

My Project

Wygenerowano za pomocą Doxygen 1.15.0

1 Indeks klas	1
1.1 Lista klas	1
2 Indeks plików	3
2.1 Lista plików	3
3 Dokumentacja klas	5
3.1 Dokumentacja klasy BST	5
3.1.1 Opis szczegółowy	5
3.1.2 Dokumentacja funkcji składowych	6
3.1.2.1 findPath()	6
3.1.2.2 insert()	6
3.1.2.3 remove()	6
3.1.3 Dokumentacja przyjaciół i powiązanych symboli	6
3.1.3.1 FileHandler	6
3.2 Dokumentacja klasy FileHandler	7
3.2.1 Opis szczegółowy	7
3.2.2 Dokumentacja funkcji składowych	7
3.2.2.1 loadFromBinary()	7
3.2.2.2 loadFromText()	8
3.2.2.3 saveToBinary()	8
3.2.2.4 saveToText()	8
4 Dokumentacja plików	11
4.1 Dokumentacja pliku C:/Users/micha/Desktop/Drzewo-BST-master/Drzewo-BST-master/BST.cpp	11
4.1.1 Opis szczegółowy	11
4.2 Dokumentacja pliku C:/Users/micha/Desktop/Drzewo-BST-master/Drzewo-BST-master/BST.h	11
4.2.1 Opis szczegółowy	12
4.3 BST.h	12
4.4 Dokumentacja pliku C:/Users/micha/Desktop/Drzewo-BST-master/Drzewo-BST-master/FileHandler.cpp	13
4.4.1 Opis szczegółowy	13
4.5 Dokumentacja pliku C:/Users/micha/Desktop/Drzewo-BST-master/Drzewo-BST-master/FileHandler.h	13
4.5.1 Opis szczegółowy	13
4.6 FileHandler.h	14
4.7 Dokumentacja pliku C:/Users/micha/Desktop/Drzewo-BST-master/Drzewo-BST-master/main.cpp	14
4.7.1 Opis szczegółowy	14
4.7.2 Dokumentacja funkcji	15
4.7.2.1 clearInputBuffer()	15
4.7.2.2 main()	15
Skorowidz	17

Rozdział 1

Indeks klas

1.1 Lista klas

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

BST	Implementacja drzewa binarnego poszukiwań (BST)	5
FileHandler	Klasa obsługująca operacje plikowe dla drzewa BST	7

Rozdział 2

Indeks plików

2.1 Lista plików

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

C:/Users/micha/Desktop/Drzewo-BST-master/Drzewo-BST-master/ BST.cpp	11
Implementacja metod klasy BST	
C:/Users/micha/Desktop/Drzewo-BST-master/Drzewo-BST-master/ BST.h	11
Definicja klasy BST (Binary Search Tree)	
C:/Users/micha/Desktop/Drzewo-BST-master/Drzewo-BST-master/ FileHandler.cpp	13
Implementacja metod klasy FileHandler	
C:/Users/micha/Desktop/Drzewo-BST-master/Drzewo-BST-master/ FileHandler.h	13
Definicja klasy FileHandler	
C:/Users/micha/Desktop/Drzewo-BST-master/Drzewo-BST-master/ main.cpp	14
Główny plik programu z interfejsem użytkownika (menu)	

Rozdział 3

Dokumentacja klas

3.1 Dokumentacja klasy BST

Implementacja drzewa binarnego poszukiwan ([BST](#)).

```
#include <BST.h>
```

Metody publiczne

- **BST ()**
Konstruktor domyslny, tworzy puste drzewo.
- **~BST ()**
Destruktor, zwalnia pamiec po wszystkich wezlach.
- void **insert** (int data)
Publiczna metoda dodajaca element do drzewa.
- void **remove** (int data)
Publiczna metoda usuwajaca element z drzewa.
- void **clear** ()
Publiczna metoda usuwajaca wszystkie elementy z drzewa.
- vector< int > **findPath** (int data)
Wyszukuje sciezke od korzenia do wezla o podanej wartosci.
- void **display** ()
Wyswietla menu wyboru metody wyswietlania drzewa i je wyswietla.

Przyjaciele

- class [FileHandler](#)
Zaprzyjaznienie klasy [FileHandler](#).

3.1.1 Opis szczegółowy

Implementacja drzewa binarnego poszukiwan ([BST](#)).

- Klasa przechowuje elementy w uporzadkowanej strukturze drzewa, umoziwiajac szybkie wyszukiwanie, dodawanie i usuwanie elementow.

