

Drzewo Decyzyjne
v3.0

Generated by Doxygen 1.9.5

1 Todo List	1
2 Class Index	3
2.1 Class List	3
3 File Index	5
3.1 File List	5
4 Class Documentation	7
4.1 PunktDrzewaDecyzyjnego Struct Reference	7
4.1.1 Detailed Description	7
4.1.2 Member Data Documentation	7
4.1.2.1 atrybut	8
4.1.2.2 indeks	8
4.1.2.3 klasyfikacja	8
4.1.2.4 klasyfikacjaOstateczna	8
4.1.2.5 wymaganie	8
4.1.2.6 znakTestu	8
5 File Documentation	9
5.1 funkcje.cpp File Reference	9
5.1.1 Function Documentation	9
5.1.1.1 pobierzDane()	9
5.1.1.2 pobierzDrzewo()	10
5.1.1.3 pobierzIndeks()	10
5.1.1.4 porownanie()	10
5.1.1.5 zapiszDoPliku()	11
5.2 Funkcje.h File Reference	11
5.2.1 Function Documentation	12
5.2.1.1 pobierzDane()	12
5.2.1.2 pobierzDrzewo()	12
5.2.1.3 pobierzIndeks()	13
5.2.1.4 porownanie()	13
5.2.1.5 zapiszDoPliku()	13
5.3 Funkcje.h	14
5.4 Projekt_semestralny_drzewo_decyzyjne_sekcja06.cpp File Reference	14
5.4.1 Function Documentation	14
5.4.1.1 main()	15
5.5 Struktura.h File Reference	15
5.6 Struktura.h	16
Index	17

Chapter 1

Todo List

Member main (p. 14) (int argc, char *argv[])

Czytanie parametrów drzewa decyzyjnego i przechowanie ich. Można w sumie użyć struktury do tego by później było łatwiej z if-ami może — DONE

Porównywanie wartości do drzewa decyzyjnego - tu by w sumie przydało się to parami sprawdzać (tzn. `wyskok[0]` i `wzrost[0]`) — DONE

Opracować jak wywołać ten program z konsoli i wraz z podaniem parametrów (plików tekstowych) - wchodzisz do CMD do projektu -> x64 -> Debug -> nazwa.exe i lecisz — DONE

Opracować żeby program podawał instrukcje co jak zrobić gdy użytkownik nie poda plików wejściowych a także może niech poda lokalizację gdzie zapisał plik tekstowy z wynikiem — DONE

Opisać wszystko z pomocą Doxygena - ogarnąć w ogóle jak się to coś robi — DONE

Naprawić porównywanie i zapisywanie do wektorów bo program nie jest uniwersalny :(- zamiast wektorów 'koszykowka' i 'lekkoatletyka' trzeba by użyć listy wektorów o długości == ilości klasyfikacji — DONE

Have fun :)

Chapter 2

Class Index

2.1 Class List

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

PunktDrzewaDecyzyjnego	7
-----------------------------------------	---

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

funkcje.cpp	9
Funkcje.h	11
Projekt_semestralny_drzewo_decyzyjne_sekcja06.cpp	14
Struktura.h	15

Chapter 4

Class Documentation

4.1 PunktDrzewaDecyzyjnego Struct Reference

```
#include <Struktura.h>
```

Public Attributes

- `std::string` **atrybut**
- `std::string` **znakTestu**
- `double` **wymaganie** {}
- `int` **indeks** {}
- `std::string` **klasyfikacja**
- `std::string` **klasyfikacjaOstateczna**

4.1.1 Detailed Description

Struktura opisuje punkt w drzewie decyzyjnym

Parameters

<i>atrybut</i>	Nazwa atrybutu do testowania
<i>znakTestu</i>	Jaki test ma zostac wykonany
<i>wymaganie</i>	Wartosc graniczna (do ktorej bedziemy przyrownywac)
<i>indeks</i>	Indeks nastepnego punktu drzewa decyzyjnego kiedy test sie nie powiedzie
<i>klasyfikacja</i>	Do ktorej dyscypliny zakwalifikowac kiedy test sie powiedzie
<i>klasyfikacjaOstateczna</i>	Wykorzystane przy ostatnim punkcie drzewa kiedy test sie nie powiedzie

