

Automatyczne tworzenie dokumentacji

Seweryn Kowalski

doxygen

- Autor projektu: Dimitri van Heesch
- Strona projektu: <http://www.doxygen.org>
- Doxygen jest systemem dokumentowania oprogramowania pisanego w C++, C, Java, Objective-C, Python, IDL, Fortran, VHDL, PHP, C# oraz D. System ten może generować dokumentację w HTML oraz źródła dla LATEX'a. Dokumentacja może mieć format RTF (MS-Word), PostScript, PDF (z hyper-łącznikami), skompresowanego HTML oraz stron podręcznika w systemie UNIX dla programu man.
- Dokumentacja może być generowana bezpośrednio ze źródeł, jak też plików stowarzyszonych z nimi.
- Doxygen jest programem tworzonym dla systemu Linux i Mac OS w oparciu o bibliotekę Qt (doxywizard). Oprogramowanie to jest przenośne. Może być uruchomione pod systemem UNIX i jego klonami. Dostępne jest również w postaci binarnej dla systemu MS Windows.

Projekty dokumentowane z użyciem doxygen

- Adobe Open Source – strona domowa projektów ASL (Adobe Source Libraries).
- KDE - dokumentacja różnych bibliotek.
- Samba
- OSCAR (<http://www.oscar-net.org>) tworzy ogólną architekturę dla systemów robotycznych z ukierunkowaniem na roboty mobilne.
- XEngine (<http://xengine.sourceforge.net>) “silnik” dla wizualizacji 3D w czasie rzeczywistym, jest on niezależny od platformy i typu API wykorzystywanego renderowania.
- Xerces C++ Parser – biblioteka umożliwiająca pisanie aplikacji, które wymagają parsera plików XML.

Przykładowe systemy tworzenia dokumentacji

Niekomercyjne:

- AutoDOC
- HappyDoc
- C2HTML
- HyperSQL
- DocClass
- PHPDoc
- Cocoon
- Doc++
- CppDoc
- Javadoc
- gtk-doc
- cxxwrap
- CcDoc
- Epydoc
- KDoc
- Tydoc
- Cxref
- Autoduck
- RoboDoc
- ScanDoc
- Synopsis
- Perceps
- VBDOX
- Natural Docs
- phpDocumentor
- HeaderDoc
- Cxx2HTML
- ReThree-C++
- HTMLgen

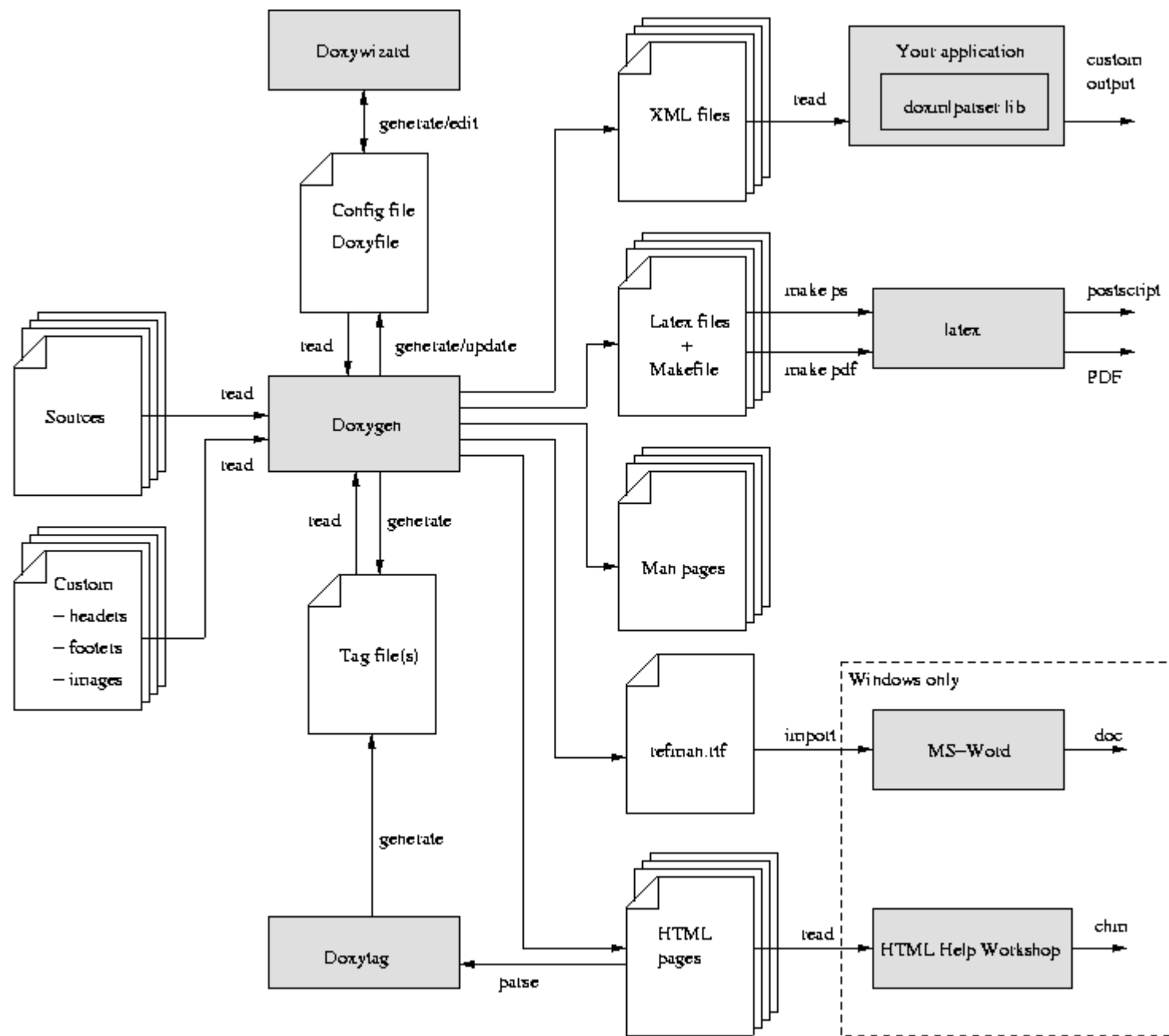
Komercyjne:

- CC-Rider
- DocJet
- Doc-o-matic
- ObjectManual
- Together
- DocBuilder
- VBXC

Doxygen – komponenty

- doxygen - program generujący dokumentację na podstawie utworzonego wcześniej pliku konfiguracyjnego i przeglądanych plików (źródeł programów i nie tylko).
- doxytag - program pomocniczy pozwalający integrować zewnętrzną dokumentację (do której doxygen nie ma bezpośredniego dostępu) z dokumentacją tworzoną przez doxygen.
- doxywizard - graficzna aplikacja ułatwiająca tworzenie pliku konfiguracyjnego
- dla dokumentacji danego projektu.
- Komponenty stowarzyszone:
- graphviz - oprogramowanie wykorzystywane do tworzenia rysunków grafów. Strona projektu: <http://www.research.att.com/sw/tools/graphviz/>

