

Decyzje

1.0

Generated by Doxygen 1.9.1

1 Class Index	1
1.1 Class List	1
2 File Index	3
2.1 File List	3
3 Class Documentation	5
3.1 decyzja Struct Reference	5
3.1.1 Detailed Description	5
3.1.2 Member Data Documentation	5
3.1.2.1 d_atrybut	6
3.1.2.2 d_id	6
3.1.2.3 d_nie	6
3.1.2.4 d_tak	6
3.1.2.5 d_wartosc	6
3.1.2.6 d_znak	6
4 File Documentation	7
4.1 main.cpp File Reference	7
4.2 model.cpp File Reference	7
4.2.1 Function Documentation	8
4.2.1.1 czy_liczba()	8
4.2.1.2 czy_liczba_naturalna()	8
4.2.1.3 dodawanie_do_mapy()	9
4.2.1.4 dzielenie_warunku()	9
4.2.1.5 generowanie_pliku_wyjscia()	10
4.2.1.6 postepowanie_drzewa()	10
4.2.1.7 przypisywanie_do_drzewa()	11
4.2.1.8 przypisywanie_do_warunku()	11
4.2.1.9 warunek_galezi()	11
4.3 model.h File Reference	12
4.3.1 Function Documentation	13
4.3.1.1 czy_liczba()	13
4.3.1.2 czy_liczba_naturalna()	13
4.3.1.3 dodawanie_do_mapy()	13
4.3.1.4 dzielenie_warunku()	14
4.3.1.5 generowanie_pliku_wyjscia()	14
4.3.1.6 postepowanie_drzewa()	15
4.3.1.7 przypisywanie_do_drzewa()	15
4.3.1.8 przypisywanie_do_warunku()	16
4.3.1.9 warunek_galezi()	16
4.4 struktury.h File Reference	16
4.5 widok.cpp File Reference	17

4.5.1 Function Documentation	17
4.5.1.1 czytanie_pliku_atrybutu()	17
4.5.1.2 czytanie_pliku_warunkowego()	17
4.5.1.3 poprawnosc_odczytu()	18
4.6 widok.h File Reference	18
4.6.1 Function Documentation	19
4.6.1.1 czytanie_pliku_atrybutu()	19
4.6.1.2 czytanie_pliku_warunkowego()	19
4.6.1.3 poprawnosc_odczytu()	20
Index	21

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

decyzja	5
-----------------------------------	---

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

main.cpp	7
model.cpp	7
model.h	12
struktury.h	16
widok.cpp	17
widok.h	18

Chapter 3

Class Documentation

3.1 decyzja Struct Reference

```
#include <struktury.h>
```

Public Attributes

- int [d_id](#)
- string [d_atrybut](#)
- string [d_znak](#)
- float [d_wartosc](#)
- string [d_tak](#)
- string [d_nie](#)

3.1.1 Detailed Description

Struktura określająca budowę gałęzi drzewa binarnego

Parameters

<i>d_id</i>	- numer gałęzi drzewa
<i>d_atrybut</i>	- atrybut warunku
<i>d_znak</i>	- znak warunku
<i>d_wartosc</i>	- wartosc warunku
<i>d_tak</i>	- indeks wyjścia tak
<i>d_nie</i>	- indeks wyjścia nie

3.1.2 Member Data Documentation

3.1.2.1 d_atrybut

```
string decyzja::d_atrybut
```

3.1.2.2 d_id

```
int decyzja::d_id
```

3.1.2.3 d_nie

```
string decyzja::d_nie
```

3.1.2.4 d_tak

```
string decyzja::d_tak
```

3.1.2.5 d_wartosc

```
float decyzja::d_wartosc
```

3.1.2.6 d_znak

```
string decyzja::d_znak
```

The documentation for this struct was generated from the following file:

- [struktury.h](#)

Chapter 4

File Documentation

4.1 main.cpp File Reference

```
#include <string>
#include <vector>
#include <fstream>
#include <iostream>
#include <unordered_map>
#include <locale.h>
#include "model.h"
#include "widok.h"
#include "struktury.h"
```

Include dependency graph for main.cpp:

4.2 model.cpp File Reference

```
#include <string>
#include <vector>
#include <fstream>
#include <unordered_map>
#include "model.h"
#include "widok.h"
#include "struktury.h"
```

Include dependency graph for model.cpp:

Functions

- bool [czy_liczba](#) (const string &wyraz)
Funkcja sprawdzająca czy ciąg wyrazów to liczba zmiennoprzecinkowa.
- bool [czy_liczba_naturalna](#) (const string &wyraz)
Procedura sprawdzająca czy ciąg znaków to liczba naturalna.
- void [przypisywanie_do_drzewa](#) (vector< [decyzja](#) > &drzewo, string *tablica, const int &linia)
Procedura wpisująca dane do drzewa.
- void [przypisywanie_do_warunku](#) (vector< vector< float >> &przypadki, vector< float > &wiersz)
Procedura wpisująca dane do wektora wektorów przypadków.

- void `dzielenie_warunku` (`decyzja` &obiekt, string &warunek)
Procedura zmieniająca ciąg znaków warunku na pola struktury.
- void `generowanie_pliku_wyjscia` (unordered_map< string, vector< vector< float >>> &mapa, ofstream &wyjscie)
Procedura generująca plik wyjściowy z mapy asocjacyjnej.
- bool `warunek_galezi` (int nr_przypadku, int nr_galezi, int nr_war_atrybutu, const vector< vector< float >> &przypadek, const vector< `decyzja` > &drzewo)
Funkcja sprawdzająca warunek gałęzi drzewa z atrybutem przypadku.
- void `dodawanie_do_mapy` (unordered_map< string, vector< vector< float >>> &mapa, const string &klucz, const vector< float > &przypadki)
Procedura wpisująca wektor przypadku do mapy asocjacyjnej.
- int `postepowanie_drzewa` (int nr_galezi, const int nr_przypadku, const vector< `decyzja` > &drzewo, const vector< string > &atrybut, const vector< vector< float >> &przypadek, unordered_map< string, vector< vector< float >>> &przyporzadkowane_przypadki)
Funkcja poruszania się programu po drzewie decyzyjnym.

4.2.1 Function Documentation

4.2.1.1 czy_liczba()

```
bool czy_liczba (
    const string & wyraz )
```

Funkcja sprawdzająca czy ciąg wyrazów to liczba zmiennoprzecinkowa.

Parameters

wyraz	- ciąg znaków, który ma zostać sprawdzony
-------	---

Returns

true - jeśli wyraz jest liczbą
false - jeśli wyraz nie jest liczbą

4.2.1.2 czy_liczba_naturalna()

```
bool czy_liczba_naturalna (
    const string & wyraz )
```

Procedura sprawdzająca czy ciąg znaków to liczba naturalna.

Parameters

wyraz	- ciąg znaków, który ma zostać sprawdzony
-------	---

Returns

true - jeśli wyraz jest liczbą
false - jeśli wyraz nie jest liczbą

4.2.1.3 dodawanie_do_mapy()

```
void dodawanie_do_mapy (
    unordered_map< string, vector< vector< float >>> & mapa,
    const string & klucz,
    const vector< float > & przypadki )
```

Procedura wpisująca wektor przypadku do mapy asocjacyjnej.

Parameters

<i>mapa</i>	- mapa asocjacyjna, do której wpisywane sa wektory przypadków
<i>klucz</i>	- klucz mapy asocjacyjnej
<i>przypadki</i>	- wektor wartości przypadku, który jest wpisywany do mapy

Returns

-

4.2.1.4 dzielenie_warunku()

```
void dzielenie_warunku (
    decyzja & obiekt,
    string & warunek )
```

Procedura zmieniająca ciąg znaków warunku na pola struktury.

Parameters

<i>obiekt</i>	- obiekt wektora struktur, do którego wpisywane są wartości
<i>warunek</i>	- ciąg znaków warunku, pobrane z pliku tekstowego

Returns

-

4.2.1.5 generowanie_pliku_wyjscia()

```
void generowanie_pliku_wyjscia (
    unordered_map< string, vector< vector< float >>> & mapa,
    ofstream & wyjście )
```

Procedura generująca plik wyjściowy z mapy asocjacyjnej.

Parameters

<i>mapa</i>	- mapa asocjacyjna, z której wpisywane są wektory przypadków
<i>wyjście</i>	- strumień wyjścia pliku tekstowego

Returns

-

4.2.1.6 postepowanie_drzewa()

```
int postepowanie_drzewa (
    int nr_galezi,
    const int nr_przypadku,
    const vector< decyzja > & drzewo,
    const vector< string > & atrybut,
    const vector< vector< float >> & przypadek,
    unordered_map< string, vector< vector< float >>> & przyporządkowane_przypadki )
```

Funkcja poruszania się programu po drzewie decyzyjnym.