4.1.2 Member Data Documentation

4.1.2.1 atrybut

```
std::string PunktDrzewaDecyzyjnego::atrybut
```

4.1.2.2 indeks

```
int PunktDrzewaDecyzyjnego::indeks {}
```

4.1.2.3 klasyfikacja

```
std::string PunktDrzewaDecyzyjnego::klasyfikacja
```

4.1.2.4 klasyfikacjaOstateczna

```
std::string PunktDrzewaDecyzyjnego::klasyfikacjaOstateczna
```

4.1.2.5 wymaganie

```
double PunktDrzewaDecyzyjnego::wymaganie {}
```

4.1.2.6 znakTestu

```
std::string PunktDrzewaDecyzyjnego::znakTestu
```

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

- **Struktura.h**

Chapter 5

File Documentation

5.1 funkcje.cpp File Reference

```
#include <iostream>
#include <sstream>
#include <vector>
#include <fstream>
#include <map>
#include <set>
#include <string>
#include <iomanip>
#include "Funkcje.h"
```

Functions

- void **pobierzDane** (std::string nazwaPliku, std::map< int, std::vector< std::pair< std::string, double > > > &atrybuty, std::vector< std::string > &nazwyAtrybutow)
- void **pobierzIndeks** (std::string nazwaPliku, int &indeksMaksymalny)
- void **pobierzDrzewo** (std::string nazwaPliku, std::map< int, **PunktDrzewaDecyzyjnego** > &drzewo, int &indeksMaksymalny, std::set< std::string > &nazwyPrzydzielenia)
- void **porownanie** (std::map< int, **PunktDrzewaDecyzyjnego** > &drzewo, std::map< int, std::vector< std::pair< std::string, double > > > &atrybuty, int &indeksMax, std::map< std::string, std::vector< double > > &przydzieleni)
- void **zapiszDoPliku** (std::string nazwaPlikuWyjsciowego, std::map< std::string, std::vector< double > > &przydzieleni, std::set< std::string > &nazwyPrzydzielenia, std::vector< std::string > &nazwyAtrybutow)

5.1.1 Function Documentation

5.1.1.1 pobierzDane()

```
void pobierzDane (
    std::string nazwaPliku,
    std::map< int, std::vector< std::pair< std::string, double > > > & atrybuty,
    std::vector< std::string > & nazwyAtrybutow )
```

Funkcja pobiera dane z pliku wejsciowego

Parameters

<i>nazwaPliku</i>	Nazwa pliku wyjściowego (w formacie .txt)
<i>atrybuty</i>	Referencja do mapy wektorów przechowującej atrybuty z pliku wejściowego
<i>nazwyAtrybutow</i>	Referencja do wektora zawierającego nazwy atrybutów z pliku wejściowego

5.1.1.2 pobierzDrzewo()

```
void pobierzDrzewo (
    std::string nazwaPliku,
    std::map< int, PunktDrzewaDecyzyjnego > & drzewo,
    int & indeksMaksymalny,
    std::set< std::string > & nazwyPrzydzielenia )
```

Funkcja pobiera dane z pliku drzewa i zapisuje je do mapy

Parameters

<i>nazwaPliku</i>	Nazwa pliku drzewa (w formacie .txt)
<i>drzewo</i>	Referencja na mapę drzewa decyzyjnego
<i>indeksMaksymalny</i>	Referencja do maksymalnego indeksu drzewa
<i>nazwyPrzydzielenia</i>	Referencja do setu przechowującego miejsca przydzielenia w drzewie

5.1.1.3 pobierzIndeks()

```
void pobierzIndeks (
    std::string nazwaPliku,
    int & indeksMaksymalny )
```

Funkcja pobiera indeks maksymalny w pliku z drzewem

Parameters

<i>nazwaPliku</i>	Nazwa pliku drzewa (w formacie .txt)
<i>indeksMaksymalny</i>	Referencja do maksymalnego indeksu drzewa

