Виноградова М.В.

МГТУ им. Н.Э. Баумана, кафедра СОИУ (ИУ5)

Средства составления проектной документации

Учебно-методические материалы «Средства составления проектной документации» представляют собой методические указания к лабораторной работе по дисциплине «Технологии разработки программного обеспечения» (по направлению магистерской подготовки).

Цель работы:

Изучить базовые возможности и получить навыки создания документации на основе исходного кода программ с применением CASE средств.

В материалах рассмотрены принципы составления проектной документации на основе исходного кода программ. Приведены инструкции по работе и команды документирования исходных кодов для автоматического составления документации в среде Doxygen. Рассмотрены примеры использования указанных средств. В заключительной части методических указаний приведены контрольные вопросы, список рекомендуемой литературы и пример задания.

Ознакомившись с методическими указаниями и разобрав приведенные в нем примеры, студент может получить у преподавателя свой вариант задания и приступить к его выполнению.

Оглавление

Теория. Использование Doxygen для документирования кода

Doxygen – это кроссплатформенная система документирования кода с поддержкой языков C++, C, Java, Objective-C, PHP, C# (список можно уточнить на сайте проекта).

Для создания документации достаточно просто писать комментарии в коде, придерживаясь нескольких простых правил.

Doxygen умеет анализировать исходный код проекта и создавать удобную документацию в формате HTML, Latex, RTF, XML, man, CHM.

Использование Doxygen

Использовать Doxygen просто – для этого надо просто запустить программу, указав ей путь к файлу с настройками. Файл с настройками представляет собой текстовой файл, который можно редактировать как в текстовом редакторе, так и с помощью специальных программ, например Doxygate.

В настройках описывается внешний вид документации, какие сущности и отношения между ними следует включать в нее, имя проекта, путь к анализируемым файлам и так далее.

Комментирование в стиле Doxygen

Doxygen поддерживает несколько стилей комментариев, например:

```
/**
Комментарии
*/
```

Обратите внимание, что последовательность символов /** сообщает программе, что начинается комментарий предназначенный для нее. Начиная с этого места и до завершающих символов */ следуют комментарии.

Есть несколько директив, которые помогают Doxygen составить грамотную документацию. Вот основные:

@brief **\brief**	Краткое описание комментируемой сущности. Пример: @brief Функция для поиска пользователя в базе данных
@param **\param**	Параметры передаваемые функции. Пример: `@param name Имя пользователя`
@return **\return**	Возвращаемое функцией значение. Пример: `@return Информация о пользователе`

@throw \throw Исключения выбрасываемые функцией.

Пример:

```
@throw DatabaseError Если произошла ошибка при подключении
к базе данных
```

Вместо собачки **@** можно использовать слэш **\.**

Пример комментариев для класса:

```
/**
@brief Класс для работы с базой данных

Осуществляет подключение к базе при создании
и закрывает соединение при уничтожении
*/
class Database
{
public:
/**
@brief Конструктор
@param connectionString Строка подключения к базе данных
@throw ConnectionError Если подключение не удалось
*/
   explicit Database(const std::string& connectionString);
/**
@brief Поиск пользователя в базе данных

Данная функция делает выборку из базы данных
по имени пользователя и возвращает структуру с информацией
о нем. Ожидается, что соединение с базой данных установлено
и пользователь существует
@param name Имя пользователя
@return Информация о пользователе
@throw DatabaseError Если произошла ошибка при подключении
к базе данных
@throw InvalidRequest Если пользоваделя в базе не существует
*/
   CustomerPtr GetCustomer(const std::string& name);
};
```

Для разделения краткой и детальной информации можно использовать пустую строку (см. выше) или запись вида:

```
/// Класс для работы с базой данных
/** Осуществляет подключение к базе при создании
и закрывает соединение при уничтожении */
```

первая строка будет краткой информацией, остальные – детальной.

Комментарии указываются перед объектам, которые описывают.

Для написания комментария после объекта (на той же строке) используют запись вида:

```
Int number;  ///< Комментарий
```

Для комментирования на уровне проекта, а не файла служат следующие директивы:

@mainpage Основная страница проекта, то с чего начинается просмотр документации.

Пример:

```
@mainpage Приложение для учета пользователей
```

@page Дополнительные страницы проекта. Я рассматриваю их как логически обособленные части проекта. Пример:

```
@page Database Database
Утилиты для работы с базой данных
```

@ref Ссылки на страницы проекта, с их помощью можно организовать ссылки с главной страницы проекта на страницы подпроектов. Пример:

```
@ref Database Утилиты для работы с базой данных
```

Для организации четкой структуры я рекомендую в каждом подпроекте создавать файл **description.h** в котором будет директива **@page**, описание для чего нужен этот подпроект и принципы работы с ним.

