

Lista dwukierunkowa C++

Wygenerowano za pomocą Doxygen 1.12.0

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	6
3.1.2 Dokumentacja konstruktora i destruktora	6
3.1.2.1 BST()	6
3.1.2.2 ~BST()	6
3.1.3 Dokumentacja funkcji składowych	7
3.1.3.1 BST_dodanie_elementu()	7
3.1.3.2 BST_inorder()	7
3.1.3.3 BST_inorder_wyswietl()	7
3.1.3.4 BST_postorder()	7
3.1.3.5 BST_postorder_wyswietl()	7
3.1.3.6 BST_preorder()	7
3.1.3.7 BST_preorder_wyswietl()	8
3.1.3.8 BST_szukaj()	8
3.1.3.9 BST_szukaj_drogi()	8
3.1.3.10 BST_usun_cale_drzewo()	8
3.1.3.11 BST_usun_cale_drzewo_wyswietl()	9
3.1.3.12 BST_usun_element()	9
3.1.3.13 BST_usuwanie_elementu()	9
3.1.3.14 BST_zapisz_do_pliku()	9
3.1.3.15 BST_zapisz_inorder_do_pliku()	10
3.1.4 Dokumentacja atrybutów składowych	10
3.1.4.1 root	10
3.2 Dokumentacja struktury Node	10
3.2.1 Opis szczegółowy	11
3.2.2 Dokumentacja konstruktora i destruktora	11
3.2.2.1 Node()	11
3.2.3 Dokumentacja atrybutów składowych	11
3.2.3.1 data	11
3.2.3.2 left	11
3.2.3.3 right	12
3.3 Dokumentacja klasy Pliki	12
3.3.1 Opis szczegółowy	12
3.3.2 Dokumentacja funkcji składowych	12
3.3.2.1 Pliki_Wczytanie_z_pliku_binarnego()	12
3.3.2.2 Pliki_Wczytanie_z_pliku_tekstowego()	12

3.3.2.3 Pliki_Zapisanie_wezla()	13
3.3.2.4 Pliki_Zapisywanie_do_pliki_binarnego()	13
4 Dokumentacja plików	15
4.1 Dokumentacja pliku BST.cpp	15
4.2 Dokumentacja pliku BST.h	15
4.3 BST.h	16
4.4 Dokumentacja pliku main.cpp	17
4.4.1 Dokumentacja funkcji	18
4.4.1.1 main()	18
4.5 Dokumentacja pliku Plik.cpp	18
4.6 Dokumentacja pliku Plik.h	19
4.7 Plik.h	20
Skorowidz	21

Rozdział 1

Indeks klas

1.1 Lista klas

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

BST	Klasa reprezentująca drzewo poszukiwan binarnych	5
Node	Struktura reprezentująca wezel w drzewie poszukiwan binarnych	10
Pliki	Klasa do operacji na plikach zwiazanych z drzewem BST	12

Rozdział 2

Indeks plików

2.1 Lista plików

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

BST.cpp	15
BST.h	15
main.cpp	17
Plik.cpp	18
Plik.h	19

Rozdział 3

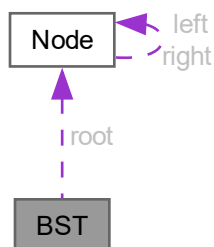
Dokumentacja klas

3.1 Dokumentacja klasy BST

Klasa reprezentująca drzewo poszukiwan binarnych.

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

Diagram współpracy dla BST:



Metody publiczne

- `BST ()`
Wskaznik na korzen drzewa.
- `~BST ()`
Destruktor drzewa `BST`, usuwa wszystkie wezly.
- `void BST_dodanie_elementu (int v)`
Dodaje nowy element do drzewa `BST`.
- `void BST_usuwanie_elementu (int v)`
Usuwa element o podanej wartosci z drzewa.
- `Node * BST_usun_element (Node *node, int v)`
Usuwa element z drzewa zaczynajac od danego wezla.