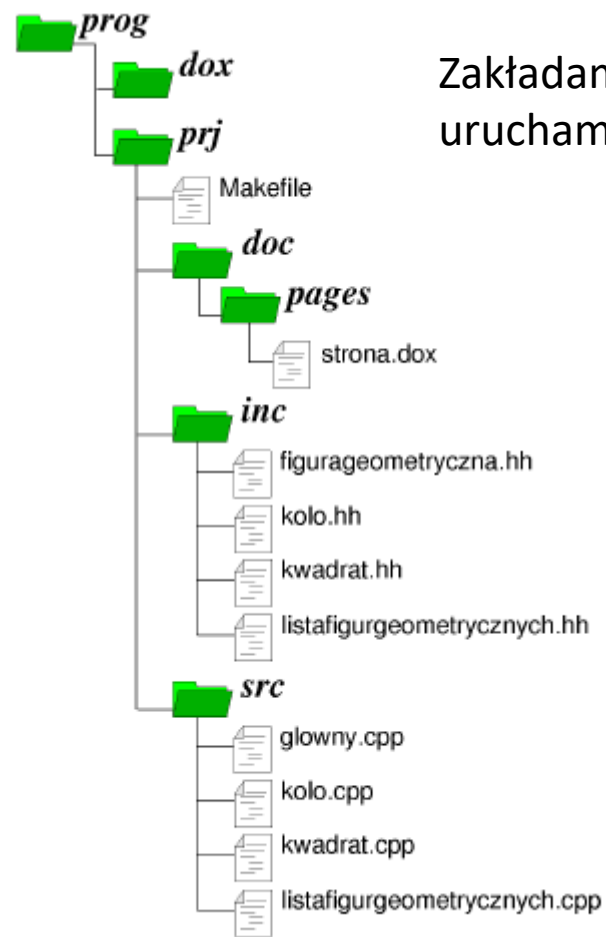
Przepływ informacji



Tworzenie dokumentacji

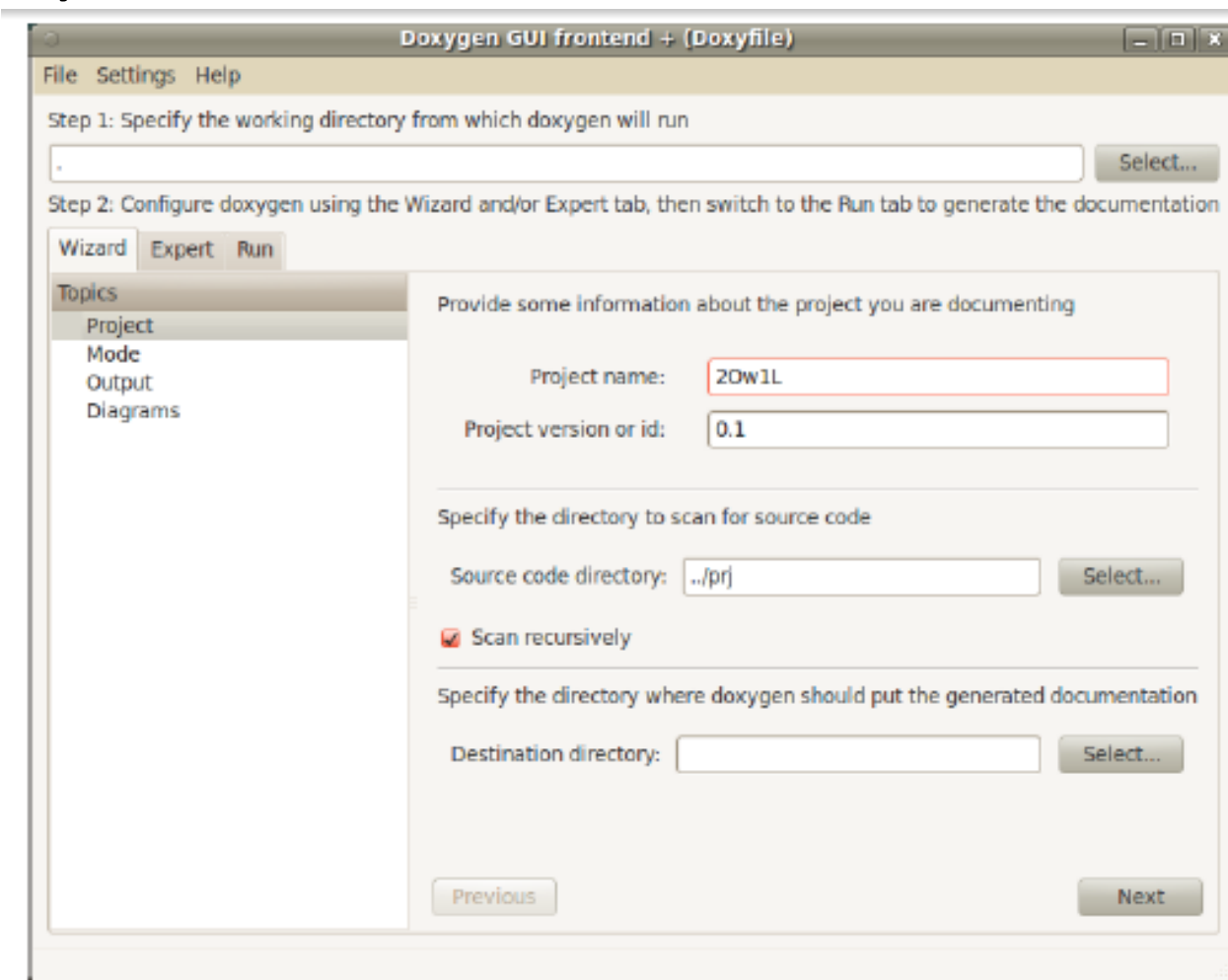
Konfiguracja generacji dokumentacji

- Katalogi i rozmieszczenie plików



Zakładamy, że w katalogu **dox** uruchamiamy aplikację **doxywizard**.

Konfiguracja projektu – okienko aplikacji doxywizard



The screenshot shows the 'Doxygen GUI frontend + (Doxyfile)' window. It has a menu bar with 'File', 'Settings', and 'Help'. The main area is divided into two steps:

Step 1: Specify the working directory from which doxygen will run. There is a text input field and a 'Select...' button.

Step 2: Configure doxygen using the Wizard and/or Expert tab, then switch to the Run tab to generate the documentation. Below this are three tabs: 'Wizard' (selected), 'Expert', and 'Run'.

Under the 'Wizard' tab, there is a 'Topics' list on the left with 'Project', 'Mode', 'Output', and 'Diagrams'. The 'Project' topic is selected.

The main configuration area for the 'Project' topic contains the following fields:

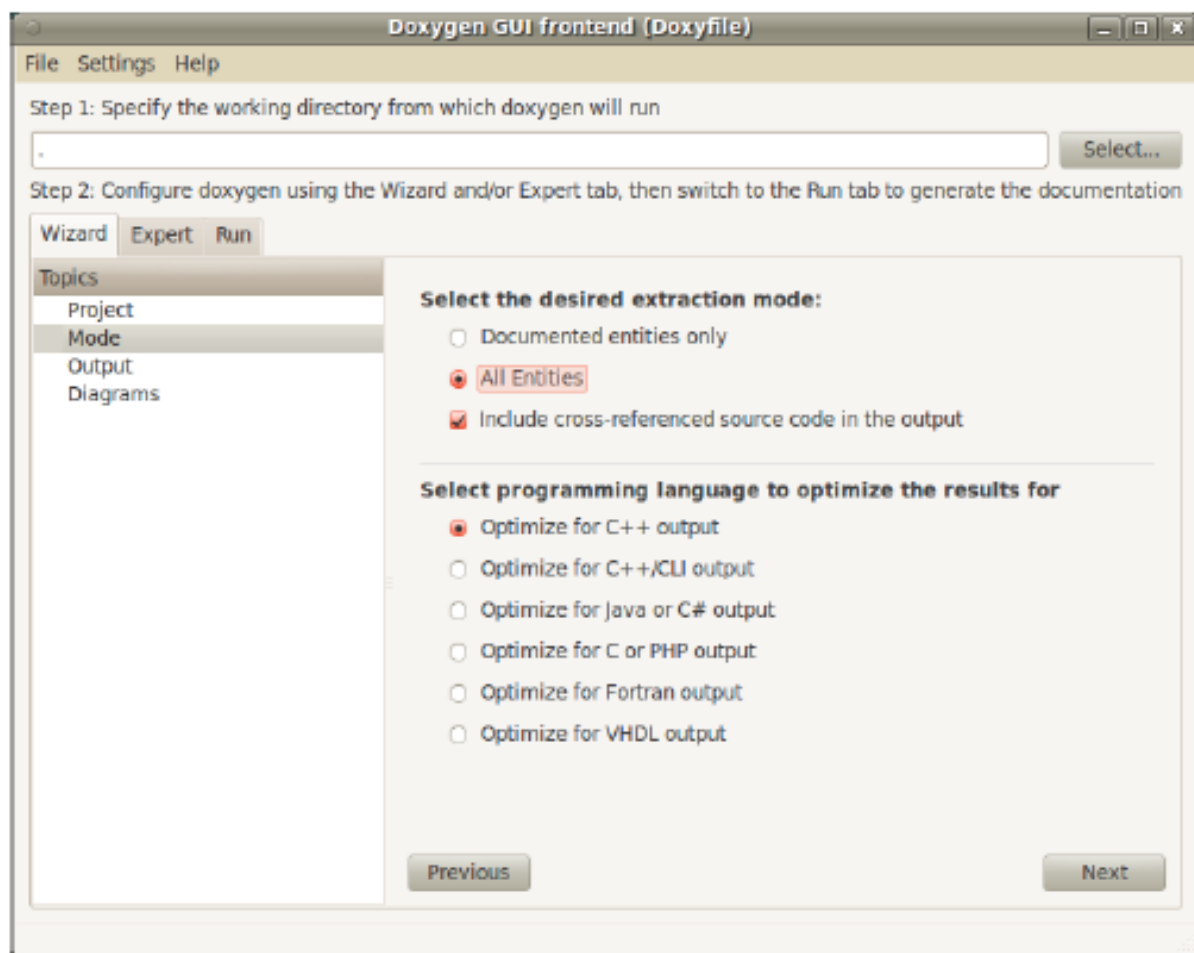
- Project name:** 20w1L
- Project version or id:** 0.1
- Specify the directory to scan for source code:**
 - Source code directory:** ../prj (with a 'Select...' button)
 - ☒ **Scan recursively**
- Specify the directory where doxygen should put the generated documentation:**
 - Destination directory:** (empty text field with a 'Select...' button)

At the bottom, there are 'Previous' and 'Next' buttons.

