

My Project

Generated by Doxygen 1.8.17

1 Class Index	1
1.1 Class List	1
2 File Index	3
2.1 File List	3
3 Class Documentation	5
3.1 katalog Struct Reference	5
3.1.1 Detailed Description	5
3.2 plik Struct Reference	6
3.2.1 Detailed Description	6
4 File Documentation	7
4.1 Struktury.h File Reference	7
4.1.1 Macro Definition Documentation	8
4.1.1.1 ARCHIWUM	8
4.1.1.2 catalog	8
4.1.1.3 DATA_FORMAT	9
4.1.1.4 iloscZnakow	9
4.1.1.5 KONIEC	9
4.1.1.6 LiczbaNawigacji	9
4.1.1.7 MAX_SIZE_FILENAME	9
4.1.1.8 NowaSciezka	9
4.1.1.9 wyciety	9
4.1.2 Typedef Documentation	9
4.1.2.1 katalog	9
4.1.2.2 plik	10
4.1.3 Function Documentation	10
4.1.3.1 czas()	10
4.1.3.2 CzytajParametry()	10
4.1.3.3 DodajKatalog()	11
4.1.3.4 DodajPlik()	11
4.1.3.5 Free2DTab()	12
4.1.3.6 Init2DTab()	12
4.1.3.7 MemoryRemove()	12
4.1.3.8 Menu()	13
4.1.3.9 OdczytajPlik()	13
4.1.3.10 OdczytajPlikiKataloguZArchiwum()	13
4.1.3.11 OdczytajStrukture()	14
4.1.3.12 PlikDoKatalogu()	14
4.1.3.13 PobierzArchiwum()	15
4.1.3.14 PoliczPliki()	15
4.1.3.15 PoliczSlova()	15

4.1.3.16 printERROR()	16
4.1.3.17 UsunKatalog()	16
4.1.3.18 UsunPlik()	16
4.1.3.19 UtworzArchiwum()	17
4.1.3.20 WpiszPlikDoArchiwum()	17
4.1.3.21 WyodrebnijPlikNaDysk()	17
4.1.3.22 Wypelnij_Katalog()	18
4.1.3.23 WypelnijArchiwum()	18
4.1.3.24 WypelnijPlik()	18
4.1.3.25 Wyświetl_Plik()	19
4.1.3.26 WyświetlKatalogi()	19
4.1.3.27 Wytnij_Koniec()	19
4.1.3.28 Wytnij_początek()	21
4.1.3.29 ZapiszStruktureWArchiwum()	21
4.1.3.30 ZnajdzKatalog()	22
4.1.3.31 ZnajdzPlik()	22
4.1.3.32 ZwolnijStukture()	22

Index**23**

Chapter 1

Class Index

1.1 Class List

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

katalog	5
plik	6

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

Struktury.h	7
-----------------------------	-------	---

Chapter 3

Class Documentation

3.1 katalog Struct Reference

```
#include <Struktury.h>
```

Public Attributes

- char **nazwa_katalogu** [[MAX_SIZE_FILENAME](#)]
- char * **sciezka**
- struct [plik](#) * **pFile**
- struct [katalog](#) * **pCatalog**
- struct [katalog](#) * **pNextCatalog**

3.1.1 Detailed Description

Struktura dynamiczna opisująca katalog. Jest w postaci drzewa-BST z podwieszonymi listami jednokierunkowymi plików.

See also

typedef struct [plik](#)

Parameters

<i>nazwa_katalogu</i>	tablica znaków przechowująca nazwę katalogu, ograniczona do 261 znaków
<i>sciezka</i>	wskaźnik na tablicę dynamiczną znaków, która przechowuje lokalizację katalogu

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

- [Struktury.h](#)

3.2 plik Struct Reference

```
#include <Struktury.h>
```

Public Attributes

- char **nazwa_pliku** [[MAX_SIZE_FILENAME](#)]
- char **data_modyfikacji** [[DATA_FORMAT](#)]
- unsigned int **rozmiar_pliku**
- char * **tresc_pliku**
- struct [plik](#) * **pNextPlik**

3.2.1 Detailed Description

Struktura dynamiczna opisująca plik. Jest w postaci listy jednokierunkowej.