3.1.2 Dokumentacja funkcji składowych

3.1.2.1 `findPath()`

```
vector< int > BST::findPath (
    int data)
```

Wyszukuje sciezke od korzenia do wezla o podanej wartosci.

Parametry

<i>data</i>	Wartosc do znalezienia.
-------------	-------------------------

Zwrota

Wektor (STL) zawierajacy wartosci wezelow na sciezce. Pusty wektor, jesli elementu nie znaleziono.

3.1.2.2 `insert()`

```
void BST::insert (
    int data)
```

Publiczna metoda dodajaca element do drzewa.

Parametry

<i>data</i>	Wartosc do dodania.
-------------	---------------------

3.1.2.3 `remove()`

```
void BST::remove (
    int data)
```

Publiczna metoda usuwajaca element z drzewa.

Parametry

<i>data</i>	Wartosc do usuniecia.
-------------	-----------------------

3.1.3 Dokumentacja przyjaciół i powiązanych symboli

3.1.3.1 `FileHandler`

```
friend class FileHandler [friend]
```

Zaprzylaznienie klasy `FileHandler`.

- Pozwala klasie `FileHandler` na dostep do prywatnych składowych (`root`) i prywatnych metod (`serialize`, `deserialize`, `saveToText`) klasy `BST`.

Dokumentacja dla tej klasy została wygenerowana z plików:

- C:/Users/micha/Desktop/Drzewo-BST-master/Drzewo-BST-master/[BST.h](#)
- C:/Users/micha/Desktop/Drzewo-BST-master/Drzewo-BST-master/[BST.cpp](#)

3.2 Dokumentacja klasy FileHandler

Klasa obslujaca operacje plikowe dla drzewa **BST**.

```
#include <FileHandler.h>
```

Metody publiczne

- bool **saveToText** (**BST** &tree, const string &filename)
Zapisuje drzewo do pliku tekstopowego (w kolejnosci Inorder).
- bool **saveToBinary** (**BST** &tree, const string &filename)
Zapisuje (serializuje) cala strukture drzewa do pliku binarnego.
- bool **loadFromBinary** (**BST** &tree, const string &filename)
Wczytuje (deserializuje) cala strukture drzewa z pliku binarnego.
- bool **loadFromText** (**BST** &tree, const string &filename)
Wczytuje liczby z pliku tekstopowego i dodaje je do drzewa.

3.2.1 Opis szczegółowy

Klasa obslujaca operacje plikowe dla drzewa **BST**.

- Umoziwi zapisywanie stanu drzewa do plikow tekstowych i binarnych oraz odczytywanie danych (zastepujac lub dodajac do) drzewa. Jest zaprzyjazniona z klasa **BST**, aby miec dostep do jej prywatnych metod.

3.2.2 Dokumentacja funkcji składowych

3.2.2.1 **loadFromBinary()**

```
bool FileHandler::loadFromBinary (
    BST & tree,
    const string & filename)
```

Wczytuje (deserializuje) cala strukture drzewa z pliku binarnego.

Ostrzeżenie

Ta operacja usuwa (czysci) istniejace drzewo przed wczytaniem.

Parametry

<i>tree</i>	Referencja do obiektu drzewa BST , ktore ma byc zastapione.
<i>filename</i>	Nazwa binarnego pliku wejsciowego.

Zwraca

true jesli odczyt sie powiodl, false w przeciwnym razie.

3.2.2.2 loadFromText()

```
bool FileHandler::loadFromText (
    BST & tree,
    const string & filename)
```

Wczytuje liczby z pliku tekstowego i dodaje je do drzewa.

Nota

Ta operacja dodaje elementy do istniejącego drzewa (nie czysci go).

Parametry

<i>tree</i>	Referencja do obiektu drzewa BST .
<i>filename</i>	Nazwa pliku tekstowego z danymi (liczby oddzielone bialymi znakami).

Zwroca

true jesli odczyt sie powiodl, false w przeciwnym razie.

3.2.2.3 saveToBinary()

```
bool FileHandler::saveToBinary (
    BST & tree,
    const string & filename)
```

Zapisuje (serializuje) cała strukturę drzewa do pliku binarnego.

Parametry

<i>tree</i>	Referencja do obiektu drzewa BST .
<i>filename</i>	Nazwa binarnego pliku wyjsciowego.

Zwroca

true jesli zapis sie powiodl, false w przeciwnym razie.