Ważniejsze ustawienia

Katalog roboczy: . (bieżący)
Katalog ze źródłami: ../prj
Przeglądanie katalogów: rekursywne

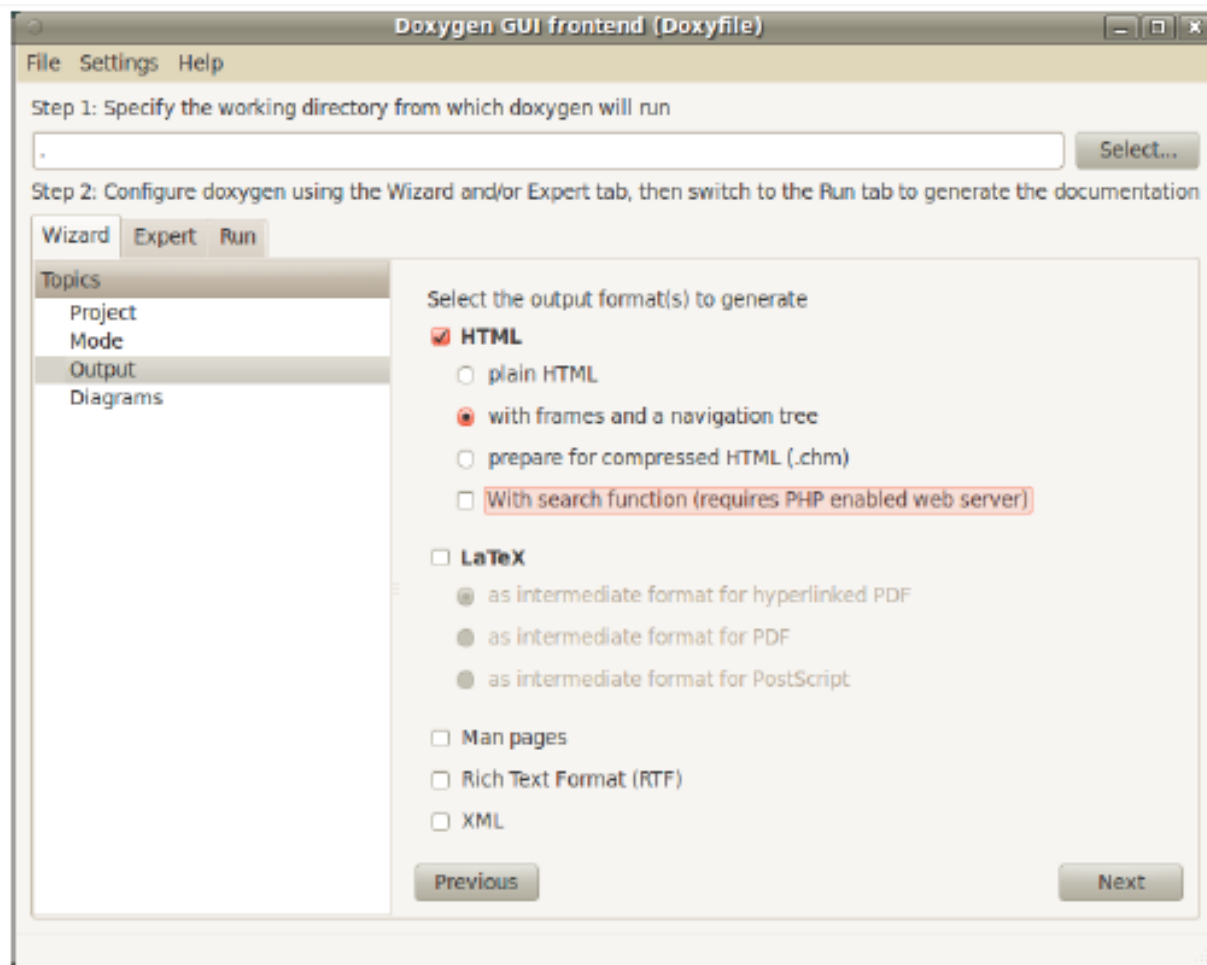
Konfiguracja trybu pracy



Ważniejsze trybu pracy

- Domyślnie w dokumentacji znajdą się tylko elementy, które są publiczne.
- *(All Entities)* W dokumentacji znajdą się również elementy, które nie są zdokumentowane.
- *(Include cross-referenced . . .)* W dokumentacji znajdzie się kod plików nagłówkowych wraz z odsyłaczami do definicji poszczególnych klas.

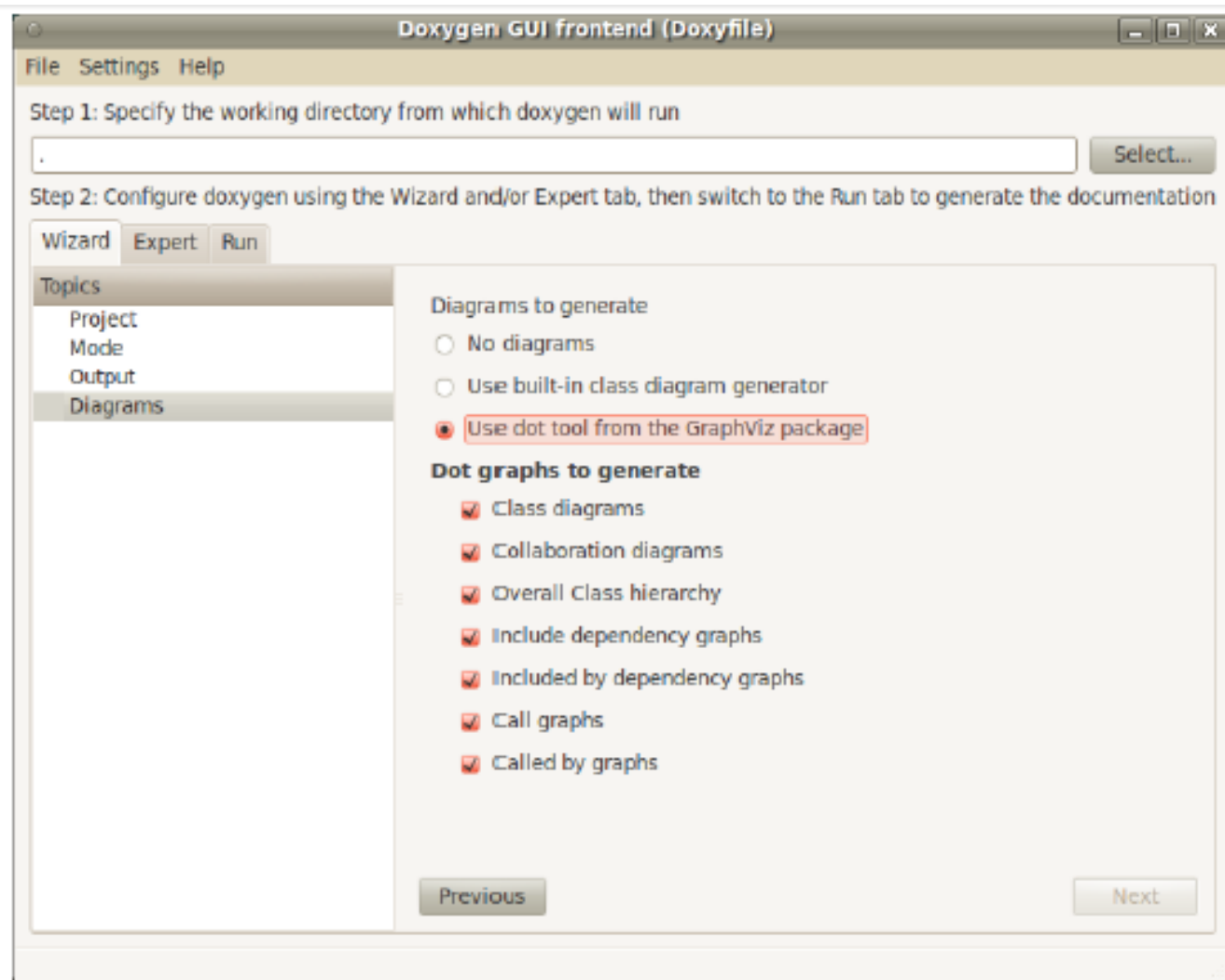
Konfiguracja formatu generowanej dokumentacji



Ważniejsze trybu pracy

- Domyślnie dokumentacja będzie generowana w podkatalogu `html` (zostanie on utworzony w trakcie pierwszego uruchomienia programu doxygen).
- (*with frames . . .*) Zostanie utworzone boczny rozwijalny indeks.