- `Node * BST_szukaj_drogi (Node *node, int v)`
Szuka sciezki do wezla o podanej wartosci.
- `void BST_szukaj (int v)`
Szuka wezla o podanej wartosci.
- `void BST_preorder (Node *node)`
Wykonuje przejście preorder zaczynając od podanego wezla.
- `void BST_preorder_wyswietl ()`
Wyswietla elementy drzewa w kolejnosci preorder.
- `void BST_inorder (Node *node)`
Wykonuje przejście inorder zaczynając od podanego wezla.
- `void BST_inorder_wyswietl ()`
Wyswietla elementy drzewa w kolejnosci inorder.
- `void BST_postorder (Node *node)`
Wykonuje przejście postorder zaczynając od podanego wezla.
- `void BST_postorder_wyswietl ()`
Wyswietla elementy drzewa w kolejnosci postorder.
- `void BST_usun_cale_drzewo (Node *node)`
Usuwa cale drzewo zaczynając od podanego wezla.
- `void BST_usun_cale_drzewo_wyswietl ()`
Usuwa cale drzewo i zwalnia przydzielona pamiec.
- `void BST_zapisz_do_pliku (const std::string &nazwa_pliku)`
Zapisuje drzewo BST do pliku tekstowego.
- `void BST_zapisz_inorder_do_pliku (Node *node, std::ofstream &plik)`
Zapisuje elementy drzewa w kolejnosci inorder do pliku.

Atrybuty publiczne

- `Node * root`

3.1.1 Opis szczegółowy

Klasa reprezentująca drzewo poszukiwan binarnych.

3.1.2 Dokumentacja konstruktora i destruktora

3.1.2.1 BST()

```
BST::BST ()
```

Wskaznik na korzen drzewa.

Konstruktor tworzący nowe drzewo `BST`.

3.1.2.2 ~BST()

```
BST::~~BST ()
```

Destruktor drzewa `BST`, usuwa wszystkie wezly.

3.1.3 Dokumentacja funkcji składowych

3.1.3.1 BST_dodanie_elementu()

```
void BST::BST_dodanie_elementu (  
    int v)
```

Dodaje nowy element do drzewa [BST](#).

Parametry

<i>v</i>	Wartosc całkowita do dodania.
----------	-------------------------------

3.1.3.2 BST_inorder()

```
void BST::BST_inorder (  
    Node * node)
```

Wykonuje przejście inorder zaczynając od podanego węzła.

Parametry

<i>node</i>	Wskaźnik na węzeł początkowy.
-------------	-------------------------------

3.1.3.3 BST_inorder_wyswietl()

```
void BST::BST_inorder_wyswietl ()
```

Wyswietla elementy drzewa w kolejności inorder.

3.1.3.4 BST_postorder()

```
void BST::BST_postorder (  
    Node * node)
```

Wykonuje przejście postorder zaczynając od podanego węzła.

Parametry

<i>node</i>	Wskaźnik na węzeł początkowy.
-------------	-------------------------------

3.1.3.5 BST_postorder_wyswietl()

```
void BST::BST_postorder_wyswietl ()
```

Wyswietla elementy drzewa w kolejności postorder.

3.1.3.6 BST_preorder()

```
void BST::BST_preorder (  
    Node * node)
```

Wykonuje przejście preorder zaczynając od podanego węzła.

Parametry

<i>node</i>	Wskaźnik na węzeł początkowy.
-------------	-------------------------------

3.1.3.7 BST_preorder_wyswietl()

```
void BST::BST_preorder_wyswietl ()
```

Wyswietla elementy drzewa w kolejności preorder.

3.1.3.8 BST_szukaj()

```
void BST::BST_szukaj (  
    int v)
```

Szuka węzła o podanej wartości.

Parametry

<i>v</i>	Wartość do znalezienia.
----------	-------------------------

3.1.3.9 BST_szukaj_drogi()

```
Node * BST::BST_szukaj_drogi (  
    Node * node,  
    int v)
```

Szuka ścieżki do węzła o podanej wartości.

Parametry

<i>node</i>	Wskaźnik na węzeł, od którego zaczyna się szukanie.
<i>v</i>	Wartość do znalezienia.

Zwraca

Wskaźnik na węzeł o podanej wartości lub nullptr, jeśli nie znaleziono.