В свою очередь в корне проекта я также создаю файл **description.h** в котором немного рассказано о проекте в целом и приведены ссылки на его части:

```
/**
@mainpage Приложение для учета пользователей
Состоит из следующих частей:
- @ref Database Утилиты для работы с базой данных
*/
```

Чтобы привести фрагмент кода (например пример работы с классом) используется конструкция @code – @endcode:

```
/**
Пример использования Foo:
@code
Foo f();
f.Run();
@endcode
*/
```

Списки можно создавать с помощью символа - (минус). Пример:

```
/**
Список:
- Первый пункт
- Второй пункт
*/
```

Перечисления я оформляю так:

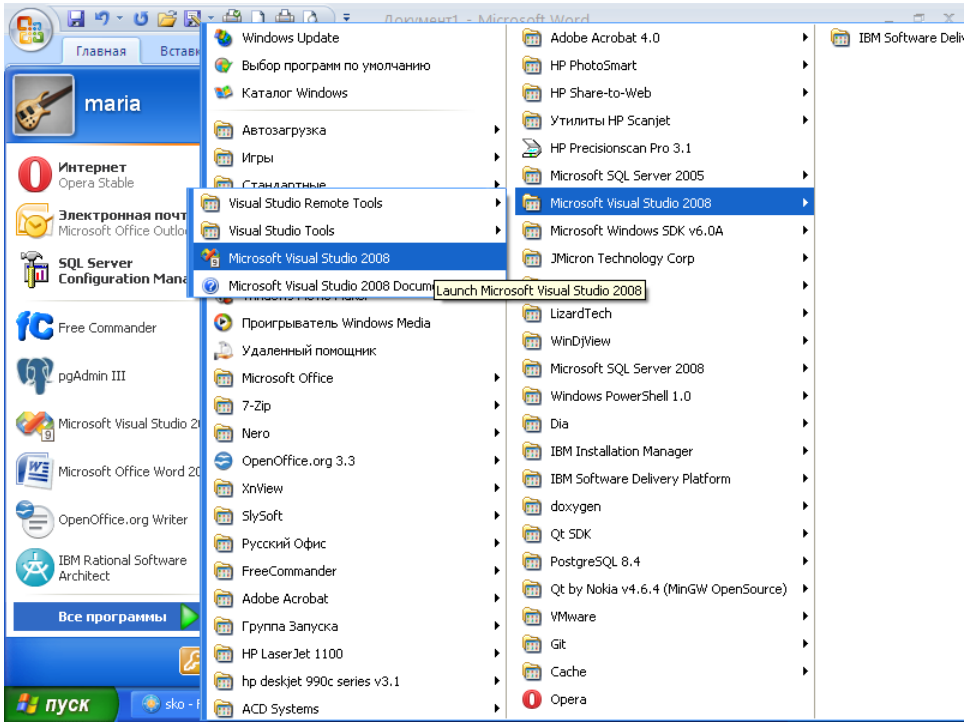
```
/**
@brief Режимы устройства
*/
enum Mode
{
   Mode_On,  /**< Включено */
   Mode_Off  /**< Выключено */
};
```

В большинстве случаев приведенных директив достаточно, с полным списком вы можете ознакомиться в руководстве.