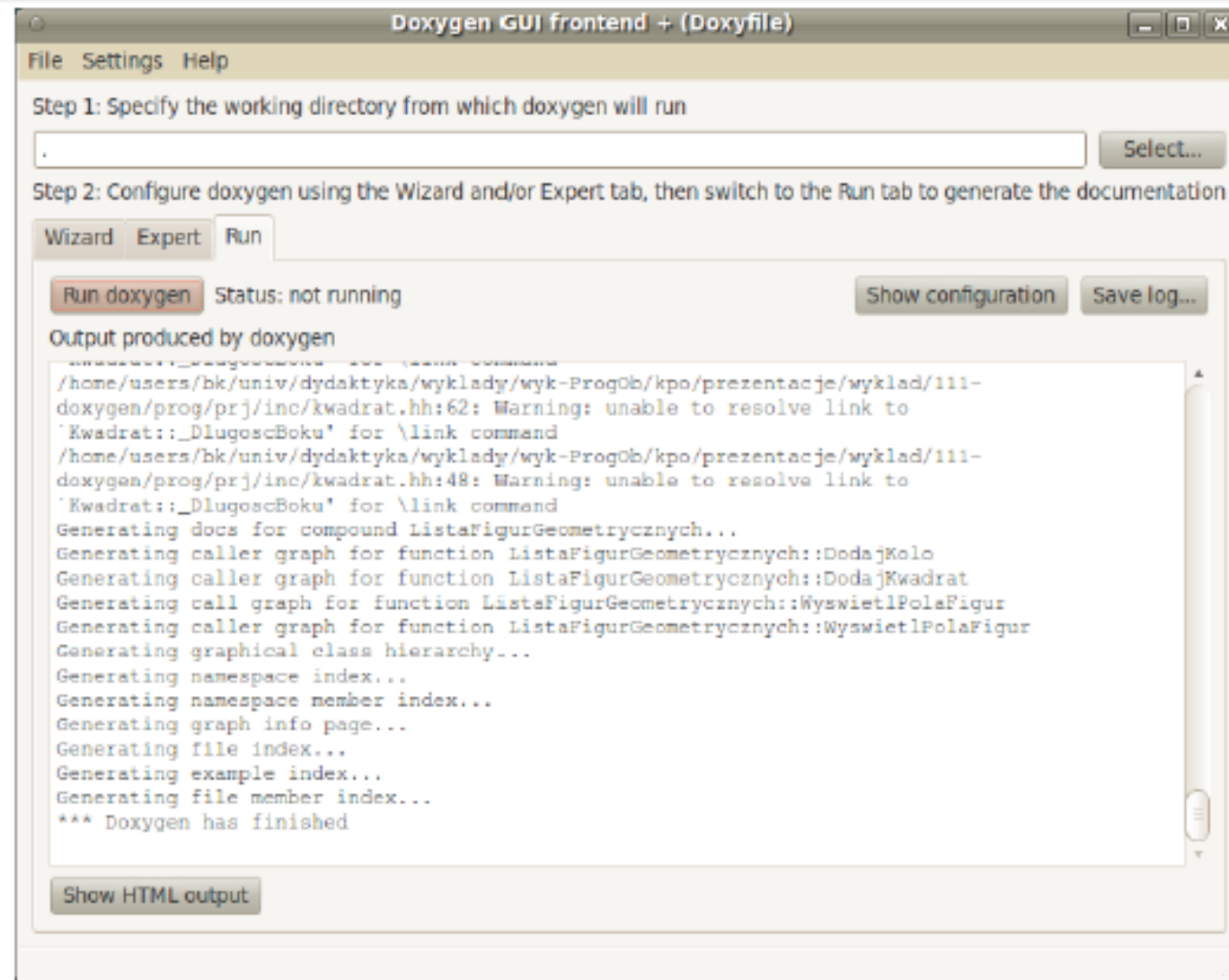
Konfiguracja generacji diagramów



Ważniejsze trybu pracy

- Jeśli mamy zainstalowany GraphViz dobrze jest zaznaczyć wszystkie rodzaje diagramów. Pozwoli to mieć pełną informację o strukturze programu i wzajemnych zależności między poszczególnymi metodami i funkcjami.

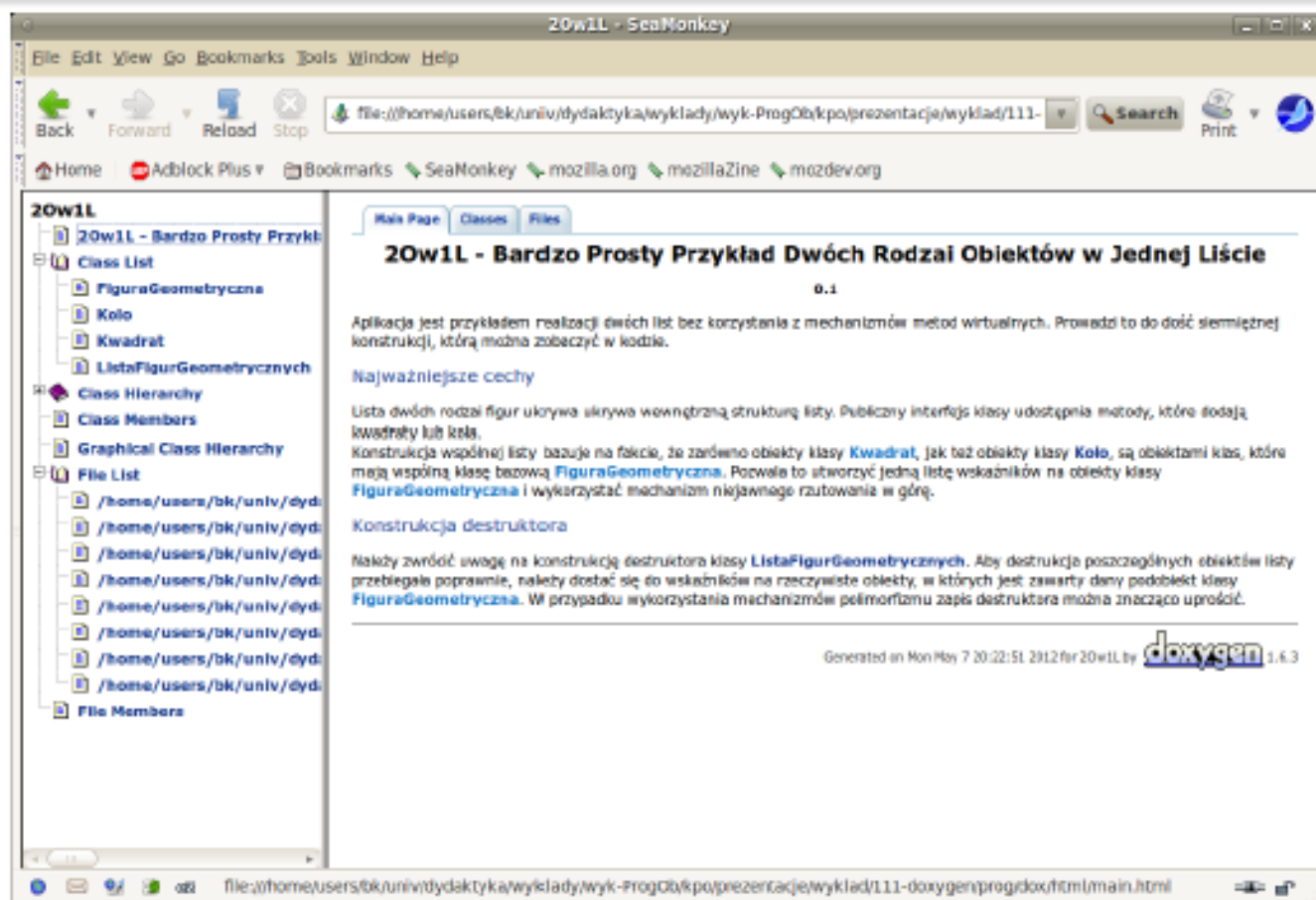
Generowanie dokumentacji



Faza generacji

- Gdy mamy wszystko ustawione, dobrze jest wcześniej zachować konfigurację do pliku, a następnie uruchomić fazę generacji dokumentacji.

Resultat



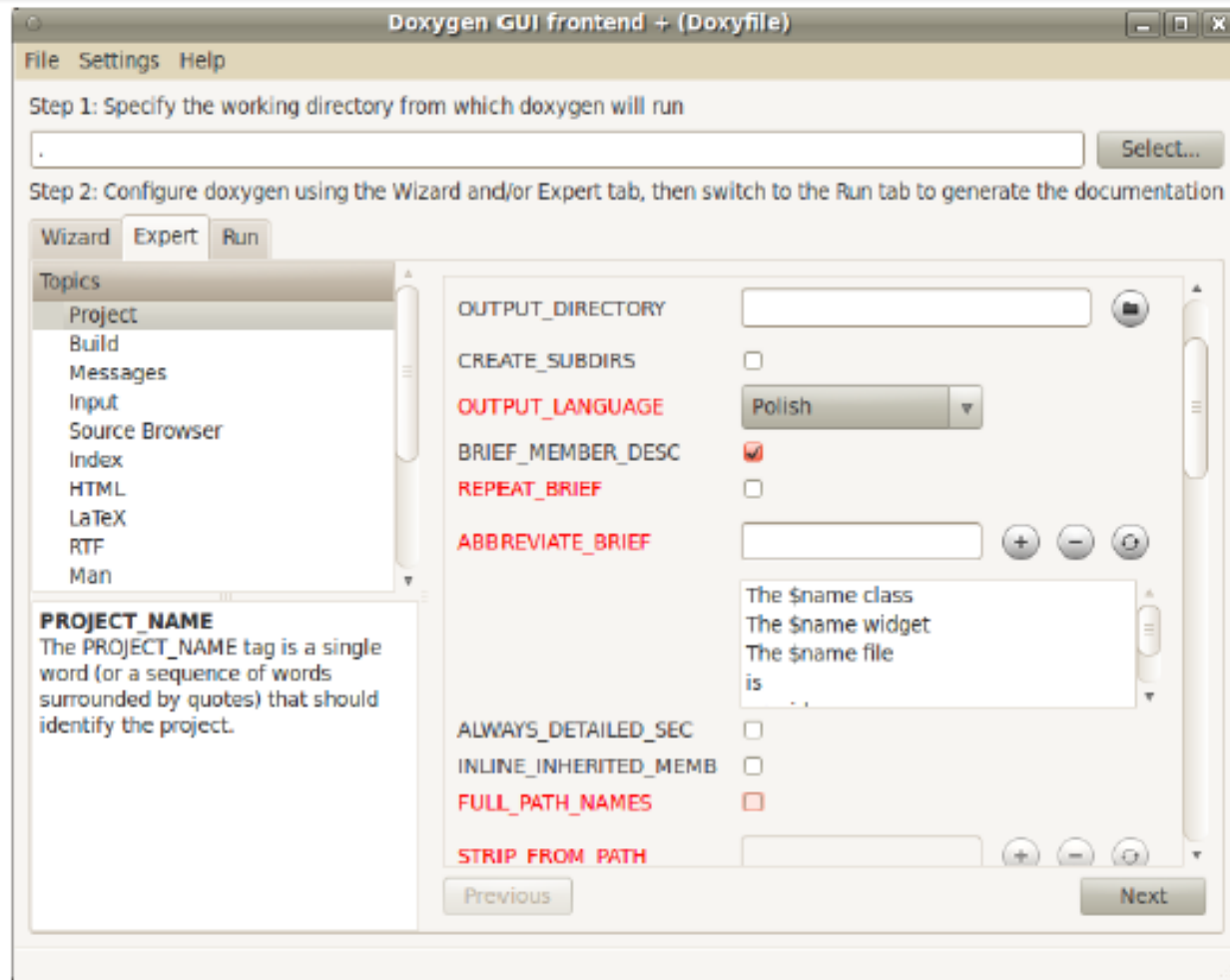
Cechy wygenerowanej dokumentacji

Korzystanie z uproszczonego trybu do zadawania ustawień konfiguracyjnych powoduje, że wygenerowana dokumentacja ma następujące wady:

- dodatkowe elementy opisu generowane przez doxygen są zawsze tylko w języku angielskim,
- nazwy plików są podawane z pełną ścieżką dostępu,
- dokumentacja nie jest generowana dla atrybutów klasy, które znajdują się w sekcji prywatnej. Jeżeli dokumentacja jest robiona dla użytkownika *zewnętrznego*, to jest to dobre rozwiązanie. Jednak jeśli robimy ją na własne potrzeby i chcemy mieć pełen zestaw informacji, to takie rozwiązanie jest niekorzystne.