3.1.3.10 BST_usun_cale_drzewo()

```
void BST::BST_usun_cale_drzewo (  
    Node * node)
```

Usuwa całe drzewo zaczynając od podanego węzła.

Parametry

<i>node</i>	Wskaźnik na korzeń drzewa lub poddrzewa do usunięcia.
-------------	---

3.1.3.11 BST_usun_cale_drzewo_wyswietl()

```
void BST::BST_usun_cale_drzewo_wyswietl ()
```

Usuwa całe drzewo i zwalnia przydzieloną pamięć.

3.1.3.12 BST_usun_element()

```
Node * BST::BST_usun_element (
    Node * node,
    int v)
```

Usuwa element z drzewa zaczynając od danego węzła.

Parametry

<i>node</i>	Wskaźnik na węzeł, od którego zaczyna się usuwanie.
<i>v</i>	Wartość elementu do usunięcia.

Zwraca

Wskaźnik na korzeń poddrzewa po usunięciu elementu.

3.1.3.13 BST_usuwanie_elementu()

```
void BST::BST_usuwanie_elementu (
    int v)
```

Usuwa element o podanej wartości z drzewa.

Parametry

<i>v</i>	Wartość elementu do usunięcia.
----------	--------------------------------

3.1.3.14 BST_zapisz_do_pliku()

```
void BST::BST_zapisz_do_pliku (
    const std::string & nazwa_pliku)
```

Zapisuje drzewo **BST** do pliku tekstowego.

Parametry

<i>nazwa_pliku</i>	Nazwa pliku, do ktorego zostanie zapisane drzewo.
--------------------	---

3.1.3.15 BST_zapisz_inorder_do_pliku()

```
void BST::BST_zapisz_inorder_do_pliku (
    Node * node,
    std::ofstream & plik)
```

Zapisuje elementy drzewa w kolejnosci inorder do pliku.

Parametry

<i>node</i>	Wskaźnik na aktualny węzeł drzewa.
<i>plik</i>	Referencja do obiektu std::ofstream, do ktorego zostaną zapisane elementy.

3.1.4 Dokumentacja atrybutów składowych

3.1.4.1 root

```
Node* BST::root
```

Dokumentacja dla tej klasy została wygenerowana z plików:

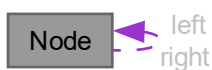
- [BST.h](#)
- [BST.cpp](#)

3.2 Dokumentacja struktury Node

Struktura reprezentująca węzeł w drzewie poszukiwan binarnych.

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

Diagram współpracy dla Node:



Metody publiczne

- `Node` (int v)

Wskaźnik na prawego potomka wezła.

Atrybuty publiczne

- int `data`
- `Node * left`

Zmienna przechowująca wartość wezła.

- `Node * right`

Wskaźnik na lewego potomka wezła.

3.2.1 Opis szczegółowy

Struktura reprezentująca węzeł w drzewie poszukiwań binarnych.

3.2.2 Dokumentacja konstruktora i destruktor

3.2.2.1 Node()

```
Node::Node (
    int v) [inline]
```

Wskaźnik na prawego potomka wezła.

Konstruktor tworzący nowy węzeł.

Parametry

v	Wartość całkowita do zapisania w węźle.
---	---

3.2.3 Dokumentacja atrybutów składowych

3.2.3.1 data

```
int Node::data
```

3.2.3.2 left

```
Node* Node::left
```

Zmienna przechowująca wartość wezła.

3.2.3.3 right

`Node* Node::right`

Wskaźnik na lewego potomka wezła.

Dokumentacja dla tej struktury została wygenerowana z pliku:

- [BST.h](#)

3.3 Dokumentacja klasy Plik

Klasa do operacji na plikach związanych z drzewem [BST](#).

```
#include <Plik.h>
```

Statyczne metody publiczne

- static void [Plik_Wczytanie_z_pliku_tekstowego](#) ([BST](#) &tree, const std::string &filename, bool clearTree)
Wczytuje dane z pliku tekstowego do drzewa [BST](#).
- static void [Plik_Zapisanie_wezla](#) (std::ofstream &file, [Node](#) *node)
Zapisuje wezle drzewa [BST](#) do pliku.
- static void [Plik_Zapisywanie_do_pliku_binarnego](#) ([BST](#) &tree, const std::string &filename)
Zapisuje cale drzewo [BST](#) do pliku binarnego.
- static void [Plik_Wczytanie_z_pliku_binarnego](#) ([BST](#) &tree, const std::string &filename)
Wczytuje dane z pliku binarnego do drzewa [BST](#).

