main.cpp при объявлении функции main().

Обычные комментарии C++ используются:
1. В файле One.cpp при реализации методов класса One.
2. В файле main.cpp в коде функции main().

См.: [Правила кодирования] (@ref coding)
```

Результат отображения содержимого *readme.md* на главной странице

Пример использования Doxygen для создания диаграмм

На рис.1 приведен пример диаграммы связей, полученной в результате обработки системой Doxygen следующего текста:

```
class Invisible { };
class Truncated : public Invisible { };
class Undocumented { };
class PublicBase : public Truncated { };
template<class T> class Templ { };
class ProtectedBase { };
class PrivateBase { };
class Used { };
class Inherited : public PublicBase, protected ProtectedBase,
                  private PrivateBase, public Undocumented, public Templ<int> {
private:
    Used *m_usedClass;
};
```

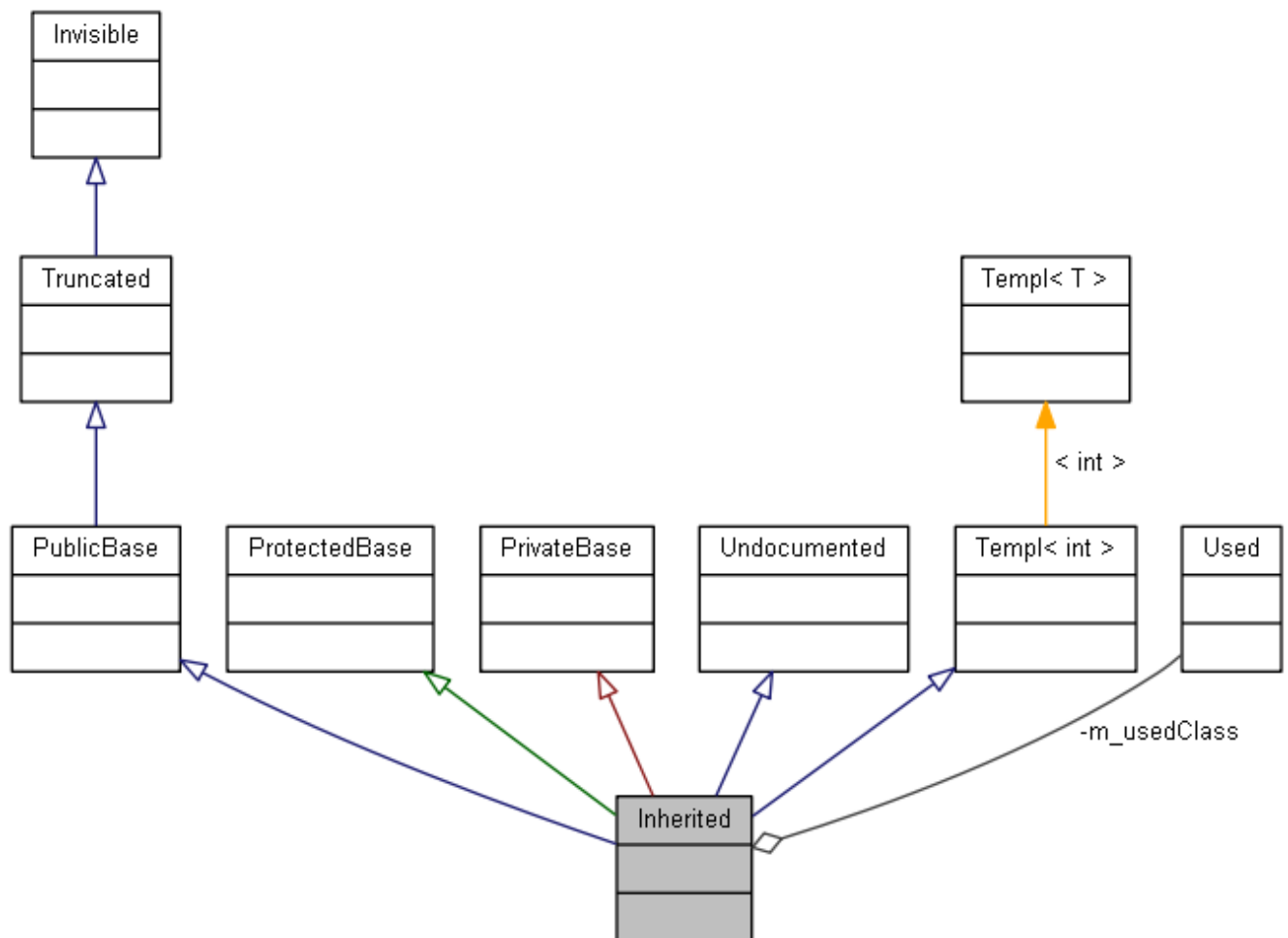


Рисунок 1 – Диаграмма связей. Сгенерирована системой Doxygen

Некоторые команды системы *Doxygen*

Команды располагаются в теле комментариев, оформленных указанным выше способом, и начинаются с символа `@` (*at-sign*) или `\` (*backslash*).

Ниже приводится краткое описание некоторых команд:

<code>@author author_name</code>	Указывается имя автора
<code>@authors</code>	То же, что и <code>@author</code>
<code>@code ... @endcode</code>	Операторные скобки определяют блок текста, который полностью включается в генерируемый документ. Стиль текста будет отличаться от обычного.
<code>@date xx.xx.xx</code>	Указывается дата
<code>@exception</code>	То же, что и <code>@throw</code>
<code>@file file_name.ext Описание</code>	Указывает на то, что документируется файл с исходным кодом по имени " <code>file_name.ext</code> "
<code>@param param_name Описание</code>	Описание параметра функции или метода. Имя параметра – " <code>param_name</code> ". Можно указать направление параметра атрибутами <code>[in]</code> , <code>[in,out]</code> или <code>[out]</code> , например, <code>@param[out]</code> ...
<code>@remark Текст примечания</code>	Содержит текст примечания
<code>@remarks</code>	То же, что и <code>@remark</code>
<code>@return Описание</code>	Описание возвращаемого результата
<code>@returns</code>	То же, что и <code>@return</code>
<code>@sa</code>	То же, что и <code>@see</code>
<code>@see references</code>	Перекрестная ссылка. Несколько ссылок можно разделять символом <code> </code>
<code>@throw <exception_object> Описание</code>	Описание выбрасываемого исключения
<code>@throws</code>	То же, что и <code>@throw</code>
<code>@version version_info</code>	Информация о версии

Примечание: полная справка по командам системы *Doxygen* находится в <C:\Program Files\doxygen\html\commands.html>