Parameters

<i>nr_galezi</i>	- indeks gałęzi, której warunek jest rozpatrywany
<i>nr_przypadku</i>	- indeks wektora, sprawdzanego przypadku
<i>drzewo</i>	- wektor struktur zawierający drzewo decyzyjne
<i>atrybut</i>	- wektor atrybutów, z którym porównywany jest atrybut warunku gałęzi drzewa
<i>przypadek</i>	- wektor przechowujący wektor wartości atrybutów przypadku
<i>przyporządkowane_przypadki</i>	- mapa asocjacyjna zawierająca jako klucz odpowiednie etykiety, a jako wartość wektor wektorów przechowujący przypadki

Returns

0 - gdy brak błędów

13 - gdy podano nieprawidłowe dane

4.2.1.7 przypisywanie_do_drzewa()

```
void przypisywanie_do_drzewa (
    vector< decyzja > & drzewo,
    string * tablica,
    const int & linia )
```

Procedura wpisująca dane do drzewa.

Parameters

<i>drzewo</i>	- wektor struktur przechowujący <numer gałęzi> <warunek> <wartość jeśli tak> <wartość jeśli nie>
<i>*tablica</i>	- tablica pomocnicza do wpisywania danych do struktury
<i>linia</i>	- numer gałęzi

Returns

-

4.2.1.8 przypisywanie_do_warunku()

```
void przypisywanie_do_warunku (
    vector< vector< float >> & przypadki,
    vector< float > & wiersz )
```

Procedura wpisująca dane do wektora wektorów przypadków.

Parameters

<i>przypadki</i>	- wektor wektorów przechowujący wartości atrybutów przypadków
<i>wiersz</i>	- wektor pomocniczy do wpisywania danych do wektora wektorów

4.2.1.9 warunek_galezi()

```
bool warunek_galezi (
    int nr_przypadku,
    int nr_galezi,
    int nr_war_atrybutu,
    const vector< vector< float >> & przypadek,
    const vector< decyzja > & drzewo )
```

Funkcja sprawdzająca warunek gałęzi drzewa z atrybutem przypadku.

Parameters

<i>nr_przypadku</i>	- indeks wektora, sprawdzanego przypadku
---------------------	--

Parameters

<i>nr_galezi</i>	- indeks gałęzi, której warunek jest rozpatrywany
<i>przypadek</i>	- wektor przechowujący wektor wartości atrybutów przypadku
<i>drzewo</i>	- wektor struktur przechowujący warunki gałęzi

Returns

- true - jeśli warunek jest spełniony
- false - jeśli warunek nie jest spełniony

4.3 model.h File Reference

```
#include <string>
#include <vector>
#include <sstream>
#include <fstream>
#include <unordered_map>
#include "struktury.h"
#include "widok.h"
```

Include dependency graph for model.h: This graph shows which files directly or indirectly include this file:

Functions

- bool [czy_liczba](#) (const string &wyraz)
Funkcja sprawdzająca czy ciąg wyrazów to liczba zmiennoprzecinkowa.
- bool [czy_liczba_naturalna](#) (const string &wyraz)
Procedura sprawdzająca czy ciąg znaków to liczba naturalna.
- void [przypisywanie_do_drzewa](#) (vector< [decyzja](#) > &drzewo, string *tablica, const int &linia)
Procedura wpisująca dane do drzewa.
- void [przypisywanie_do_warunku](#) (vector< vector< float >> &przypadki, vector< float > &wiersz)
Procedura wpisująca dane do wektora wektorów przypadków.
- void [dzielenie_warunku](#) ([decyzja](#) &obiekt, string &warunek)
Procedura zmieniająca ciąg znaków warunku na pola struktury.
- void [generowanie_pliku_wyjscia](#) (unordered_map< string, vector< vector< float >>> &mapa, ofstream &wyjscie)
Procedura generująca plik wyjściowy z mapy asocjacyjnej.
- bool [warunek_galezi](#) (int nr_przypadku, int nr_galezi, int nr_war_atrybutu, const vector< vector< float >> &przypadek, const vector< [decyzja](#) > &drzewo)
Funkcja sprawdzająca warunek gałęzi drzewa z atrybutem przypadku.
- void [dodawanie_do_mapy](#) (unordered_map< string, vector< vector< float >>> &mapa, const string &klucz, const vector< float > &przypadki)
Procedura wpisująca wektor przypadku do mapy asocjacyjnej.
- int [postepowanie_drzewa](#) (int nr_galezi, const int nr_przypadku, const vector< [decyzja](#) > &drzewo, const vector< string > &atrybut, const vector< vector< float >> &przypadek, unordered_map< string, vector< vector< float >>> &przyporzadkowane_przypadki)
Funkcja poruszania się programu po drzewie decyzyjnym.