3.3.1 Opis szczegółowy

Klasa do operacji na plikach związanych z drzewem [BST](#).

3.3.2 Dokumentacja funkcji składowych

3.3.2.1 Plik_Wczytanie_z_pliku_binarnego()

```
void Plik::Plik_Wczytanie_z_pliku_binarnego (  
    BST & tree,  
    const std::string & filename) [static]
```

Wczytuje dane z pliku binarnego do drzewa [BST](#).

Parametry

<i>tree</i>	Referencja do drzewa BST , do ktorego zostana wczytane dane.
<i>filename</i>	Nazwa pliku binarnego do wczytania.

3.3.2.2 Plik_Wczytanie_z_pliku_tekstowego()

```
void Plik::Plik_Wczytanie_z_pliku_tekstowego (  
    BST & tree,  
    const std::string & filename,  
    bool clearTree) [static]
```

Wczytuje dane z pliku tekstowego do drzewa [BST](#).

Parametry

<i>tree</i>	Referencja do drzewa BST , do którego zostaną wczytane dane.
<i>filename</i>	Nazwa pliku tekstowego do wczytania.
<i>clearTree</i>	Flaga określająca, czy drzewo powinno być wyczyszczone przed wczytaniem.

3.3.2.3 Plik_Zapisanie_wezla()

```
void Plik::Plik_Zapisanie_wezla (  
    std::ofstream & file,  
    Node * node) [static]
```

Zapisuje węzeł drzewa [BST](#) do pliku.

Parametry

<i>file</i>	Referencja do obiektu <code>std::ofstream</code> , do którego zostanie zapisany węzeł.
<i>node</i>	Wskaznik na węzeł drzewa BST , który ma zostać zapisany.

3.3.2.4 Plik_Zapisywanie_do_pliki_binarnego()

```
void Plik::Plik_Zapisywanie_do_pliki_binarnego (  
    BST & tree,  
    const std::string & filename) [static]
```

Zapisuje całe drzewo [BST](#) do pliku binarnego.

Parametry

<i>tree</i>	Referencja do drzewa BST , które ma zostać zapisane.
<i>filename</i>	Nazwa pliku binarnego, do którego zostanie zapisane drzewo.

Dokumentacja dla tej klasy została wygenerowana z plików:

- [Plik.h](#)
- [Plik.cpp](#)

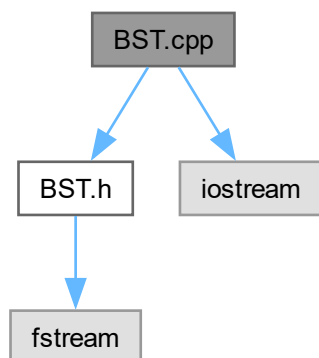
Rozdział 4

Dokumentacja plików

4.1 Dokumentacja pliku BST.cpp

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

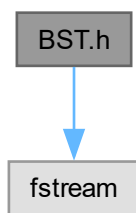
Wykres zależności załączania dla BST.cpp:



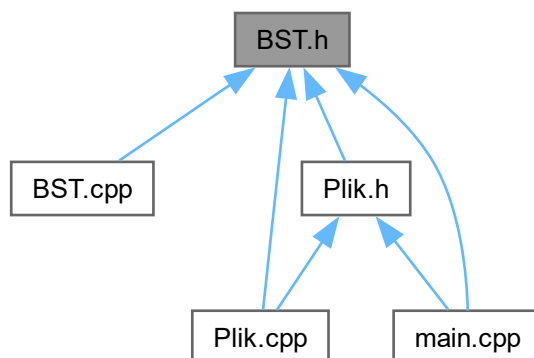
4.2 Dokumentacja pliku BST.h

```
#include <fstream>
```

Wykres zależności załączania dla BST.h:



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



Komponenty

- struct [Node](#)
Struktura reprezentująca węzeł w drzewie poszukiwan binarnych.
- class [BST](#)
Klasa reprezentująca drzewo poszukiwan binarnych.