Parameters

<i>nazwa_pliku</i>	tablica znaków przechowująca nazwę pliku, ograniczona do 261 znaków
<i>data_modyfikacji</i>	tablica znaków przechowująca datę zarchiwizowania pliku, ograniczona do 20 znaków
<i>rozmiar_pliku</i>	przechowuje liczbę znaków odczytanych z pliku
<i>tresc_pliku</i>	wskaźnik na tablicę dynamiczną znaków, która przechowuje zawartość odczytanego pliku

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

- [Struktury.h](#)

Chapter 4

File Documentation

4.1 Struktury.h File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdbool.h>
#include <time.h>
```

Classes

- struct [plik](#)
- struct [katalog](#)

Macros

- #define [_CRT_SECURE_NO_WARNINGS](#)
- #define [MAX_SIZE_FILENAME](#) 261
- #define [DATA_FORMAT](#) 20
- #define [LiczbaNawigacji](#) 3
- #define [ARCHIWUM](#) "Archiwum"
- #define [KONIEC](#) "KONIEC"
- #define [catalog](#) 2
- #define [NowaSciezka](#) 0
- #define [wyciety](#) 1
- #define [iloscZnakow](#) 1

Typedefs

- typedef FILE * [file](#)
- typedef struct [plik](#) [plik](#)
- typedef struct [katalog](#) [katalog](#)

Functions

- void [czas](#) (char *godzina)
- size_t [PoliczSlowa](#) (char *s1)
- void [Wytnij_poczatek](#) (char *sciezka, char **wyrazWydzielony, char **nowaSciezka)
- void [Wytnij_Koniec](#) (char *Sciezka, char **wyrazWyciety)
- void [Init2DTab](#) (char ***Dynamic, int size)
- void [Free2DTab](#) (char ***Dynamic, int size)
- int [PoliczPliki](#) (katalog *root)
- void [printERROR](#) ()
- plik * [WypelnijPlik](#) (plik *pNowy, const char *filename, char *data_modyfikacji, unsigned int rozmiar, char *tresc)
- bool [DodajPlik](#) (katalog **root, const char *filename, char *data_modyfikacji, unsigned int rozmiar, char *tresc)
- int [OdczytajPlik](#) (katalog **root, const char *filename, char *Sciezka)
- int [UsunPlik](#) (plik **head, const char *filename)
- void [Wyswietl_Plik](#) (plik *head, int wciecie)
- plik * [ZnajdzPlik](#) (katalog *root, const char *filename)
- bool [WyodrebnijPlikNaDysk](#) (katalog **root, const char *filename)
- void [Wypelnij_Katalog](#) (katalog **root, const char *catalogName, char *Sciezka)
- void [DodajKatalog](#) (katalog **root, char *Sciezka)
- void [WyswietlKatalogi](#) (katalog *root, int wciecie)
- void [ZwolnijStukture](#) (katalog **root)
- int [UsunKatalog](#) (katalog **root, char *Sciezka)
- katalog * [ZnajdzKatalog](#) (katalog *root, char *Sciezka)
- bool [PlikDoKatalogu](#) (katalog **root, const char *filename, char *data_modyfikacji, unsigned int rozmiar, char *tresc, char *Sciezka)
- void [WpiszPlikDoArchiwum](#) (plik *head, file pfile, int liczbaPlikow)
- void [WypelnijArchiwum](#) (katalog *root, file pfile)
- bool [ZapiszStruktureWArchiwum](#) (katalog *root)
- void [OdczytajPlikiKataloguZArchiwum](#) (katalog **root, file pfile)
- bool [OdczytajStrukture](#) (katalog **root, file pFile)
- bool [PobierzArchiwum](#) (katalog **NowyRoot)
- void [Menu](#) (katalog **root)
- bool [CzytajParametry](#) (int argc, char **argv, char **Path, char ***FileNames)
- bool [UtworzArchiwum](#) ()
- void [MemoryRemove](#) (katalog **NowyRoot, char *Path, char **FileNames, int argc)

4.1.1 Macro Definition Documentation

4.1.1.1 ARCHIWUM

```
#define ARCHIWUM "Archiwum"
```

Łańcuch znakowy określający nazwę archiwum, plik binarny przechowujący zarchiwizowane dane.

4.1.1.2 katalog

```
#define katalog 2
```

Określa indeks w tablicy słów pod którym znajduje się ostatni wyraz ścieżki.

4.1.1.3 DATA_FORMAT

```
#define DATA_FORMAT 20
```

Wielkość tablicy dynamicznej przechowującej format daty.

4.1.1.4 iloscZnakow

```
#define iloscZnakow 1
```

Określa liczbę znaków jaka ma zostać wpisana lub odczytana do lub z pliku.

4.1.1.5 KONIEC

```
#define KONIEC "KONIEC"
```

Łańcuch znakowy określający wyraz kończący wydzielanie ścieżek katalogów.

4.1.1.6 LiczbaNawigacji

```
#define LiczbaNawigacji 3
```

Określa liczbę słów do zainicjalizowania w tablicy słów, potrzebnej do operowania na wydzielaniu ścieżek katalogu.

4.1.1.7 MAX_SIZE_FILENAME