5.1.1.4 porownanie()

```
void porownanie (
    std::map< int, PunktDrzewaDecyzyjnego > & drzewo,
    std::map< int, std::vector< std::pair< std::string, double > > > & atrybuty,
```

```
int & indeksMax,
std::map< std::string, std::vector< double > > & przydzieleni )
```

Funkcja przyrównuje wartości z pliku wejściowego do drzewa decyzyjnego i przypisuje wszystkie dane do odpowiednich miejsc

Parameters

<i>drzewo</i>	Referencja na mapę drzewa decyzyjnego
<i>atrybuty</i>	Referencja na mapę wektorów z atrybutami osób z pliku wejściowego
<i>indeksMax</i>	Referencja na zmienną z maksymalnym indeksem drzewa decyzyjnego
<i>przydzieleni</i>	Referencja na mapę przechowującą przydzielone już osoby

5.1.1.5 zapiszDoPliku()

```
void zapiszDoPliku (
    std::string nazwaPlikuWyjsciowego,
    std::map< std::string, std::vector< double > > przydzieleni,
    std::set< std::string > nazwyPrzydzielenia,
    std::vector< std::string > nazwyAtrybutow )
```

Funkcja zapisuje osoby przydzielone do pliku wyjściowego

Parameters

<i>nazwaPlikuWyjsciowego</i>	Zmienna przechowująca nazwę pliku z końcowym efektem kodu
<i>przydzieleni</i>	Mapa z osobami przydzielonymi
<i>nazwyPrzydzielenia</i>	Set przechowujący miejsca przydzielenia z drzewa
<i>nazwyAtrybutow</i>	Wektor z nazwami atrybutów przedstawionych w pliku wejściowym

5.2 Funkcje.h File Reference

```
#include <vector>
#include <string>
#include <fstream>
#include <sstream>
#include <map>
#include <set>
#include "Struktura.h"
```

Functions

- void **pobierzDane** (std::string nazwaPliku, std::map< int, std::vector< std::pair< std::string, double > > > & atrybuty, std::vector< std::string > & nazwyAtrybutow)
- void **pobierzIndeks** (std::string nazwaPliku, int & indeksMaksymalny)

- void **pobierzDrzewo** (std::string nazwaPliku, std::map< int, **PunktDrzewaDecyzyjnego** > &drzewo, int &indeksMaksymalny, std::set< std::string > &nazwyPrzydzielenia)
- void **porownanie** (std::map< int, **PunktDrzewaDecyzyjnego** > &drzewo, std::map< int, std::vector< std::pair< std::string, double > > > &atrybuty, int &indeksMax, std::map< std::string, std::vector< double > > &przydzieleni)
- void **zapiszDoPliku** (std::string nazwaPlikuWyjsciowego, std::map< std::string, std::vector< double > > &przydzieleni, std::set< std::string > &nazwyPrzydzielenia, std::vector< std::string > &nazwyAtrybutow)

5.2.1 Function Documentation

5.2.1.1 pobierzDane()

```
void pobierzDane (
    std::string nazwaPliku,
    std::map< int, std::vector< std::pair< std::string, double > > > & atrybuty,
    std::vector< std::string > & nazwyAtrybutow )
```

Funkcja pobiera dane z pliku wejsciowego

Parameters

<i>nazwaPliku</i>	Nazwa pliku wyjsciowego (w formacie .txt)
<i>atrybuty</i>	Referencja do mapy wektorow przechowujacej atrybuty z pliku wejsciowego
<i>nazwyAtrybutow</i>	Referencja do wektora zawierajacego nazwy atrybutow z pliku wejsciowego

5.2.1.2 pobierzDrzewo()

```
void pobierzDrzewo (
    std::string nazwaPliku,
    std::map< int, PunktDrzewaDecyzyjnego > & drzewo,
    int & indeksMaksymalny,
    std::set< std::string > & nazwyPrzydzielenia )
```