3.2.2.4 saveToText()

```
bool FileHandler::saveToText (
    BST & tree,
    const string & filename)
```

Zapisuje drzewo do pliku tekstowego (w kolejnosci Inorder).

Parametry

<i>tree</i>	Referencja do obiektu drzewa BST , ktore ma byc zapisane.
<i>filename</i>	Nazwa pliku wyjsciowego.

Zwroca

true jesli zapis sie powiodl, false w przeciwnym razie.

Dokumentacja dla tej klasy została wygenerowana z plików:

- C:/Users/micha/Desktop/Drzewo-BST-master/Drzewo-BST-master/[FileHandler.h](#)
- C:/Users/micha/Desktop/Drzewo-BST-master/Drzewo-BST-master/[FileHandler.cpp](#)

Rozdział 4

Dokumentacja plików

4.1 Dokumentacja pliku C:/Users/micha/Desktop/Drzewo-BST-master/← Drzewo-BST-master/BST.cpp

Implementacja metod klasy [BST](#).

```
#include "BST.h"
#include <iostream>
```

4.1.1 Opis szczegółowy

Implementacja metod klasy [BST](#).

4.2 Dokumentacja pliku C:/Users/micha/Desktop/Drzewo-BST-master/← Drzewo-BST-master/BST.h

Definicja klasy [BST](#) (Binary Search Tree).

```
#include <iostream>
#include <vector>
#include <string>
#include <fstream>
#include <iomanip>
```

Komponenty

- class [BST](#)

Implementacja drzewa binarnego poszukiwań ([BST](#)).

4.2.1 Opis szczegółowy

Definicja klasy **BST** (Binary Search Tree).

- Plik ten zawiera deklaracje klasy **BST**, w tym jej wewnętrzna strukturę Node oraz wszystkie metody do zarządzania drzewem (dodawanie, usuwanie, wyszukiwanie, wyświetlanie).

4.3 BST.h

[Idź do dokumentacji tego pliku.](#)

```

00001
00002
00003 #pragma once
00004
00005 #include <iostream>
00006 #include <vector>
00007 #include <string>
00008 #include <fstream>
00009 #include <iomanip> // Do printGraphical
00010
00011 using namespace std;
00012
00013 // Uzywamy forward-declaration, aby uniknac cyklicznych zaleznosci
00014 class FileHandler;
00015
00016 class BST {
00017 private:
00018     struct Node {
00019         int data;
00020         Node* left;
00021         Node* right;
00022
00023         Node(int val) : data(val), left(nullptr), right(nullptr) {}
00024     };
00025
00026     Node* root;
00027
00028 // --- Metody pomocnicze (rekurencyjne) ---
00029
00030     Node* insert(Node* node, int data);
00031
00032     Node* remove(Node* node, int data);
00033
00034     Node* findMin(Node* node);
00035
00036     void clear(Node* node);
00037
00038     bool findPath(Node* node, int data, vector<int>& path);
00039
00040 // --- Metody wyswietlania ---
00041
00042     void printPreorder(Node* node);
00043
00044     void printInorder(Node* node);
00045
00046     void printPostorder(Node* node);
00047
00048     void printGraphical(Node* node, int space, int count);
00049
00050 // --- Metody pomocnicze do zapisu/odczytu (wywolywane przez FileHandler) ---
00051
00052     void saveToText(Node* node, ofstream& outFile);
00053
00054     void serialize(Node* node, ofstream& outFile);
00055
00056     Node* deserialize(ifstream& inFile);
00057
00058 public:
00059     BST();
00060
00061     ~BST();
00062
00063     void insert(int data);
00064
00065     void remove(int data);
00066
00067
00068
00069
00070
00071
00072
00073
00074
00075
00076
00077
00078
00079
00080
00081
00082
00083
00084
00085
00086
00087
00088
00089
00090
00091
00092
00093
00094
00095
00096
00097
00098
00099
00100
00101
00102
00103
00104
00105
00106
00107
00108
00109
00110
00111
00112
00113
00114
00115
00116
00117
00118
00119
00120
00121
00122
00123
00124
00125
00126
00127
00128
00129
00130
00131
00132
00133
00134
00135
00136
00137
00138
00139
00140
00141
00142
00143
00144
00145
00146
00147
00148
00149
00150
00151
00152
00153
00154
00155

```

4.4 Dokumentacja pliku C:/Users/micha/Desktop/Drzewo-BST-master/Drzewo-BST-master/FileHandler.cpp

```
00159     void clear();  
00160  
00167     vector<int> findPath(int data);  
00168  
00172     void display();  
00173  
00179     friend class FileHandler;  
00180 };
```

4.4 Dokumentacja pliku C:/Users/micha/Desktop/Drzewo-BST-master/Drzewo-BST-master/FileHandler.cpp

Implementacja metod klasy [FileHandler](#).

```
#include "FileHandler.h"  
#include <fstream>  
#include <iostream>
```

4.4.1 Opis szczegółowy

Implementacja metod klasy [FileHandler](#).