4.3 BST.h

[Idź do dokumentacji tego pliku.](#)

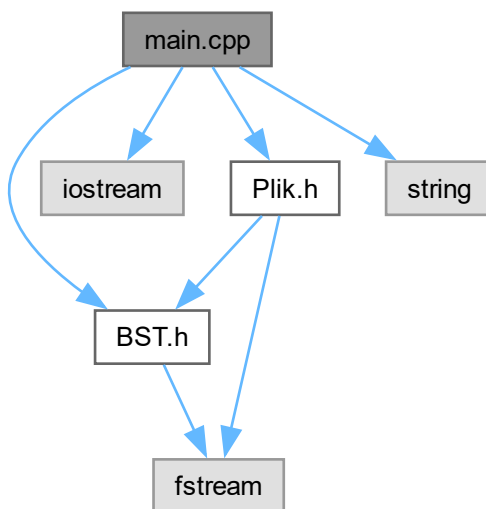
```
00001 #pragma once
00002 #include <fstream>
00003
00006 struct Node {
00007     int data;
00008     Node* left;
```

```
00009     Node* right;
00010
00013     Node(int v) : data(v), left(nullptr), right(nullptr) {}
00014 };
00015
00018 class BST {
00019 public:
00021     Node* root;
00023
00025     BST();
00026
00028     ~BST();
00029
00032     void BST_dodanie_elementu(int v);
00033
00036     void BST_usuwanie_elementu(int v);
00037
00042     Node* BST_usun_element(Node* node, int v);
00043
00048     Node* BST_szukaj_drogi(Node* node, int v);
00049
00052     void BST_szukaj(int v);
00053
00056     void BST_preorder(Node* node);
00057
00059     void BST_preorder_wyswietl();
00060
00063     void BST_inorder(Node* node);
00064
00066     void BST_inorder_wyswietl();
00067
00070     void BST_postorder(Node* node);
00071
00073     void BST_postorder_wyswietl();
00074
00077     void BST_usun_cale_drzewo(Node* node);
00078
00080     void BST_usun_cale_drzewo_wyswietl();
00081
00084     void BST_zapisz_do_pliku(const std::string& nazwa_pliku);
00085
00089     void BST_zapisz_inorder_do_pliku(Node* node, std::ofstream& plik);
00090 };
```

4.4 Dokumentacja pliku main.cpp

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

Wykres zależności załączania dla main.cpp:



Funkcje

- int `main()`

4.4.1 Dokumentacja funkcji

4.4.1.1 `main()`

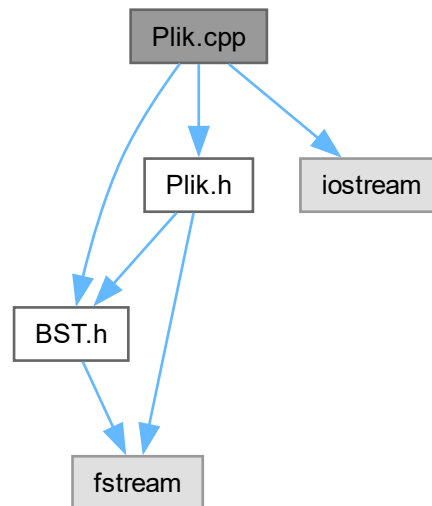
```
int main ()
```

4.5 Dokumentacja pliku Plik.cpp

```
#include "Plik.h"  
#include "BST.h"
```

```
#include <iostream>
```

Wykres zależności załączania dla Plik.cpp:

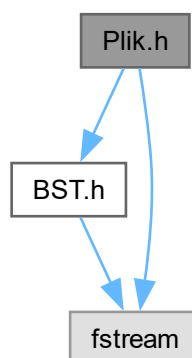


4.6 Dokumentacja pliku Plik.h

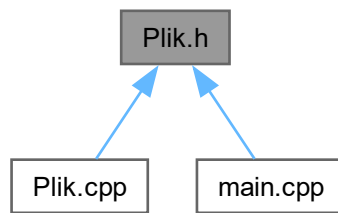
```
#include "BST.h"
```

```
#include <fstream>
```

Wykres zależności załączania dla Plik.h:



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



Komponenty

- class `Pliki`

Klasa do operacji na plikach związanych z drzewem `BST`.