4.3.1 Function Documentation

4.3.1.1 czy_liczba()

```
bool czy_liczba (
    const string & wyraz )
```

Funkcja sprawdzająca czy ciąg wyrazów to liczba zmiennoprzecinkowa.

Parameters

wyraz	- ciąg znaków, który ma zostać sprawdzony
-------	---

Returns

true - jeśli wyraz jest liczbą
false - jeśli wyraz nie jest liczbą

4.3.1.2 czy_liczba_naturalna()

```
bool czy_liczba_naturalna (
    const string & wyraz )
```

Procedura sprawdzająca czy ciąg znaków to liczba naturalna.

Parameters

wyraz	- ciąg znaków, który ma zostać sprawdzony
-------	---

Returns

true - jeśli wyraz jest liczbą
false - jeśli wyraz nie jest liczbą

4.3.1.3 dodawanie_do_mapy()

```
void dodawanie_do_mapy (
    unordered_map< string, vector< vector< float >>> & mapa,
    const string & klucz,
    const vector< float > & przypadki )
```

Procedura wpisująca wektor przypadku do mapy asocjacyjnej.

Parameters

<i>mapa</i>	- mapa asocjacyjna, do której wpisywane sa wektory przypadków
<i>klucz</i>	- klucz mapy asocjacyjnej
<i>przypadki</i>	- wektor wartości przypadku, który jest wpisywany do mapy

Returns

-

4.3.1.4 dzielenie_warunku()

```
void dzielenie_warunku (
    decyzja & obiekt,
    string & warunek )
```

Procedura zmieniająca ciąg znaków warunku na pola struktury.

Parameters

<i>obiekt</i>	- obiekt wektora struktur, do którego wpisywane są wartości
<i>warunek</i>	- ciąg znaków warunku, pobrane z pliku tekstowego

Returns

-

4.3.1.5 generowanie_pliku_wyjscia()

```
void generowanie_pliku_wyjscia (
    unordered_map< string, vector< vector< float >>> & mapa,
    ofstream & wyjscie )
```

Procedura generująca plik wyjściowy z mapy asocjacyjnej.

Parameters

<i>mapa</i>	- mapa asocjacyjna, z której wpisywane sa wektory przypadków
<i>wyjscie</i>	- strumień wyjścia pliku tekstowego

Returns

-

4.3.1.6 `postepowanie_drzewa()`

```
int postepowanie_drzewa (
    int nr_galezi,
    const int nr_przypadku,
    const vector< decyzja > & drzewo,
    const vector< string > & atrybut,
    const vector< vector< float >> & przypadek,
    unordered_map< string, vector< vector< float >>> & przyporzadkowane_przypadki )
```

Funkcja poruszania się programu po drzewie decyzyjnym.

Parameters

<i>nr_galezi</i>	- indeks gałęzi, której warunek jest rozpatrywany
<i>nr_przypadku</i>	- indeks wektora, sprawdzanego przypadku
<i>drzewo</i>	- wektor struktur zawierający drzewo decyzyjne
<i>atrybut</i>	- wektor atrybutów, z którym porównywany jest atrybut warunku gałęzi drzewa
<i>przypadek</i>	- wektor przechowujący wektor wartości atrybutów przypadku
<i>przyporzadkowane_przypadki</i>	- mapa asocjacyjna zawierająca jako klucz odpowiednie etykiety, a jako wartość wektor wektorów przechowujący przypadki

Returns

- 0 - gdy brak błędów
- 13 - gdy podano nieprawidłowe dane

4.3.1.7 `przypisywanie_do_drzewa()`

```
void przypisywanie_do_drzewa (
    vector< decyzja > & drzewo,
    string * tablica,
    const int & linia )
```

Procedura wpisująca dane do drzewa.