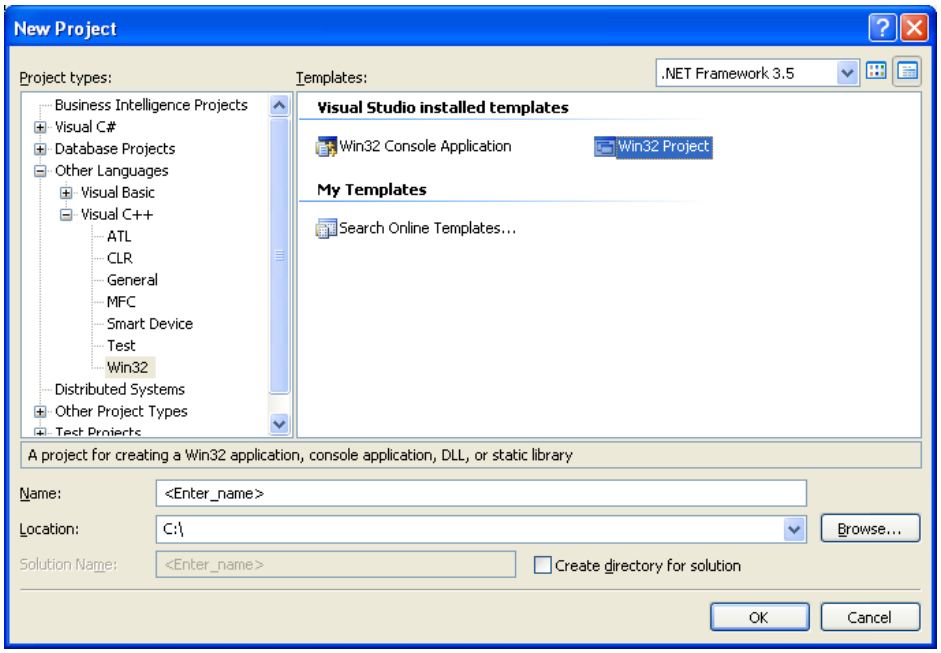
Практика – пример создания документации

Создание классов программного проекта в MS Visual Studio

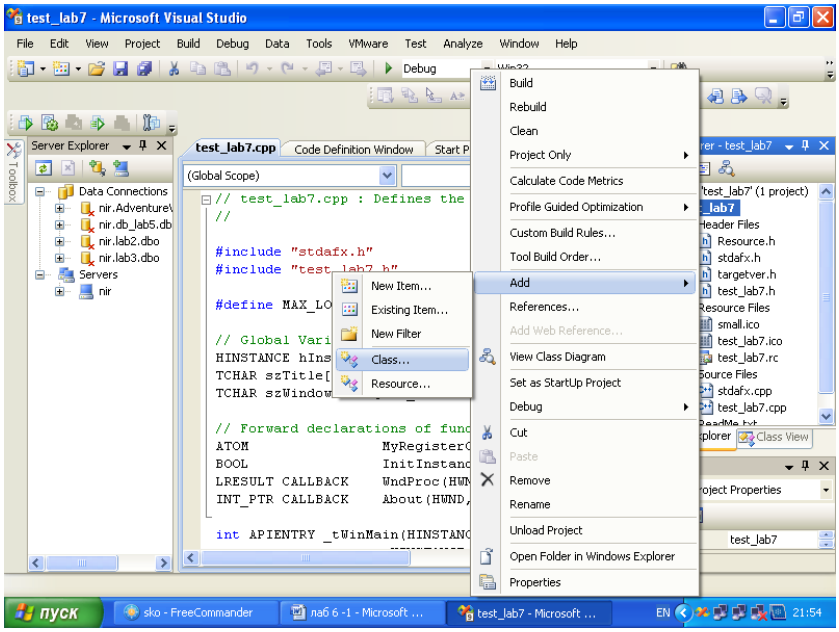
Открыть MS Visual Studio:



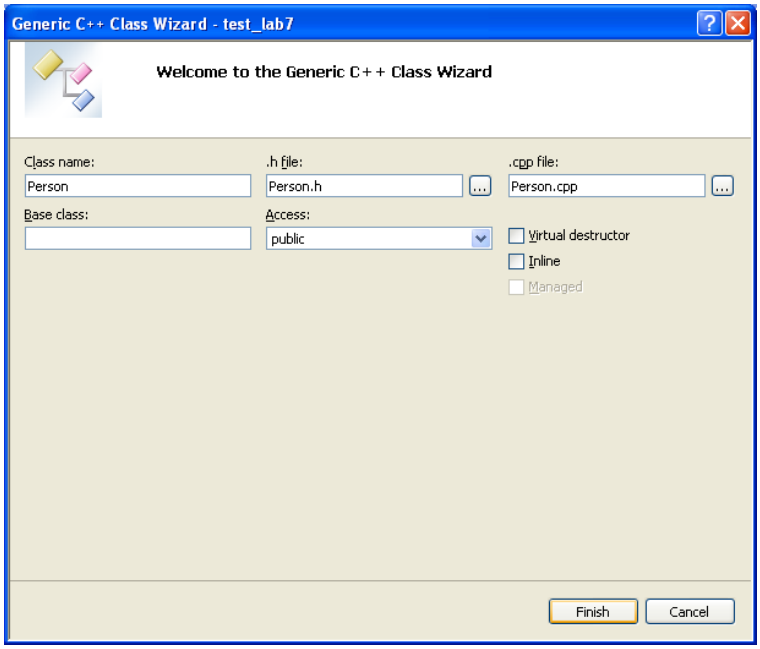
Создать новый проект и добавить в него несколько классов (по предметной области своего варианта). Выбрать в меню File – New – Project, указать тип проекта (C++ или C#), например Win32 для C++ , ввести имя проекта и путь размещения:



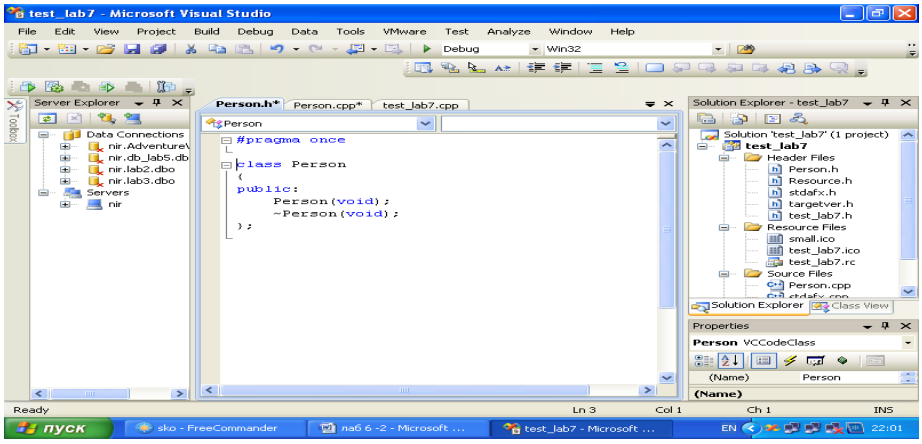
Добавить в проект новые классы. Для добавления класса из контекстного меню проекта выбрать Add – Class:



Далее в мастере указать тип класса – C++ и нажать Add, после чего указать имя класса (названия файлов остаются по умолчанию) и параметры (при необходимости):



Нажать Finish. Класс будет создан и открыт для редактирования:



В класс вручную добавить переменные и функции, например:

```
#pragma once

class Person
{
protected:
     char *_fio;
     int _age;
     char* _addr;
public:
     Person(void);
     Person(char * fio,int age, char * addr);
     ~Person(void);
     char * getFullInfo();
     bool changeAddr(char *newAddr);
};
```

Добавление в проект комментариев специального формата

Добавить в классы комментарии в формате Doxygen:

- для файлов (.h, .cpp),

- для класса,

-для переменных класса,

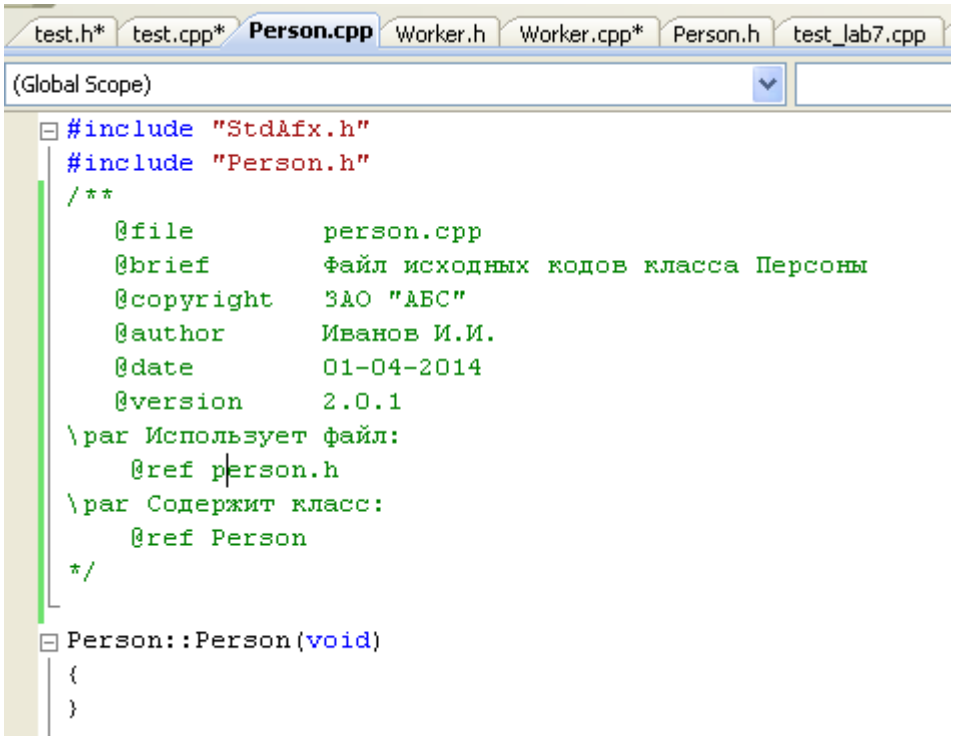
-для функций (с указанием параметров и возвращаемого значения)