4.7 Plik.h

[Idź do dokumentacji tego pliku.](#)

```
00001 #pragma once
00002
00003 #include "BST.h"
00004 #include <fstream>
00005
00008
00009 class Pliki {
00010
00011 public:
00016     static void Pliki_Wczytanie_z_pliku_tekstowego(BST& tree, const std::string& filename, bool
clearTree);
00017
00021     static void Pliki_Zapisanie_wezla(std::ofstream& file, Node* node);
00022
00026     static void Pliki_Zapisywanie_do_pliku_binarnego(BST& tree, const std::string& filename);
00027
00031     static void Pliki_Wczytanie_z_pliku_binarnego(BST& tree, const std::string& filename);
00032 };
```


Skorowidz

~BST
 BST, [6](#)

BST, [5](#)
 ~BST, [6](#)
 BST, [6](#)
 BST_dodanie_elementu, [7](#)
 BST_inorder, [7](#)
 BST_inorder_wyswietl, [7](#)
 BST_postorder, [7](#)
 BST_postorder_wyswietl, [7](#)
 BST_preorder, [7](#)
 BST_preorder_wyswietl, [8](#)
 BST_szukaj, [8](#)
 BST_szukaj_drogi, [8](#)
 BST_usun_cale_drzewo, [8](#)
 BST_usun_cale_drzewo_wyswietl, [9](#)
 BST_usun_element, [9](#)
 BST_usuwanie_elementu, [9](#)
 BST_zapisz_do_pliku, [9](#)
 BST_zapisz_inorder_do_pliku, [10](#)
 root, [10](#)

BST.cpp, [15](#)
BST.h, [15](#)
BST_dodanie_elementu
 BST, [7](#)
BST_inorder
 BST, [7](#)
BST_inorder_wyswietl
 BST, [7](#)
BST_postorder
 BST, [7](#)
BST_postorder_wyswietl
 BST, [7](#)
BST_preorder
 BST, [7](#)
BST_preorder_wyswietl
 BST, [8](#)
BST_szukaj
 BST, [8](#)
BST_szukaj_drogi
 BST, [8](#)
BST_usun_cale_drzewo
 BST, [8](#)
BST_usun_cale_drzewo_wyswietl
 BST, [9](#)
BST_usun_element
 BST, [9](#)
BST_usuwanie_elementu
 BST, [9](#)

BST_zapisz_do_pliku
 BST, [9](#)
BST_zapisz_inorder_do_pliku
 BST, [10](#)

data
 Node, [11](#)

left
 Node, [11](#)

main
 main.cpp, [18](#)
main.cpp, [17](#)
 main, [18](#)

Node, [10](#)
 data, [11](#)
 left, [11](#)
 Node, [11](#)
 right, [11](#)

Plik.cpp, [18](#)
Plik.h, [19](#)
Pliki, [12](#)
 Pliki_Wczytanie_z_pliku_binarnego, [12](#)
 Pliki_Wczytanie_z_pliku_tekstowego, [12](#)
 Pliki_Zapisanie_wezla, [13](#)
 Pliki_Zapisywanie_do_pliku_binarnego, [13](#)
Pliki_Wczytanie_z_pliku_binarnego
 Pliki, [12](#)
Pliki_Wczytanie_z_pliku_tekstowego
 Pliki, [12](#)
Pliki_Zapisanie_wezla
 Pliki, [13](#)
Pliki_Zapisywanie_do_pliku_binarnego
 Pliki, [13](#)

right
 Node, [11](#)

root
 BST, [10](#)