

## Decyzje

Wygenerowano przez Doxygen 1.9.1



---

<b>1 Indeks klas</b>	<b>1</b>
1.1 Lista klas . . . . .	1
<b>2 Indeks plików</b>	<b>3</b>
2.1 Lista plików . . . . .	3
<b>3 Dokumentacja klas</b>	<b>5</b>
3.1 Dokumentacja struktury Node . . . . .	5
3.1.1 Opis szczegółowy . . . . .	5
<b>4 Dokumentacja plików</b>	<b>7</b>
4.1 Dokumentacja pliku funkcje.h . . . . .	7
4.1.1 Dokumentacja funkcji . . . . .	7
4.1.1.1 parse_string() . . . . .	7
4.1.1.2 read_data() . . . . .	8
4.1.1.3 read_tree() . . . . .	8
4.1.1.4 save_to_file() . . . . .	9
4.1.1.5 znajdz_etykiete() . . . . .	9
4.2 Dokumentacja pliku struktury.h . . . . .	9
<b>Indeks</b>	<b>11</b>



# Rozdział 1

## Indeks klas

### 1.1 Lista klas

Tutaj znajdują się klasy, struktury, unie i interfejsy wraz z ich krótkimi opisami:

<a href="#">Node</a>	5
----------------------	---



## Rozdział 2

# Indeks plików

### 2.1 Lista plików

Tutaj znajduje się lista wszystkich udokumentowanych plików z ich krótkimi opisami:

<a href="#">funkcje.h</a>	.....	7
<a href="#">struktury.h</a>	.....	9





## Rozdział 3

# Dokumentacja klas

### 3.1 Dokumentacja struktury Node

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

#### Atrybuty publiczne

- string **att**
- char **znak**
- float **wartosc**
- string **TAK**
- string **NIE**

#### 3.1.1 Opis szczegółowy

Struktura opisująca poszczególne wierzchołki drzewa

##### Parametry

<i>att</i>	Nazwa atrybutu
<i>znak</i>	operator
<i>wartosc</i>	wartość liczbową atrybutu
<i>TAK</i>	wyjście z aktualnego wierzchołka kiedy warunek jest spełniony
<i>NIE</i>	wyjście z aktualnego wierzchołka kiedy warunek nie jest spełniony

Dokumentacja dla tej struktury została wygenerowana z pliku:

- [struktury.h](#)



## Rozdział 4

# Dokumentacja plików

### 4.1 Dokumentacja pliku funkcje.h

```
#include "struktury.h"
#include <vector>
#include <fstream>
#include <iostream>
#include <sstream>
#include <unordered_map>
#include <stdlib.h>
```

#### Funkcje

- bool [read\\_tree](#) (ifstream &file, vector< [Node](#) > &tree)
- bool [parse\\_string](#) (string warunek, [Node](#) &node)
- void [read\\_data](#) (ifstream &file2, vector< vector< float >> &dane, vector< string > &atrybuty)
- string [znajdz\\_etykiety](#) (const vector< float > &att\_values, const vector< [Node](#) > &tree, const vector< string > &attributes)
- void [save\\_to\\_file](#) (ofstream &plikwyj, const unordered\_map< string, vector< vector< float >>> &wyniki)

#### 4.1.1 Dokumentacja funkcji

##### 4.1.1.1 [parse\\_string\(\)](#)

```
bool parse_string (
    string warunek,
    Node & node )
```

Funkcja wyciąga z parametru warunek, nazwę atrybutu, znak i wartość oraz zapisuje je do parametru node

**Parametry**

<i>warunek</i>	Przechowuje warunek
<i>node</i>	Struktura węzła drzewa

**Zwraca**

Funkcja zwraca false w przypadku nieprawidłowego warunku, a true po skutecznym wczytaniu struktury

**4.1.1.2 read\_data()**

```
void read_data (
    ifstream & file2,
    vector< vector< float >> & dane,
    vector< string > & atrybuty )
```

Funkcja wczytuje etykiety oraz wartości z pliku dane.txt, zapisuje je do wektorów, podane jako parametry funkcji

**Parametry**

<i>file2</i>	Plik z danymi
<i>dane</i>	Wektor z przykładami
<i>atrybuty</i>	Wektor z atrybutami

**Zwraca**

Funkcja nie zwraca niczego

**4.1.1.3 read\_tree()**

```
bool read_tree (
    ifstream & file,
    vector< Node > & tree )
```

Funkcja wczytująca z pliku struktura.txt strukturę drzewa do podanego wektora tree

**Parametry**

<i>file</i>	Plik drzewa
<i>tree</i>	Wektor przechowujący strukturę drzewa

**Zwraca**

Funkcja zwraca false w przypadku nieprawidłowego warunku, a true po skutecznym wczytaniu warunku

**4.1.1.4 save\_to\_file()**

```
void save_to_file (
    ofstream & plikwyj,
    const unordered_map< string, vector< vector< float >>> & wyniki )
```

Funkcja zapisuje już przypisane wartości i etykiety do pliku "wyjscie.txt"

**Parametry**

<i>plikwyj</i>	Plik wyjściowy
<i>wyniki</i>	Przydzielone już przykłady do etykiet

**Zwraca**

Funkcja nie zwraca niczego

**4.1.1.5 znajdz\_etykiete()**

```
string znajdz_etykiete (
    const vector< float > & att_values,
    const vector< Node > & tree,
    const vector< string > & attributes )
```

Funkcja znajduje etykietę i przypisuje jej wartości

**Parametry**

<i>att_values</i>	Wektor z wartościami atrybutów
<i>tree</i>	Wektor węzłów drzewa
<i>attributes</i>	Wektor z atrybutami

**Zwraca**

Funkcja zwraca etykietę dla zestawu danych

**4.2 Dokumentacja pliku struktury.h**

```
#include <iostream>
```

## Komponenty

- struct [Node](#)

# Indeks

funkcje.h, [7](#)  
    [parse\\_string](#), [7](#)  
    [read\\_data](#), [8](#)  
    [read\\_tree](#), [8](#)  
    [save\\_to\\_file](#), [9](#)  
    [znajdz\\_etykiete](#), [9](#)

Node, [5](#)

[parse\\_string](#)  
    funkcje.h, [7](#)

[read\\_data](#)  
    funkcje.h, [8](#)

[read\\_tree](#)  
    funkcje.h, [8](#)

[save\\_to\\_file](#)  
    funkcje.h, [9](#)

[struktury.h](#), [9](#)

[znajdz\\_etykiete](#)  
    funkcje.h, [9](#)