Aby usunąć te wady należy . . .

Opcje rozszerzone – sekcja Project

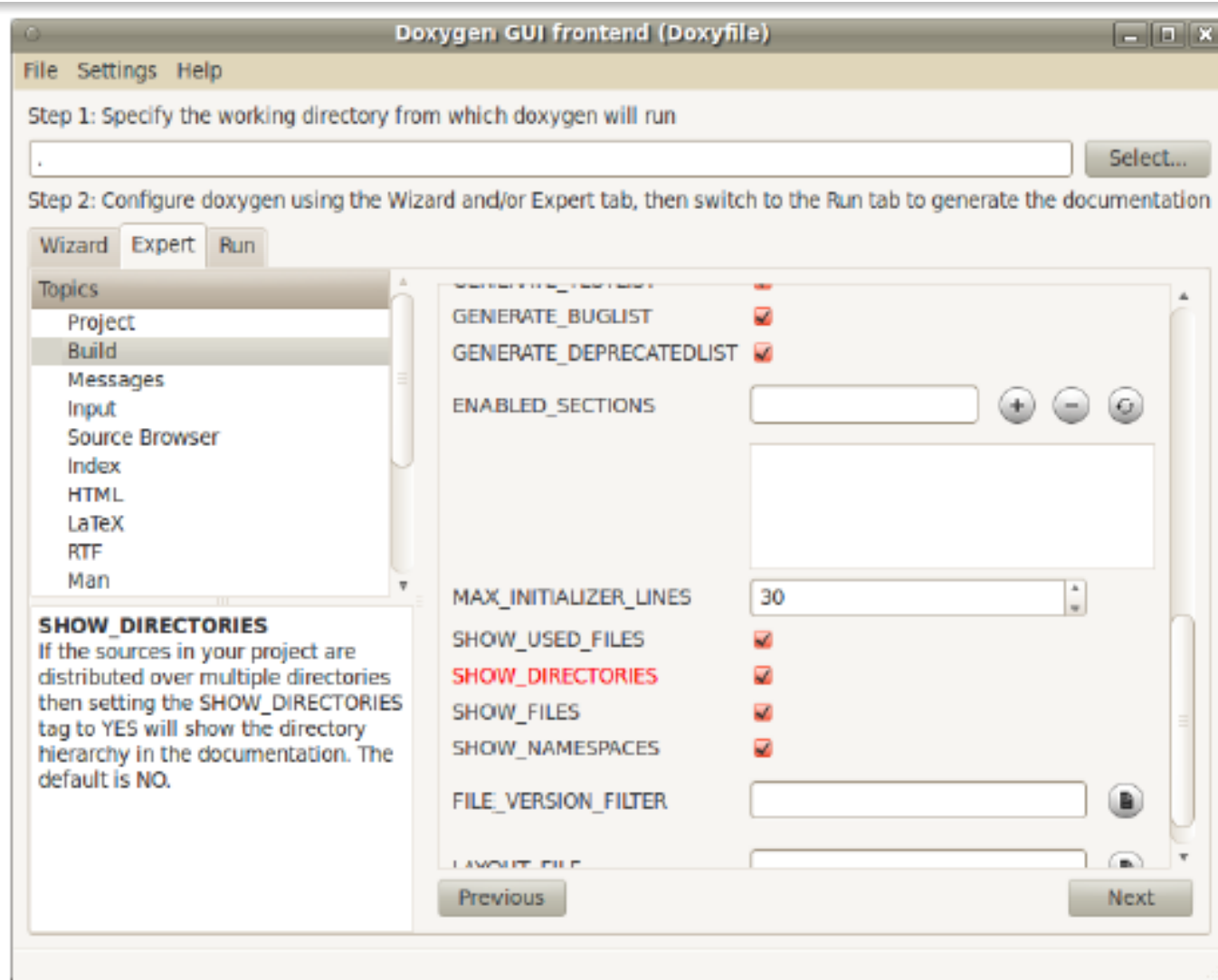


Ustawienia projektu

Należy przejść do zakładki **Expert** i wykonać następujące operacje:

- wybrać polską wersję językową.
- korzystnie jest *odznaczyć* ustawienie **REPEAT_BRIEF** (dzięki temu nie będzie powtarzany skrócony opis w pełnym opisie danego elementu),
- należy odznaczyć ustawienie **FULL_PATH_NAMES** (nazwy plików będą występowały bez pełnej ścieżki dostępu).

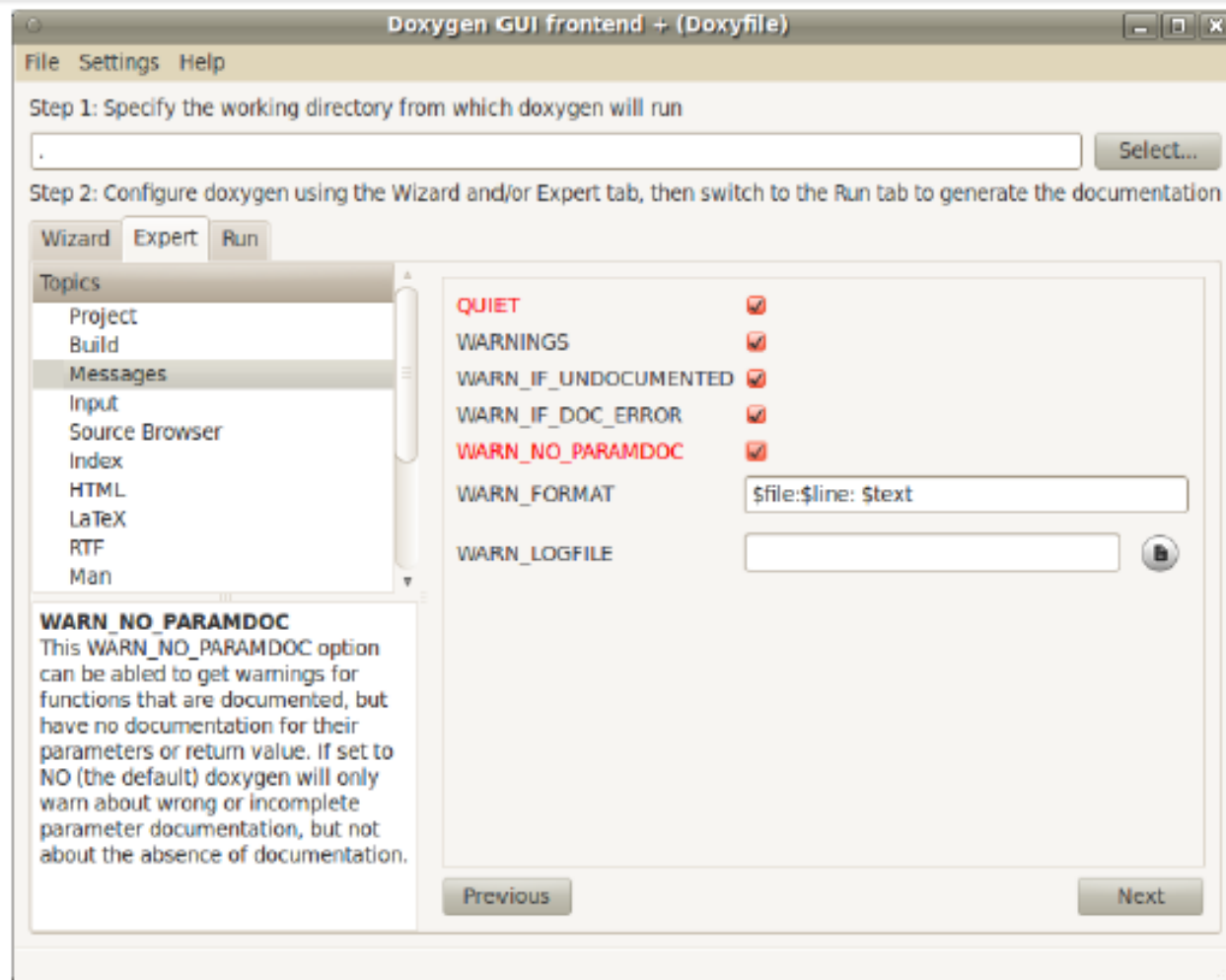
Opcje rozszerzone – sekcja Build



Ustawienia sposobu generacji dokumentu

- korzystnie jest zaznaczyć ustawienie `EXTRACT_PRIVATE`, dzięki temu pojawi się dokumentacja elementów prywatnych klasy.
- wskazane jest również zaznaczenie ustawienia `SHOW_DIRECTORIES`, pozwoli to mieć informację o strukturze katalogów, w których znajdują się dokumentowane pliki.