Добавить комментарии краткие и детальные. Например, в заголовочном файле:

```
#pragma once
/**
   @file         person.h
   @brief        Заголовочный файл класса Персоны
   @copyright    ЗАО "АБС"
   @author       Иванов И.И.
   @date         01-04-2014
   @version      2.0.1
\par Использует классы:
    @ref Person
\par Содержит класс:
    @ref Person
*/

/// Класс описания персоны
/** Содержит данные персоны и методы для работы с ними
*/
class Person
{
protected:
    char *_fio;      ///< ФИО персоны
    int _age;        ///< возраст
    char* _addr;     ///< адрес в произвольном формате
public:
```

```
public:
    /// Конструктор по умолчанию
    /** Создает персону без инициализации полей
    */
    Person(void);

    ///  Конструктор с заполнением полей класса
    /** Создает персону и инициализирует ее поля
     \param fio ФИО создаваемой персоны
     \param age возраст создаваемой персоны
     \param addr адрес создаваемой персоны
    */
    Person(char * fio,int age, char * addr);

    /// Деструктор освобождает ресурсы
    ~Person(void);

    /// Возвращает полную информацию о персоне
    /** Если сведений нет, то возвращает сообщение об ошибке
     \return строку и данными о персоне в формате (ФИО, возраст, адрес)
     */
    char * getFullInfo();

    /** \brief Изменяет адрес персоны

    Перед изменением проверяет формат строки с адресом (код, город, улица, дом-копрус-номер)
    @param newAddr новый адрес персоны
    @return успешность выполнения действия */
    bool changeAddr(char *newAddr);
};
```

В файле кодов:



Добавить в проект комментарии в формате Doxygen:

- для проекта,

- для ссылок на файлы.

Добавить комментарии краткие и детальные. Например, для описания проекта

```
/**
    @mainpage   Проект LAB7 для тестирования Doxygen

    Используется в лабораторной работе номер 6
    по курсу Технологии Проектирования
   @copyright   ЗАО "АБС"
   @author      Иванов И.И.
   @date        01-04-2014
\par Использует классы:
- @ref Person
- @ref Worker
\par Содержит файлы:
- @ref person.h
- @ref person.cpp
- @ref worker.h
- @ref worker.cpp
*/
```

Для описания раздела (модуля):

```
/**
    @page   Worker
    @brief Компонент для работы с сотрудниками

    Содержит классы сотрудника, организации и должности
*/
```

Описания проекта и модуля можно помещать в отдельные файлы или любые файла проекта (например, заголовочные).

Использование средств автоматического документирования

Запустить программу Doxygen

Выполнить настройку генерации справки.

Указать в первом окне настроек (вкладка Project):

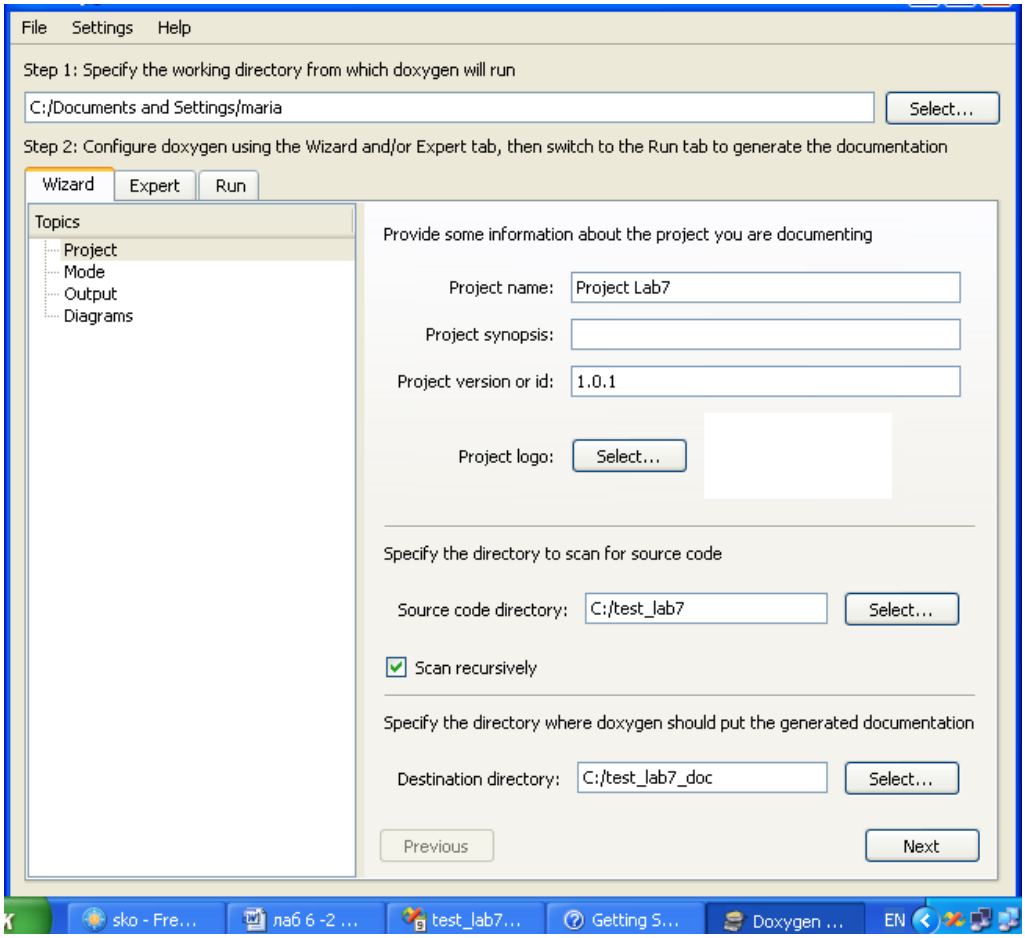
- путь к рабочему каталогу (например, каталог пользователя),