Funkcja pobiera dane z pliku drzewa i zapisuje je do mapy

Parameters

<i>nazwaPliku</i>	Nazwa pliku drzewa (w formacie .txt)
<i>drzewo</i>	Referencja na mape drzewa decyzyjnego
<i>indeksMaksymalny</i>	Referencja do maksymalnego indeksu drzewa
<i>nazwyPrzydzielenia</i>	Referencja do setu przechowujacego miejsca przydzielenia w drzewie

5.2.1.3 pobierzIndeks()

```
void pobierzIndeks (
    std::string nazwaPliku,
    int & indeksMaksymalny )
```

Funkcja pobiera indeks maksymalny w pliku z drzewem

Parameters

<i>nazwaPliku</i>	Nazwa pliku drzewa (w formacie .txt)
<i>indeksMaksymalny</i>	Referencja do maksymalnego indeksu drzewa

5.2.1.4 porownanie()

```
void porownanie (
    std::map< int, PunktDrzewaDecyzyjnego > & drzewo,
    std::map< int, std::vector< std::pair< std::string, double > > > & atrybuty,
    int & indeksMax,
    std::map< std::string, std::vector< double > > & przydzieleni )
```

Funkcja przyrównuje wartości z pliku wejściowego do drzewa decyzyjnego i przypisuje wszystkie dane do odpowiednich miejsc

Parameters

<i>drzewo</i>	Referencja na mapę drzewa decyzyjnego
<i>atrybuty</i>	Referencja na mapę wektorów z atrybutami osób z pliku wejściowego
<i>indeksMax</i>	Referencja na zmienną z maksymalnym indeksem drzewa decyzyjnego
<i>przydzieleni</i>	Referencja na mapę przechowującą przydzielone już osoby

5.2.1.5 zapiszDoPliku()

```
void zapiszDoPliku (
    std::string nazwaPlikuWyjściowego,
    std::map< std::string, std::vector< double > > przydzieleni,
    std::set< std::string > nazwyPrzydzielenia,
    std::vector< std::string > nazwyAtrybutow )
```

Funkcja zapisuje osoby przydzielone do pliku wyjściowego

Parameters

<i>nazwaPlikuWyjściowego</i>	Zmienna przechowująca nazwę pliku z końcowym efektem kodu
<i>przydzieleni</i>	Mapa z osobami przydzielonymi
<i>nazwyPrzydzielenia</i>	Set przechowujący miejsca przydzielenia z drzewa
<i>nazwyAtrybutow</i>	Wektor z nazwami atrybutów przedstawionych w pliku wejściowym

5.3 Funkcje.h

Go to the documentation of this file.

```
1 #ifndef FUNKCJE_H
2 #define FUNKCJE_H
3
4 #include <vector>
5 #include <string>
6 #include <fstream>
7 #include <sstream>
8 #include <map>
9 #include <set>
10
11 #include "Struktura.h"
12
19 void pobierzDane(std::string nazwaPliku, std::map<int, std::vector<std::pair<std::string, double>>&
    atrybuty, std::vector<std::string>& nazwyAtrybutow);
20
26 void pobierzIndeks(std::string nazwaPliku, int& indeksMaksymalny);
27
35 void pobierzDrzewo(std::string nazwaPliku, std::map<int, PunktDrzewaDecyzyjnego>& drzewo, int&
    indeksMaksymalny, std::set<std::string>& nazwyPrzydzielenia);
36
44 void porownanie(std::map<int, PunktDrzewaDecyzyjnego>& drzewo, std::map<int,
    std::vector<std::pair<std::string, double>>& atrybuty, int& indeksMax, std::map<std::string,
    std::vector<double>& przydzieleni);
45
53 void zapiszDoPliku(std::string nazwaPlikuWyjsciowego, std::map<std::string, std::vector<double>>
    przydzieleni, std::set<std::string> nazwyPrzydzielenia, std::vector<std::string> nazwyAtrybutow);
54
55
56 #endif
```

5.4 Projekt_semestralny_drzewo_decyzyjne_sekcja06.cpp File Reference

```
#include <iostream>
#include <sstream>
#include <fstream>
#include <vector>
#include <map>
#include <string>
#include <set>
#include "Funkcje.h"
```

Functions

- int **main** (int argc, char *argv[])