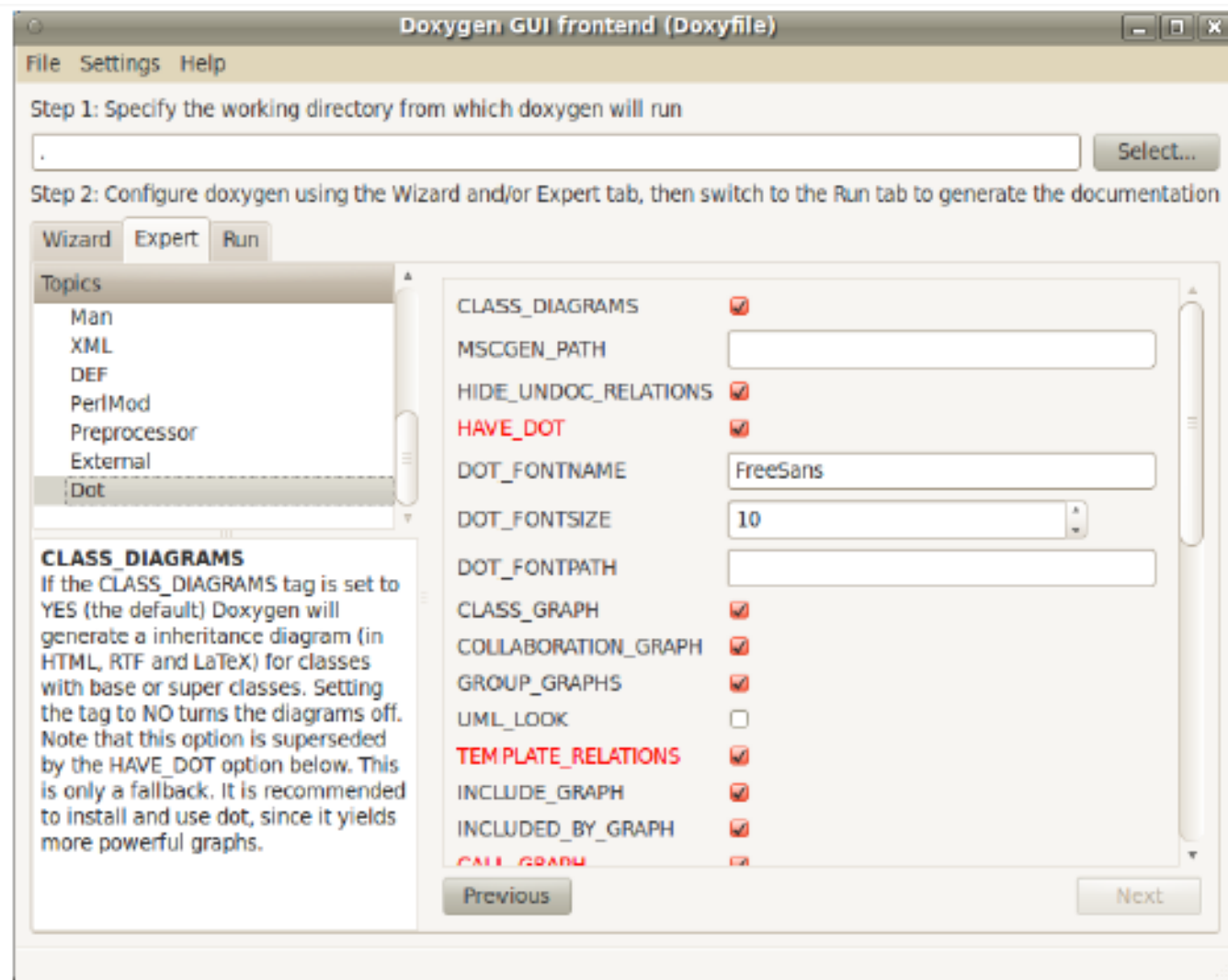
Opcje rozszerzone – sekcja Messages



Ustawienia komunikatów

- warto zaznaczyć ustawienie QUIET, dzięki temu w trakcie generacji dokumentacji nie pojawiają się komunikaty co jest tworzone (z punktu widzenia poprawności tworzonej dokumentacji zazwyczaj nie są one istotne),
- wskazane jest również zaznaczenie ustawienia WARN_NO_PARAMDOC, powoduje to generowanie ostrzeżeń, gdy jakieś parametry metody/funkcji lub wartość, którą zwraca, nie są zdokumentowane.