- название проекта,

- путь к исходному коду (папка с исходниками проекта),

- флаг рекурсивного сканирования,

- путь к создаваемой документации.

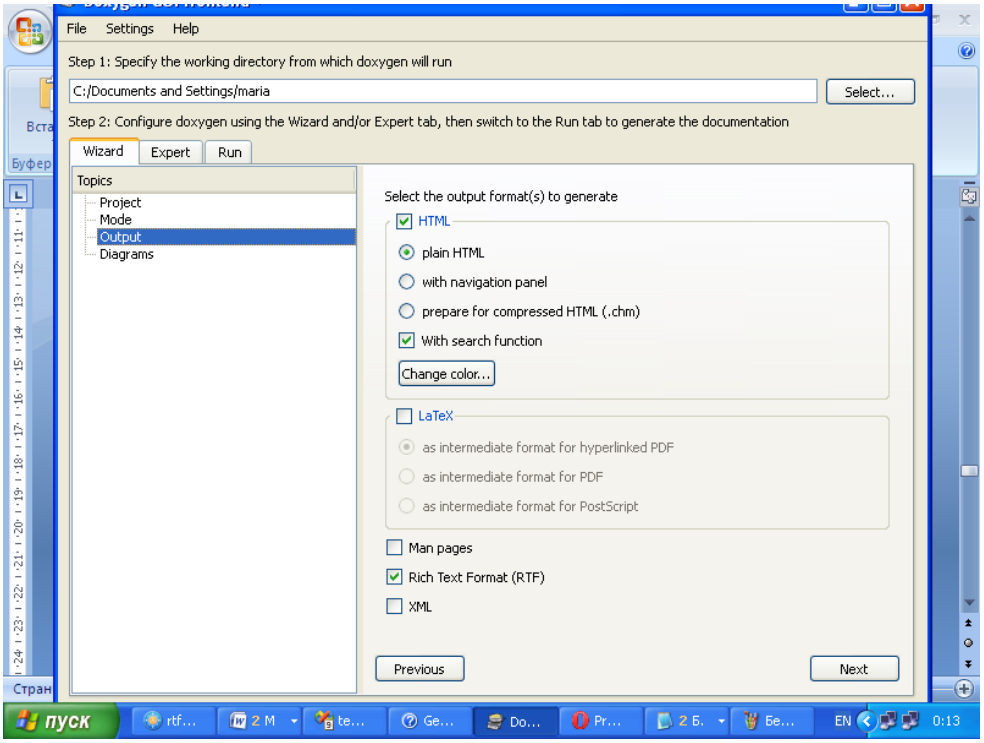


Указать на вкладке Mode:

- Язык программирования проекта.