Parameters

<i>drzewo</i>	- wektor struktur przechowujący <numer gałęzi> <warunek> <wartość jeśli tak> <wartość jeśli nie>
<i>*tablica</i>	- tablica pomocnicza do wpisywania danych do struktury
<i>linia</i>	- numer gałęzi

Returns

-

4.3.1.8 przypisywanie_do_warunku()

```
void przypisywanie_do_warunku (
    vector< vector< float >> & przypadki,
    vector< float > & wiersz )
```

Procedura wpisująca dane do wektora wektorów przypadków.

Parameters

<i>przypadki</i>	- wektor wektorów przechowujący wartości atrybutów przypadków
<i>wiersz</i>	- wektor pomocniczy do wpisywania danych do wektora wektorów

4.3.1.9 warunek_galezi()

```
bool warunek_galezi (
    int nr_przypadku,
    int nr_galezi,
    int nr_war_atrybutu,
    const vector< vector< float >> & przypadek,
    const vector< decyzja > & drzewo )
```

Funkcja sprawdzająca warunek gałęzi drzewa z atrybutem przypadku.

Parameters

<i>nr_przypadku</i>	- indeks wektora, sprawdzanego przypadku
<i>nr_galezi</i>	- indeks gałęzi, której warunek jest rozpatrywany
<i>przypadek</i>	- wektor przechowujący wektor wartości atrybutów przypadku
<i>drzewo</i>	- wektor struktur przechowujący warunki gałęzi

Returns

- true - jeśli warunek jest spełniony
- false - jeśli warunek nie jest spełniony

4.4 struktury.h File Reference

This graph shows which files directly or indirectly include this file:

Classes

- struct [decyzja](#)

4.5 widok.cpp File Reference

```
#include <string>
#include <sstream>
#include <vector>
#include <fstream>
#include <iostream>
#include "widok.h"
#include "model.h"
#include "struktury.h"
Include dependency graph for widok.cpp:
```

Functions

- bool [poprawnosc_odczytu](#) (ifstream &wejscie)
- int [czytanie_pliku_warunkowego](#) (vector< [decyzja](#) > &drzewo, ifstream &wejscie)
- int [czytanie_pliku_atrybutu](#) (vector< string > &atrybut, vector< vector< float >> &przypadki, ifstream &wejscie)

4.5.1 Function Documentation

4.5.1.1 czytanie_pliku_atrybutu()

```
int czytanie_pliku_atrybutu (
    vector< string > & atrybut,
    vector< vector< float >> & przypadki,
    ifstream & wejscie )
```

Funkcja czytająca i wpisująca do programu zawartość pliku wejściowego

Parameters

<i>atrybut</i>	- wektor przechowujący nazwy atrybutów
<i>przypadki</i>	- wektor wartości atrybutów danego przypadku
<i>wejscie</i>	- czytany plik tekstowy przypadków

Returns

- 0 - gdy brak błędów
- 13 - gdy podano nieprawidłowe dane

4.5.1.2 czytanie_pliku_warunkowego()

```
int czytanie_pliku_warunkowego (
    vector< decyzja > & drzewo,
    ifstream & wejscie )
```

Funkcja czytająca i wpisująca do programu zawartość pliku drzewa

Parameters

<i>drzewo</i>	- wektor struktur przechowujący <numer gałęzi> <warunek> <wartość jeśli tak> <wartość jeśli nie>
<i>wejście</i>	- czytany plik tekstowy z architekturą drzewa

Returns

- 0 - gdy brak błędów
- 13 - gdy podano nieprawidłowe dane

4.5.1.3 poprawnosc_odczytu()

```
bool poprawnosc_odczytu (
    ifstream & wejście )
```

Funkcja sprawdzająca otwarcie i poprawność odczytu pliku

Parameters

<i>wejście</i>	- czytany plik tekstowy
----------------	-------------------------

Returns

- true - jeżeli plik jest dobry
- false - jeżeli jest problem z odczytem

4.6 widok.h File Reference

```
#include <string>
#include <vector>
#include <fstream>
#include <iostream>
#include "struktury.h"
#include "model.h"
```

Include dependency graph for widok.h: This graph shows which files directly or indirectly include this file:

Functions

- bool [poprawnosc_odczytu](#) (ifstream &wejście)
- int [czytanie_pliku_warunkowego](#) (vector< [decyzja](#) > &drzewo, ifstream &wejście)
- int [czytanie_pliku_atrybutu](#) (vector< string > &atrybut, vector< vector< float >> &przypadki, ifstream &wejście)

4.6.1 Function Documentation

4.6.1.1 czytanie_pliku_atrybutu()

```
int czytanie_pliku_atrybutu (
    vector< string > & atrybut,
    vector< vector< float >> & przypadki,
    ifstream & wejscie )
```

Funkcja czytająca i wpisująca do programu zawartość pliku wejściowego

Parameters

<i>atrybut</i>	- wektor przechowujący nazwy atrybutów
<i>przypadki</i>	- wektor wartości atrybutów danego przypadku
<i>wejscie</i>	- czytany plik tekstowy przypadków

Returns

- 0 - gdy brak błędów
- 13 - gdy podano nieprawidłowe dane

4.6.1.2 czytanie_pliku_warunkowego()

```
int czytanie_pliku_warunkowego (
    vector< decyzja > & drzewo,
    ifstream & wejscie )
```

Funkcja czytająca i wpisująca do programu zawartość pliku drzewa

Parameters

<i>drzewo</i>	- wektor struktur przechowujący <numer gałęzi> <warunek> <wartość jeśli tak> <wartość jeśli nie>
<i>wejscie</i>	- czytany plik tekstowy z architekturą drzewa

Returns

- 0 - gdy brak błędów
- 13 - gdy podano nieprawidłowe dane

4.6.1.3 poprawnosc_odczytu()

```
bool poprawnosc_odczytu (
    ifstream & wejscie )
```

Funkcja sprawdzająca otwarcie i poprawność odczytu pliku

Parameters

<i>wejscie</i>	- czytany plik tekstowy
----------------	-------------------------

Returns

true - jeżeli plik jest dobry

false - jeżeli jest problem z odczytem

Index

czy_liczba
 model.cpp, 8
 model.h, 13

czy_liczba_naturalna
 model.cpp, 8
 model.h, 13

czytanie_pliku_atrybutu
 widok.cpp, 17
 widok.h, 19

czytanie_pliku_warunkowego
 widok.cpp, 17
 widok.h, 19

d_atrybut
 decyzja, 5

d_id
 decyzja, 6

d_nie
 decyzja, 6

d_tak
 decyzja, 6

d_wartosc
 decyzja, 6

d_znak
 decyzja, 6

decyzja, 5
 d_atrybut, 5
 d_id, 6
 d_nie, 6
 d_tak, 6
 d_wartosc, 6
 d_znak, 6

dodawanie_do_mapy
 model.cpp, 9
 model.h, 13

dzielenie_warunku
 model.cpp, 9
 model.h, 14

generowanie_pliku_wyjscia
 model.cpp, 9
 model.h, 14

main.cpp, 7

model.cpp, 7
 czy_liczba, 8
 czy_liczba_naturalna, 8
 dodawanie_do_mapy, 9
 dzielenie_warunku, 9
 generowanie_pliku_wyjscia, 9
 postepowanie_drzewa, 10
 przypisywanie_do_drzewa, 10
 przypisywanie_do_warunku, 11
 warunek_galezi, 11

model.h, 12
 czy_liczba, 13
 czy_liczba_naturalna, 13
 dodawanie_do_mapy, 13
 dzielenie_warunku, 14
 generowanie_pliku_wyjscia, 14
 postepowanie_drzewa, 14
 przypisywanie_do_drzewa, 15
 przypisywanie_do_warunku, 15
 warunek_galezi, 16

poprawnosc_odczytu
 widok.cpp, 18
 widok.h, 19

postepowanie_drzewa
 model.cpp, 10
 model.h, 14

przypisywanie_do_drzewa
 model.cpp, 10
 model.h, 15

przypisywanie_do_warunku
 model.cpp, 11
 model.h, 15

struktury.h, 16

warunek_galezi
 model.cpp, 11
 model.h, 16

widok.cpp, 17
 czytanie_pliku_atrybutu, 17
 czytanie_pliku_warunkowego, 17
 poprawnosc_odczytu, 18

widok.h, 18
 czytanie_pliku_atrybutu, 19
 czytanie_pliku_warunkowego, 19
 poprawnosc_odczytu, 19