Opcje rozszerzone – sekcja Dot

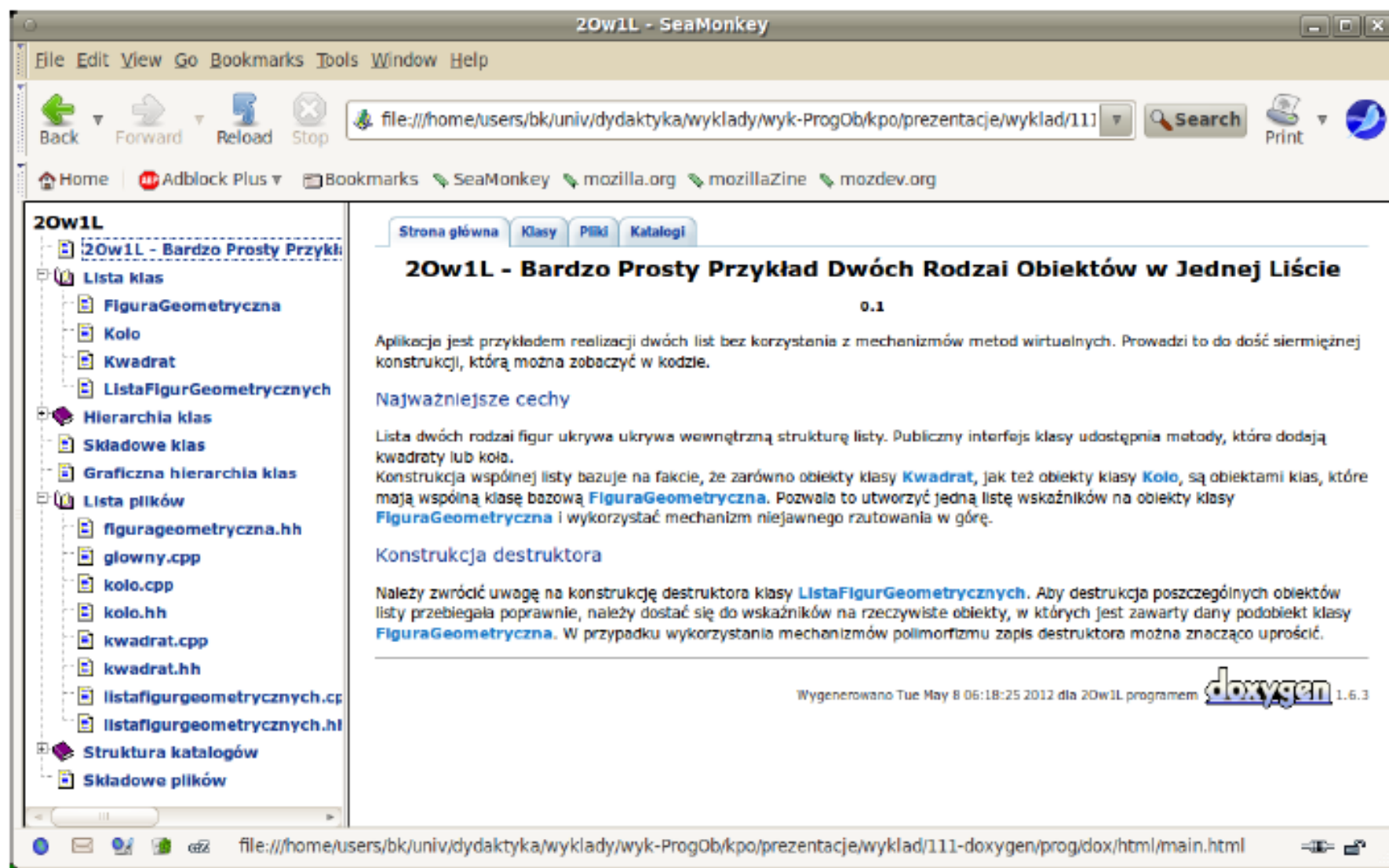


Ustawienia generacji diagramów

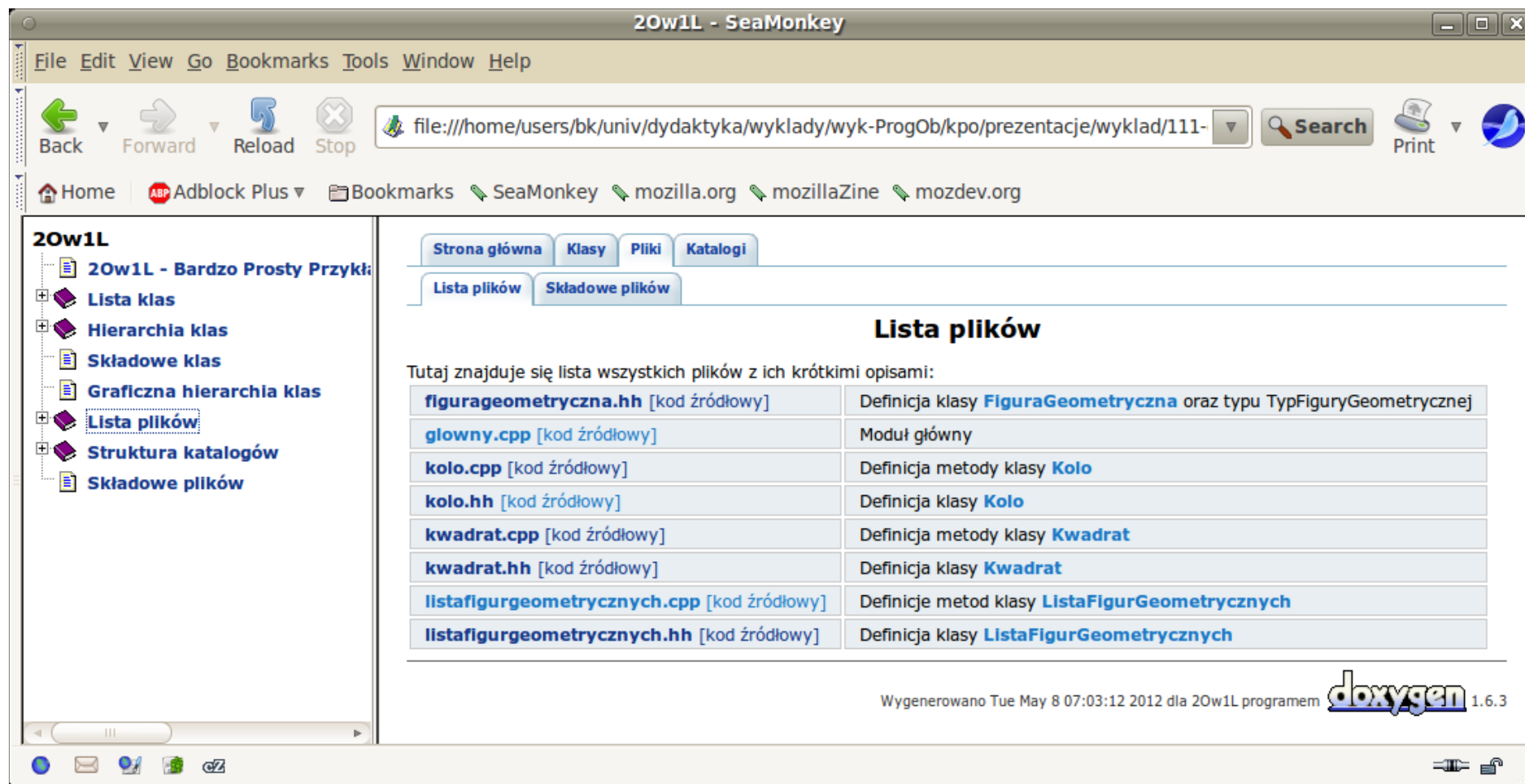
- dobrze jest zaznaczyć ustawienie **CLASS_DIAGRAM**, dzięki temu pozwoli to na umieszczenie w dokumentacji diagramów klas.

Uwaga: nie jest korzystne zaznaczanie ustawienia **UML_LOOK**. Powoduje to powstawanie bardzo szczegółowego diagramu, który staje się nieczytelny.

Strona tytułowa dokumentowanego systemu



Lista plików



The screenshot shows a web browser window titled "20w1L - SeaMonkey". The address bar displays the file path: `file:///home/users/bk/univ/dydaktyka/wyklady/wyk-ProgOb/kpo/prezentacje/wyklad/111-`. The browser's menu bar includes File, Edit, View, Go, Bookmarks, Tools, Window, and Help. The toolbar contains navigation buttons (Back, Forward, Reload, Stop), a search button, and a print button. The bookmarks bar shows links to Home, Adblock Plus, Bookmarks, SeaMonkey, mozilla.org, mozillaZine, and mozdev.org.

The main content area is divided into two panes. The left pane, titled "20w1L", contains a tree view with the following items: "20w1L - Bardzo Prosty Przykład", "Lista klas", "Hierarchia klas", "Składowe klas", "Graficzna hierarchia klas", "Lista plików" (highlighted), "Struktura katalogów", and "Składowe plików". The right pane, titled "Lista plików", contains a table of files and their descriptions.

Lista plików

Tutaj znajduje się lista wszystkich plików z ich krótkimi opisami:

figurageometryczna.hh [kod źródłowy]	Definicja klasy FiguraGeometryczna oraz typu TypFiguryGeometrycznej
glowny.cpp [kod źródłowy]	Moduł główny
kolo.cpp [kod źródłowy]	Definicja metody klasy Kolo
kolo.hh [kod źródłowy]	Definicja klasy Kolo
kwadrat.cpp [kod źródłowy]	Definicja metody klasy Kwadrat
kwadrat.hh [kod źródłowy]	Definicja klasy Kwadrat
listafiguregeometrycznych.cpp [kod źródłowy]	Definicje metod klasy ListaFigurGeometrycznych
listafiguregeometrycznych.hh [kod źródłowy]	Definicja klasy ListaFigurGeometrycznych

Wygenerowano Tue May 8 07:03:12 2012 dla 20w1L programem [doxygen](#) 1.6.3

Opis pliku kwadrat.hh dla programu doxygen

```
#ifndef KWADRAT_HH
#define KWADRAT_HH

/*!
 * \file
 * \brief Definicja klasy Kwadrat
 *
 * Plik zawiera definicję klasy Kwadrat, która
 * jest klasą pochodną i jest ona specjalizacją
 * klasy FiguraGeometryczna
 */

#include "figurageometryczna.hh"

...

#endif
```

Opis pliku: kwadrat.hh

The screenshot shows a web browser window titled "20w1L - SeaMonkey". The address bar displays the file path: `file:///home/users/bk/univ/dydaktyka/wyklady/wyk-ProgOb/kpo/prezentacje/wyklad/111`. The browser's toolbar includes navigation buttons (Back, Forward, Reload, Stop), a search bar, and a print button. The page content is organized into a sidebar and a main area.

Sidebar (Left):

- 20w1L
 - 20w1L - Bardzo Prosty Przykład
 - Lista klas
 - Hierarchia klas
 - Składowe klas
 - Graficzna hierarchia klas
 - Lista plików
 - figurageometryczna.hh
 - glowny.cpp
 - koło.cpp
 - koło.hh
 - kwadrat.cpp
 - kwadrat.hh
 - listafigurageometrycznych.cpp
 - listafigurageometrycznych.hh
 - Struktura katalogów
 - Składowe plików

Main Area (Right):

Strona główna Klasy Pliki Katalogi

Lista plików Składowe plików

prj » Inc

Dokumentacja pliku kwadrat.hh

Definicja klasy **Kwadrat**. Więcej...

```
#include "figurageometryczna.hh"
```

Wykres zależności załączania dla kwadrat.hh:

```
graph TD; kwadrat.hh --> figurageometryczna.hh; kwadrat.cpp --> kwadrat.hh;
```

Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:

Idź do kodu źródłowego tego pliku.

Komponenty

```
class Kwadrat
```

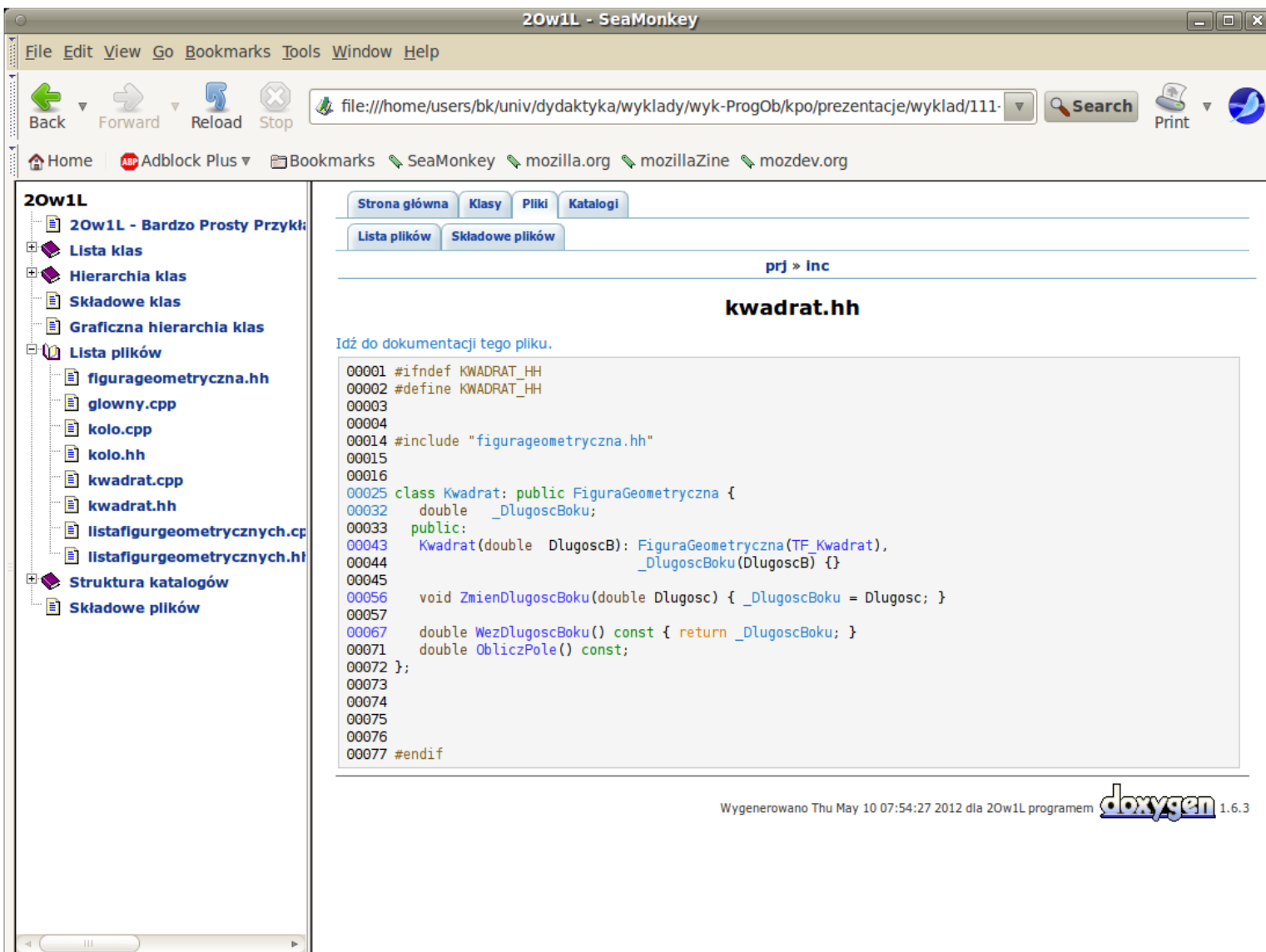
Modeluje pojęcie kwadratu. Więcej...