```
#define MAX_SIZE_FILENAME 261
```

Określa maksymalną liczbę znaków nazw plików i katalogów.

4.1.1.8 NowaSciezka

```
#define NowaSciezka 0
```

Określa indeks w tablicy słów pod którym znajduje się reszta ścieżki po wydzieleniu.

4.1.1.9 wyciety

```
#define wyciety 1
```

Określa indeks w tablicy słów pod którym znajduje się pierwszy wyraz ścieżki.

4.1.2 Typedef Documentation

4.1.2.1 katalog

```
typedef struct katalog katalog
```

Struktura dynamiczna opisująca katalog. Jest w postaci drzewa-BST z podwieszonymi listami jednokierunkowymi plików.

See also

`typedef struct plik`

Parameters

<i>nazwa_katalogu</i>	tablica znaków przechowująca nazwę katalogu, ograniczona do 261 znaków
<i>sciezka</i>	wskaźnik na tablicę dynamiczną znaków, która przechowuje lokalizację katalogu

4.1.2.2 plik

```
typedef struct plik plik
```

Struktura dynamiczna opisująca plik. Jest w postaci listy jednokierunkowej.

Parameters

<i>nazwa_pliku</i>	tablica znaków przechowująca nazwę pliku, ograniczona do 261 znaków
<i>data_modyfikacji</i>	tablica znaków przechowująca datę zarchiwizowania pliku, ograniczona do 20 znaków
<i>rozmiar_pliku</i>	przechowuje liczbę znaków odczytanych z pliku
<i>tresc_pliku</i>	wskaźnik na tablicę dynamiczną znaków, która przechowuje zawartość odczytanego pliku

4.1.3 Function Documentation**4.1.3.1 czas()**

```
void czas (
    char * godzina )
```

Funkcja odczytuje czas, który upłynął od 1970r. i na tej podstawie określa datę, w której został dodany plik do archiwum.

Parameters

<i>godzina</i>	C-string, w którym jest zapisywany format pobranego czasu
----------------	---

4.1.3.2 CzytajParametry()

```
bool CzytajParametry (
    int argc,
    char ** argv,
    char ** Path,
    char *** FileNames )
```

Funkcja dostaje parametry jakimi są przełączniki oraz nazwy plików z lini poleceń.

Parameters

<i>argc</i>	ilość parametrów z lini poleceń
<i>argv</i>	tablica ciągu znaków z lini poleceń
<i>Path</i>	zwraca oddzielony fragment z argv reprezentujący lokalizację katalogu
<i>FileNames</i>	zwraca nazwy plików wydzielone z argv

Returns

zwraca prawdę jeśli funkcja została wykonana prawidłowo lub zwraca fałsz, gdy funkcja natrafiła na błędne dane

4.1.3.3 DodajKatalog()

```
void DodajKatalog (
    katalog ** root,
    char * Sciezka )
```

Funkcja dodaje element do drzewa katalogów względem lokalizacji określonej w ścieżce.

Parameters

<i>root</i>	wskaźnik na oryginalny wskaźnik struktury katalogów drzewa-BST
<i>Sciezka</i>	C-string zawierający lokalizację katalogu

4.1.3.4 DodajPlik()

```
bool DodajPlik (
    katalog ** root,
    const char * filename,
    char * data_modyfikacji,
    unsigned int rozmiar,
    char * tresc )
```

Funkcja dodaje nowy element do listy plików w sposób posortowany względem nazwy pliku

Parameters

<i>root</i>	wskaźnik na oryginalny wskaźnik struktury katalogów drzewa-BST
<i>filename</i>	C-string zawierający nazwę pliku
<i>data_modyfikacji</i>	C-string zawierający datę odczytania pliku
<i>rozmiar</i>	przechowuje liczbę elementów przeczytanych z pliku
<i>tresc</i>	C-string przechowujący zawartość odczytanego pliku

Returns

zwraca prawdę, gdy element został dodany poprawnie w przeciwnym wypadku zwraca fałsz

4.1.3.5 Free2DTab()

```
void Free2DTab (
    char *** Dynamic,
    int size )
```

Funkcja zwalnia pamięć tablicy słów.

Parameters

<i>Dynamic</i>	wskaźnik na dwuwymiarową tablicę
<i>size</i>	liczba zawierająca jak wiele pamięci zostanie zwolnione

4.1.3.6 Init2DTab()