Указать на вкладке Output:

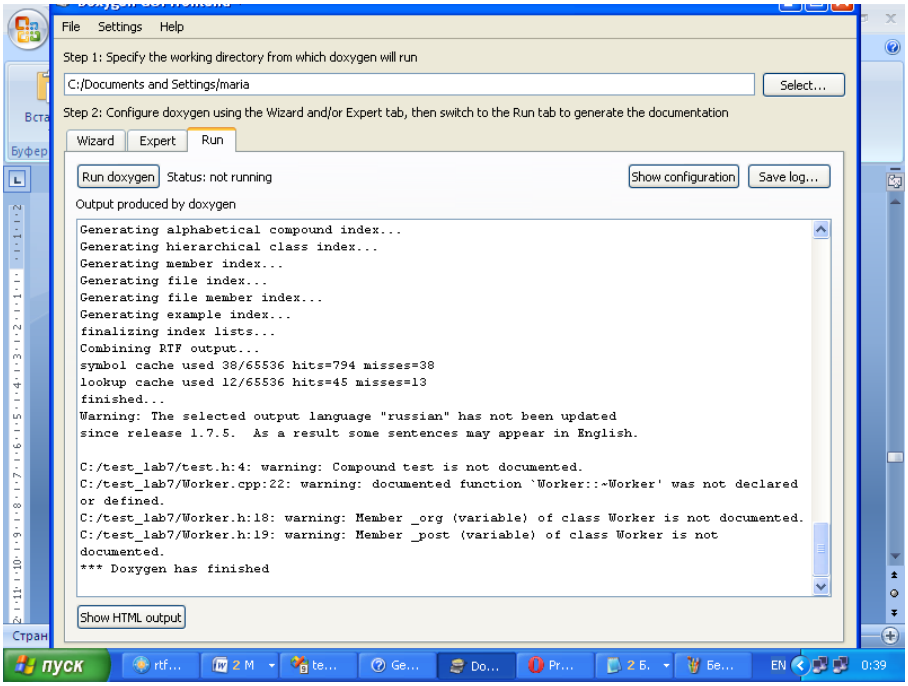
- Формат выходных документов (простой HTML и RTF).



Перейти к панели Expert и в ее вкладках указать:

- (Project) - Выходной язык – Русский,
- (Project) - Кодировка Windows-1251,
- (Input) Кодировка Windows-1251,

Перейти на вкладку Run и нажать Run Doxygen. На экране будут отображены сообщения о генерации документов по исходному коду.

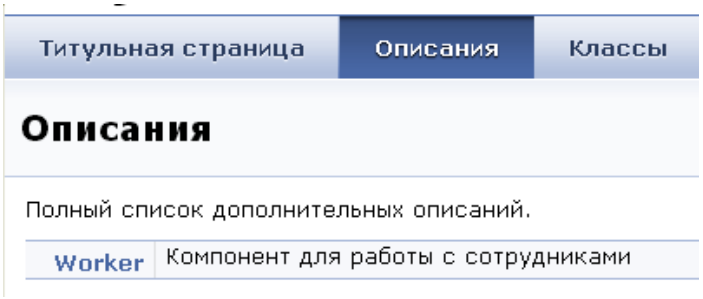


Если генерация прошла успешно, то откроется для нажатия кнопка Show HTML Output. При нажатии на нее в окне браузера откроется главная страница созданной документации.

Примеры результатов документирования

Страница проекта:

Страница Описаний (модулей):



Страница классов:

Project Lab7 1.0.1

Титульная страница | Классы | Файлы

Классы | Алфавитный указатель классов | Иерархия классо

Классы

Классы с их кратким описанием.

Person	Класс описания персоны
Worker	Класс описания сотрудника

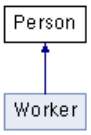
Класс Персоны (фрагмент):

Класс Person

Класс описания персоны Подробнее...

```
#include <Person.h>
```

Граф наследования:Person:



Полный список членов класса

Открытые члены

	Person (void) Конструктор по умолчанию
	Person (char *fio, int age, char *addr) Конструктор с заполнением полей класса
	~Person (void) Деструктор освобождает ресурсы
char *	**getFullInfo** () Возвращает полную информацию о персоне
bool	**changeAddr** (char *newAddr) Изменяет адрес персоны

Защищенные данные

char *	**_fio** ФИО персоны

Подробное описание

Класс описания персоны

Содержит данные персоны и методы для работы с ними

Конструктор(ы)

Person::Person (void)

Конструктор по умолчанию

Создает персону без инициализации полей

```
Person::Person ( char * fio,
                 int    age,
                 char * addr
               )
```

Конструктор с заполнением полей класса

Создает персону и инициализирует ее поля

Аргументы:

- **fio** ФИО создаваемой персоны
- **age** возраст создаваемой персоны
- **addr** адрес создаваемой персоны

Методы

Методы

bool Person::changeAddr (char * newAddr)