Opis szczegółowy

Plik zawiera definicję klasy **Kwadrat**, która jest klasą pochodną i jest ona specjalizacją klasy **FiguraGeometryczna**

Definicja w pliku **kwadrat.hh**.

Kod pliku (w dokumentacji): kwadrat.hh



The screenshot shows the SeaMonkey web browser displaying the documentation for the file `kwadrat.hh` in the `2Ow1L` project. The browser's address bar shows the file path: `file:///home/users/bk/univ/dydaktyka/wyklady/wyk-ProgOb/kpo/prezentacje/wyklad/111-`. The left sidebar shows the project structure, including a list of files where `kwadrat.hh` is highlighted. The main content area displays the C++ code for `kwadrat.hh`, which defines a `Kwadrat` class inheriting from `FiguraGeometryczna`. The code includes preprocessor directives, an include statement, class member declarations, and implementations for `ZmienDlugoscBoku`, `WezDlugoscBoku`, and `ObliczPole`. The bottom of the page indicates the documentation was generated by Doxygen 1.6.3 on May 10, 2012.

2Ow1L - SeaMonkey

File Edit View Go Bookmarks Tools Window Help

Back Forward Reload Stop `file:///home/users/bk/univ/dydaktyka/wyklady/wyk-ProgOb/kpo/prezentacje/wyklad/111-` Search Print

Home Adblock Plus Bookmarks SeaMonkey mozilla.org mozillaZine mozdev.org

2Ow1L

- 2Ow1L - Bardzo Prosty Przykład
- Lista klas
- Hierarchia klas
- Składowe klas
- Graficzna hierarchia klas
- Lista plików
 - figurageometryczna.hh
 - glowny.cpp
 - koło.cpp
 - koło.hh
 - kwadrat.cpp
 - kwadrat.hh
 - listafigurageometrycznych.cpp
 - listafigurageometrycznych.hh
- Struktura katalogów
- Składowe plików

Strona główna Klasy Pliki Katalogi

Lista plików Składowe plików

prj > inc

kwadrat.hh

[Idź do dokumentacji tego pliku.](#)

```
00001 #ifndef KWADRAT_HH
00002 #define KWADRAT_HH
00003
00004
00014 #include "figurageometryczna.hh"
00015
00016
00025 class Kwadrat: public FiguraGeometryczna {
00032     double _DlugoscBoku;
00033 public:
00043     Kwadrat(double DlugoscB): FiguraGeometryczna(TF_Kwadrat),
00044                                     _DlugoscBoku(DlugoscB) {}
00045
00056     void ZmienDlugoscBoku(double Dlugosc) { _DlugoscBoku = Dlugosc; }
00057
00067     double WezDlugoscBoku() const { return _DlugoscBoku; }
00071     double ObliczPole() const;
00072 };
00073
00074
00075
00076
00077 #endif
```

Wygenerowano Thu May 10 07:54:27 2012 dla 2Ow1L programem **doxygen** 1.6.3

Opis szczegółowy przed definicja metody w kwadrat.cpp

```
#include "kwadrat.hh"

/*!
 * \file
 * \brief Definicja metody klasy Kwadrat
 *
 * Zawiera definicję metod klasy Kwadrat.
 */

/*!
 * Wyznacza pole powierzchni danego kwadratu
 * i zwraca jego wartość.
 *
 * \return Zwraca pole powierzchni kwadratu. Jest ono wyrażone
 *         w jednostkach niemianowanych.
 */
double Kwadrat::ObliczPole() const
{
    return _DlugoscBoku*_DlugoscBoku;
}
```

Dokumentacja pliku kwadrat.cpp

2Ow1L - SeaMonkey

File Edit View Go Bookmarks Tools Window Help

Back Forward Reload Stop

file:///home/users/bk/univ/dydaktyka/wyklady/wyk-ProgOb/kpo/prezentacje/wyklad/111- Search Print

Home Adblock Plus Bookmarks SeaMonkey mozilla.org mozillaZine mozdev.org

2Ow1L

Strona główna Klasy Pliki Katalogi

Lista plików Składowe plików

prj » src

Dokumentacja pliku kwadrat.cpp

Definicja metody klasy [Kwadrat](#). Więcej...

#include "kwadrat.hh"

Wykres zależności załączania dla kwadrat.cpp:

```
graph TD
    kwadrat.cpp --> kwadrat.hh
    kwadrat.hh --> figurageometryczna.hh
```

[Idź do kodu źródłowego tego pliku.](#)

Opis szczegółowy

Zawiera definicję metod klasy [Kwadrat](#).

Definicja w pliku [kwadrat.cpp](#).

Wygenerowano Thu May 10 09:10:10 2012 dla 2Ow1L programem **doxygen** 1.6.3

2Ow1L - SeaMonkey

File Edit View Go Bookmarks Tools Window Help

Back Forward Reload Stop

file:///home/users/bk/univ/dydaktyka/wyklady/wyk-ProgOb/kpo/prezentacje/wyklad/111- Search Print

Home Adblock Plus Bookmarks SeaMonkey mozilla.org mozillaZine mozdev.org

2Ow1L

Strona główna Klasy Pliki Katalogi

Lista plików Składowe plików

prj » src

kwadrat.cpp

[Idź do dokumentacji tego pliku.](#)

```
00001 #include "kwadrat.hh"
00002
00018 double Kwadrat::ObliczPole() const
00019 {
00020     return _DlugoscBoku*_DlugoscBoku;
00021 }
```

Wygenerowano Thu May 10 09:10:10 2012 dla 2Ow1L programem **doxygen** 1.6.3

Komentowanie kodu

- Komentowanie kodu w sposób zgodny ze standardem rozumianym przez Doxygen'a w niewielkim stopniu różni się od standardowego, stosowanego przez programistów sposobu komentowania kodu w języku C++.

/*!	/**	///	///!
* ... komentarz ...	* ... komentarz ...	/// ... komentarz ...	///! ... komentarz ...
*/	*/	///	///!

Doxygen wykorzystuje dwa rodzaje opisu fragmentu kodu:

- Opis krótki
- Opis szczegółowy

```
/*!  
 * \brief Opis krótki zaczyna się tutaj.  
 *      Dalszy ciąg opisu krótkiego.  
 *  
 * Opis szczegółowy zaczyna się tutaj (po jednej pustej  
linii).  
 */
```

- Komentarz do fragmentu kodu można umieszczać w dwóch miejscach względem komentowanego kodu:
 - Przed komentowanym kodem
 - Za komentowanym kodem
- Aby Doxygen zrozumiał komentarz umieszczony za komentowanym kodem należy użyć znaku mniejszości przed komentarzem, tak jak w przykładzie poniżej:

```
int zmienna; /*!< To jest krótki opis zmiennej */
```

UWAGA: Umieszczanie komentarzy za komentowanym kodem jest dopuszczalne tylko w przypadku komentowania zmiennych i parametrów! Nie można w ten sposób komentować plików, klas, unii, struktur, grup, przestrzeni nazw, enums.

- W przypadku funkcji, oprócz komentarzy krótkich (ogólnych) i szczegółowych, warto również poświęcić kilka minut na udokumentowanie argumentów wejściowych i wyjściowych oraz zwracanych wartości. Poniżej przykład poprawnie udokumentowanej funkcji z wykorzystaniem komend specjalnych (\param oraz \return):

```
/**
 * Funkcja sprawdza, czy z trzech odcinków da się zbudować trójkąt.
 * Pobiera trzy wartości typu int i zwraca wartość typu bool.
 *
 * \param[in] x długość pierwszego odcinka.
 * \param[in] y długość drugiego odcinka.
 * \param[in] z długość trzeciego odcinka.
 * \return 1 jeśli da się zbudować trójkąt, 0 jeśli nie da się zbudować trójkąta.
 */
bool triangle(int x, int y, int z)
{
    return (x < y+z) && (y < x+z) && (z < x+y);
}
```

- Atrybut [in] umieszczony po komendzie \param jest atrybutem opcjonalnym, wskazującym że komentowany argument jest argumentem wejściowym funkcji (dostarcza danych do funkcji).
- Jeśli na liście argumentów znajduje się argument przekazywany przez referencję lub za pomocą wskaźników (jest pobierany i zmieniany w trakcie działania funkcji), należy użyć atrybutu [in,out].
- Jeśli argument nie wprowadza żadnych danych do funkcji, a jedynie funkcja zwraca wartość za pomocą argumentu, należy użyć atrybutu [out].

Doxygen interpretuje komendy takie jak:

- `\author` jeden lub lista kilku autorów danego fragmentu kodu
- `\version` wersja komentowanego fragmentu kodu
- `\date` data lub przedział czasowy tworzenia fragmentu kodu
- `\bug` opis nie rozwiązanego w bieżącej wersji błędu
- `\warning` ostrzeżenie przed szkodliwym działaniem tego fragmentu kodu