4.5 Dokumentacja pliku C:/Users/micha/Desktop/Drzewo-BST-master/Drzewo-BST-master/FileHandler.h

Definicja klasy [FileHandler](#).

```
#include <string>  
#include "BST.h"
```

Komponenty

- class [FileHandler](#)
Klasa obsługująca operacje plikowe dla drzewa [BST](#).

4.5.1 Opis szczegółowy

Definicja klasy [FileHandler](#).

- Klasa ta odpowiada za operacje wejścia/wyjścia na plikach (tekstowych i binarnych) w celu zapisywania i odczytywania struktury drzewa [BST](#).

4.6 FileHandler.h

[Idź do dokumentacji tego pliku.](#)

```

00001
00007
00008 #pragma once
00009
00010 #include <string>
00011 #include "BST.h" // Potrzebujemy pełnej definicji BST
00012
00013 using namespace std;
00014
00021 class FileHandler {
00022 public:
00029     bool saveToText(BST& tree, const string& filename);
00030
00037     bool saveToBinary(BST& tree, const string& filename);
00038
00046     bool loadFromBinary(BST& tree, const string& filename);
00047
00055     bool loadFromText(BST& tree, const string& filename);
00056 };

```

4.7 Dokumentacja pliku C:/Users/micha/Desktop/Drzewo-BST-master/← Drzewo-BST-master/main.cpp

Główny plik programu z interfejsem użytkownika (menu).

```

#include <iostream>
#include <string>
#include <vector>
#include <limits>
#include "BST.h"
#include "FileHandler.h"

```

Funkcje

- void **clearInputBuffer ()**
Czyszcza bufor wejściowy (stdin) w przypadku błędnego wprowadzenia danych.
- void **printMenu ()**
Wyswietla główne menu programu na standardowym wyjściu (konsola).
- int **main ()**
Główna funkcja programu, obsługująca pętle zdarzeń menu.

4.7.1 Opis szczegółowy

Główny plik programu z interfejsem użytkownika (menu).

- Zawiera funkcje **main()** oraz funkcje pomocnicze do obsługi konsolowego menu nawigacyjnego dla drzewa **BST**.

4.7.2 Dokumentacja funkcji

4.7.2.1 clearInputBuffer()

```
void clearInputBuffer ()
```

Czysci bufor wejsciowy (stdin) w przypadku blednego wprowadzenia danych.

- Usuwa flagi bledu strumienia cin i ignoruje wszystkie znaki az do napotkania znaku nowej linii.

4.7.2.2 main()

```
int main ()
```

Glowna funkcja programu, obslugujaca petle zdarzen menu.

Zwroca

0 po pomyslnym zakonczeniu programu.

Skorowidz

```
BST, 5
    FileHandler, 6
        findPath, 6
        insert, 6
        remove, 6

C:/Users/micha/Desktop/Drzewo-BST-master/Drzewo-
    BST-master/BST.cpp, 11
C:/Users/micha/Desktop/Drzewo-BST-master/Drzewo-
    BST-master/BST.h, 11, 12
C:/Users/micha/Desktop/Drzewo-BST-master/Drzewo-
    BST-master/FileHandler.cpp, 13
C:/Users/micha/Desktop/Drzewo-BST-master/Drzewo-
    BST-master/FileHandler.h, 13, 14
C:/Users/micha/Desktop/Drzewo-BST-master/Drzewo-
    BST-master/main.cpp, 14
clearInputBuffer
    main.cpp, 15

FileHandler, 7
    BST, 6
        loadFromBinary, 7
        loadFromText, 7
        saveToBinary, 8
        saveToText, 8
findPath
    BST, 6

insert
    BST, 6

loadFromBinary
    FileHandler, 7
loadFromText
    FileHandler, 7

main
    main.cpp, 15
main.cpp
    clearInputBuffer, 15
    main, 15

remove
    BST, 6

saveToBinary
    FileHandler, 8
saveToText
    FileHandler, 8
```