



# Software Documentation

## Lecture Outline



- 1)** Структура документации
- 2)** Файлы **README**, **FAQ**, **CHANGELOG**
- 3)** Руководство пользователя
- 4)** Инструменты

# Documentation Structure

## ✓ Проектная документация

- Описание продукта в общих чертах (подсистемы, технологии)
- Спецификации требований

## ✓ Техническая документация

- Расписываются API, структуры данных, алгоритмы
- Применяются генераторы (JavaDoc, NDoc, Doxygen, etc.)

## ✓ Пользовательская документация

- Вводное руководство (tutorial)
- Справочная система (информация, ключевые слова, алфавитный указатель)

## ✓ Маркетинговая документация

- слоган, логотип
- ключевой функционал в завлекательной форме
- место продукта на IT-рынке, преимущества

# Documentation Structure



Спецификации  
требований



Техническая и  
проектная  
документация



Документация  
по  
тестированию



План контроля  
качества



План управления  
проектом



План  
сопровождения



Руководство  
пользователя



План управления  
конфигурациями

# README File

- ✓ Содержание
- ✓ Аппаратные и программные требования
- ✓ Инструкции по установке
- ✓ Инструкции по эксплуатации (ключевые)
- ✓ Файл манифеста (перечень файлов ПО)
- ✓ Информация о копирайте и лицензии
- ✓ Контактная информация для дистрибьютора или программиста
- ✓ Известные баги
- ✓ Поиск и устранение неисправностей (troubleshooting)
- ✓ Разработчики и признательность
- ✓ Журнал изменений (changelog) (обычно, для программистов)
- ✓ Секция новостей (обычно, для пользователей)

# CHANGELOG, FAQ, TODO

## ❑ Файл истории изменений продукта (**ChangeLog**)

```
YYYY-MM-DD John Doe <johndoe@example.com>  
  
* myfile.ext (myfunction): my changes made additional changes  
  
* myfile.ext (unrelated_change): my changes made to myfile.ext but completely  
  unrelated to the above  
  
* anotherfile.ext (somefunction): more changes
```

❑ Файл с наиболее часто задаваемыми вопросами (**FAQ**) и ответами на них (они могут быть в справочной системе, а могут быть вынесены отдельно)

❑ Файл с перечнем планируемых изменений (**TODO**)

# User Guide & Reference Manual

- Аннотация, в которой приводится краткое изложение содержимого документа и его назначение
- Введение, содержащее ссылки на связанные документы и информацию о том, как лучше всего использовать данное руководство
- Страница содержания
- Главы, описывающие, как использовать, по крайней мере, наиболее важные функции системы
- Глава, описывающая возможные проблемы и пути их решения
- Часто задаваемые вопросы и ответы на них
- Где ещё найти информацию по предмету, контактная информация
- Глоссарий и предметный указатель

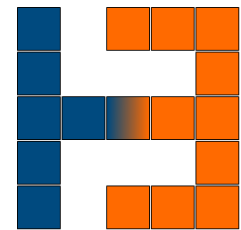
<http://www.techscribe.co.uk/ta/how-to-write-user-documentation.htm>

# Tools

Microsoft HTML Help Workshop

Helpinator

Helpndoc



Сайт разработчиков технической документации

<http://techwriters.ru>

# Documentation Generators

JavaDoc

NDoc (.NET)

Doxygen



# JavaDoc Descriptors

Дескриптор	Описание	Применим к
<b>@author</b>	Автор	класс, интерфейс
<b>@version</b>	Версия. Не более одного дескриптора на класс	класс, интерфейс
<b>@since</b>	Указывает с какой версии доступно	класс, интерфейс, поле, метод
<b>@see</b>	Ссылка на другое место в документации	класс, интерфейс, поле, метод
<b>@param</b>	Входной параметр метода	метод
<b>@return</b>	Описание возвращаемого значения	метод
<b>@exception <i>имякласса описание</i></b> <b>@throws <i>имякласса описание</i></b>	Описание исключения, которое может быть послано из метода	метод
<b>@deprecated</b>	Описание устаревших блоков кода	класс, интерфейс, метод
<b>{@link reference}</b>	Ссылка	класс, интерфейс, поле, метод
<b>{@value}</b>	Описание значения переменной	статичное поле

# JavaDoc Example

```
/**
 * <p>Проверяет, допустимый ли ход.</p>
 * <p>Например, чтобы задать ход e2-e4, напишите isValidMove(5,2,5,4);
 * Чтобы записать рокировку, укажите, откуда и куда ходит король.
 * Например, для короткой рокировки чёрных запишите isValidMove(5,8,7,8);</p>
 *
 * @param fromCol Вертикаль, на которой находится фигура (1=a, 8=h)
 * @param fromRow Горизонталь, на которой находится фигура (1...8)
 * @param toCol Вертикаль клетки, на которую выполняется ход (1=a, 8=h)
 * @param toRow Горизонталь клетки, на которую выполняется ход (1...8)
 * @return true, если ход допустим, и false, если недопустим
 */

bool isValidMove(int fromCol, int fromRow, int toCol, int toRow)
{
    ...
}
```

# Doxygen Example (C++ style)

```
///! Класс для логирования всех событий и действий пользователя, для которых разработана
система кодов \sa \ref logGroup
class CLogger
{
    ...

    /*! Формирует строку с датой и точным временем события для лога
    *   \param [in] ftime - системное время события
    *   \return время события в отформатированном строковом виде "dd/mm/yyyy hh:mm:ss.msec"
    */
    CString TimeString( FILETIME& ftime );

    ...
};

/*! \defgroup logGroup Коды записей лога
@{
                                                                    */

#define LR_USER_LOGIN          0          ///! Событие входа пользователя в систему
#define LR_USER_LOGOUT        1          ///! Событие выхода пользователя из системы
...
/* }@ */
```

# Doxygen Example

 CLogger

Класс для логирования всех событий и действий пользователя, для которых разработана система кодов

 CNetworkClient

▼  CScanUnit

 CCellUnit

 CRouteUnit

 CSubTask

 CTask

 CTaskList

 CUserList

# Doxygen Example

## Защищенные данные

---

FILE *	<b>pLogFile</b>
char	<b>cFileName</b> [255]
unsigned int	<b>nOffset</b>
unsigned int	<b>nPackOffset</b>
unsigned int	<b>nPackNextOffset</b>

## Подробное описание

---

Класс для логирования всех событий и действий пользователя, для которых разработана система кодов

**См. также**

[Коды записей лога](#)

# Doxygen Example

**CString CLogger::TimeString ( FILETIME & **ftime** )**

Формирует строку с датой и точным временем события для лога

## **Аргументы**

[in] **ftime** - системное время события

## **Возвращает**

время события в отформатированном строковом виде "dd/mm/yyyy hh:mm:ss.msec"