5.4.1 Function Documentation

5.4.1.1 main()

```
int main (
    int argc,
    char * argv[] )
```

Program jest uniwersalna implementacja drzewa decyzyjnego

Date

2023-01-21

Author

Kacper Sikorski, I rok Informatyki, 3 grupa dziekanska, sekcja 06 PPK

Todo Czytanie parametrow drzewa decyzyjnego i przechowanie ich. Mozna w sumie uzyc struktury do tego by pozniej bylo latwiej z if-ami moze — DONE

Todo Porownywanie wartosci do drzewa decyzyjnego - tu by w sumie przydalo sie to parami sprawdzac (tzn. wyskok[0] i wzrost[0]) — DONE

Todo Opracowac jak wywolac ten program z konsoli i wraz z podaniem parametrow (plikow tekstowych) - wchodzisz do CMD do projektu -> x64 -> Debug -> nazwa.exe i lecisz — DONE

Todo Opracowac zeby program podawal instrukcje co jak zrobic gdy uzytkownik nie poda plikow wejsciowych a takze moze niech poda lokalizacje gdzie zapisal plik tekstowy z wynikiem — DONE

Todo Opisac wszystko z pomoca Doxygena - ogarnac w ogole jak sie to cos robi — DONE

Todo Naprawic porownywanie i zapisywanie do wektorow bo program nie jest uniwersalny :(- zamiast wektorow 'koszykowka' i 'lekkoatletyka' trzeba by uzyc listy wektorow o dlugosci == ilosci klasyfikacji — DONE

Todo Have fun :)

5.5 Struktura.h File Reference

```
#include <vector>
#include <string>
#include <fstream>
#include <sstream>
#include <map>
```

Classes

- struct **PunktDrzewaDecyzyjnego**

5.6 Struktura.h

Go to the documentation of this file.

```
1 #ifndef STRUKTURA_H
2 #define STRUKTURA_H
3
4 #include <vector>
5 #include <string>
6 #include <fstream>
7 #include <sstream>
8 #include <map>
9
10 struct PunktDrzewaDecyzyjnego
11 {
12     std::string atrybut;
13     std::string znakTestu;
14     double wymaganie{};
15     int indeks{};
16     std::string klasyfikacja;
17     std::string klasyfikacjaOstateczna;
18 };
19
20 #endif
```

Index

- atrybut
 - PunktDrzewaDecyzyjnego, 7
- funkcje.cpp, 9
 - pobierzDane, 9
 - pobierzDrzewo, 10
 - pobierzIndeks, 10
 - porownanie, 10
 - zapiszDoPliku, 11
- Funkcje.h, 11
 - pobierzDane, 12
 - pobierzDrzewo, 12
 - pobierzIndeks, 12
 - porownanie, 13
 - zapiszDoPliku, 13
- indeks
 - PunktDrzewaDecyzyjnego, 8
- klasyfikacja
 - PunktDrzewaDecyzyjnego, 8
- klasyfikacjaOstateczna
 - PunktDrzewaDecyzyjnego, 8
- main
 - Projekt_semestralny_drzewo_decyzyjne_sekcja06.cpp, 14
- pobierzDane
 - funkcje.cpp, 9
 - Funkcje.h, 12
- pobierzDrzewo
 - funkcje.cpp, 10
 - Funkcje.h, 12
- pobierzIndeks
 - funkcje.cpp, 10
 - Funkcje.h, 12
- porownanie
 - funkcje.cpp, 10
 - Funkcje.h, 13
- Projekt_semestralny_drzewo_decyzyjne_sekcja06.cpp, 14
 - main, 14
- PunktDrzewaDecyzyjnego, 7
 - atrybut, 7
 - indeks, 8
 - klasyfikacja, 8
 - klasyfikacjaOstateczna, 8
 - wymaganie, 8
 - znakTestu, 8
 - Struktura.h, 15
 - wymaganie
 - PunktDrzewaDecyzyjnego, 8
 - zapiszDoPliku
 - funkcje.cpp, 11
 - Funkcje.h, 13
 - znakTestu
 - PunktDrzewaDecyzyjnego, 8