Изменяет адрес персоны

Перед изменением проверяет формат строки с адресом (код, город, улица, дом-копрус-номер)

Аргументы:

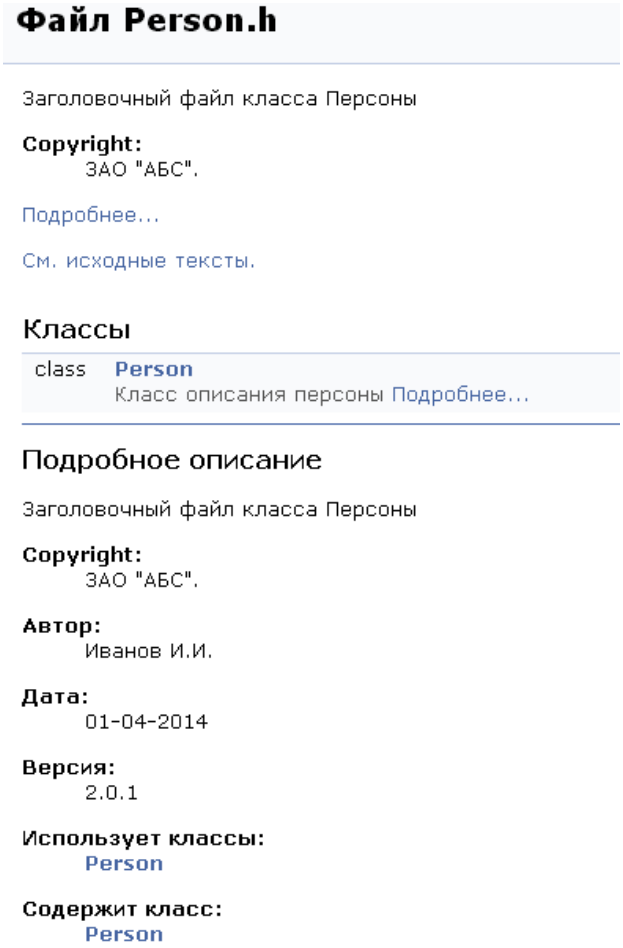
- **newAddr** новый адрес персоны

Возвращает:

успешность выполнения действия

char* Person::getFullInfo ( )

Описание файла:



Файл Person.h

Заголовочный файл класса Персоны

Copyright:
ЗАО "АБС".

Подробнее...

См. исходные тексты.

Классы

class **Person**
Класс описания персоны Подробнее...

Подробное описание

Заголовочный файл класса Персоны

Copyright:
ЗАО "АБС".

Автор:
Иванов И.И.

Дата:
01-04-2014

Версия:
2.0.1

Использует классы:
Person

Содержит класс:
Person

Для просмотра документов в формате RTF открыть файл RTF из каталога выходной документации.

Изменение документации

Если необходимо изменить документацию, то следует:

1) Открыть файлы с исходным кодом проекта,
2) Изменить в них комментарии,
3) Сохранить файлы проекта,
4) Заново запустить программу генерации документации.
5) Просмотреть полученный результат (обновить окно браузера).

Задание по документированию

1. Если система документирования Doxygen не установлена, то установить ее.
2. Создать новый прокт в Visual Studio.
3. Добавить в проект класс по своему варианту (с методами и свойствами).
4. Выполнить комментирование кода по правилам Doxygen и сохранить проект.
5. Запустить Doxygen и выполнить настройку проекта.
6. Провести документирование и просмотреть полученный результат (html и RTF).
7. Продемонстрировать:
    - краткое и подробное описание класса,
    - краткое и подробное описание функций класса,
    - описание входных и выходных параметров функций,
    - описание переменных класса,
    - описание проекта,
    - описание файла (h и cpp) и добавление ссылки на файл в описание проекта.
    - описание производного класса.

Отчет

После выполнения работы составляется отчет, который содержит:

- титульный лист,
- описание исходных требований,
- комментарии исходного кода и полученный результат для Doxygen.

Контрольные вопросы

1. Возможности и назначение Doxygen?
2. Правила комментирования кода.
3. Настройки **Doxygen.**

Источники

- Система документирования исходных кодов Doxygen:
    - Дистрибутив выдается преподавателем или скачивается из интернета.
    - Основной материал - Краткое руководство по правилам комментирования http://www.devexp.ru/2010/02/ispolzovanie-doxygen-dlya-dokumentirovaniya-koda/
    - Основной материал - Краткое руководство по использованию генератора документации http://microsin.ru/content/view/1218/1/
    - Дополнительно - Подробное руководство по комментированию http://doxygenorg.ru/old/4