```
void Init2DTab (
    char *** Dynamic,
    int size )
```

Funkcja inicjalizuje tablice słów.

Parameters

<i>Dynamic</i>	wskaźnik na dwuwymiarową tablicę
<i>size</i>	liczba zawierająca jak wiele miejsca na słowa ma zostać zainicjalizowane

4.1.3.7 MemoryRemove()

```
void MemoryRemove (
    katalog ** NowyRoot,
    char * Path,
    char ** FileNames,
    int argc )
```

Funkcja zbiorczo zwalnia pamięć struktury, tablicy słów oraz C-string w funkcji głównej main.

Parameters

<i>NowyRoot</i>	wskaźnik na oryginalny wskaźnik struktury katalogów drzewa-BST
<i>FileNames</i>	przyjmuje tablicę słów, zawierającą nazwy plików
<i>argc</i>	ilość parametrów z lini poleceń

4.1.3.8 Menu()

```
void Menu (
    katalog ** root )
```

Funkcja główna programu. Wyświetla na ekran zawartość struktury i podaje użytkownikowi opcje wyboru do wykonania na danych zawartych w strukturze oraz je wykonuje przy pomocy funkcji pomocniczych.

Parameters

<i>root</i>	wskaźnik na oryginalny wskaźnik struktury katalogów drzewa-BST
-------------	--

4.1.3.9 OdczytajPlik()

```
int OdczytajPlik (
    katalog ** root,
    const char * filename,
    char * Sciezka )
```

Funkcja odczytuje plik do zarchiwizowania.

Parameters

<i>root</i>	wskaźnik na oryginalny wskaźnik struktury katalogów drzewa-BST
<i>filename</i>	C-string zawierający nazwę pliku
<i>Sciezka</i>	lokalizacja, w której ma zostać umieszczony odczytany plik

Returns

zwraca 0 w przypadku, gdy plik do odczytu nie istnieje, -1 gdy nie został dodany poprawnie, 1 gdy plik został odczytany i dodany poprawnie

4.1.3.10 OdczytajPlikiKataloguZArchiwum()

```
void OdczytajPlikiKataloguZArchiwum (
    katalog ** root,
    file prfile )
```

Funkcja odczytuje informacje o zarchiwizowanych plikach w pliku binarnym(archiwum) i dodaje element do struktury

Parameters

<i>root</i>	wskaźnik na oryginalny wskaźnik struktury katalogów drzewa-BST
<i>prfile</i>	wskaźnik na plik binarny(archiwum)

4.1.3.11 OdczytajStrukture()

```
bool OdczytajStrukture (
    katalog ** root,
    file pFile )
```

Funkcja odczytuje informacje o zarchiwizowanych katalogach i plikach w nich zawartych. Na tej podstawie wypełnia strukture elementami.

Parameters

<i>root</i>	wskaźnik na oryginalny wskaźnik struktury katalogów drzewa-BST
<i>pFile</i>	wskaźnik na plik binarny(archiwum)

Returns

zwraca prawdę, gdy element zostanie odczytany w przeciwnym wypadku fałsz

4.1.3.12 PlikDoKatalogu()

```
bool PlikDoKatalogu (
    katalog ** root,
    const char * filename,
    char * data_modyfikacji,
    unsigned int rozmiar,
    char * tresc,
    char * Sciezka )
```

Funkcja dodaje plik do katalogu na podstawie ścieżki.

Parameters

<i>root</i>	wskaźnik na oryginalny wskaźnik struktury katalogów drzewa-BST
<i>filename</i>	C-string zawierający nazwę pliku
<i>data_modyfikacji</i>	C-string zawierający datę odczytania pliku
<i>rozmiar</i>	przechowuje liczbę elementów sczytanych z pliku
<i>tresc</i>	C-string przechowujący zawartość odczytanego pliku
<i>Sciezka</i>	C-string zawierający lokalizację katalogu

Returns

zwraca - prawdę, gdy plik został dodany poprawnie do katalogu. Fałsz - katalog nie został znaleziony

4.1.3.13 PobierzArchiwum()

```
bool PobierzArchiwum (
    katalog ** NowyRoot )
```

Funkcja otwiera archiwum i wykonuje OdczytajStrukture.

See also

[OdczytajStrukture\(katalog **root, file pFile\)](#)

Parameters

<i>NowyRoot</i>	wskaźnik na oryginalny wskaźnik struktury katalogów drzewa-BST
-----------------	--

Returns

zwraca prawdę, gdy wykonana się poprawnie, w przypadku braku archiwum lub pustego archiwum zwraca fałsz

4.1.3.14 PoliczPliki()

```
int PoliczPliki (
    katalog * root )
```

Funkcja zliczająca liczbę plików w danym węźle drzewa-BST.

Parameters

<i>root</i>	wskaźnik na korzeń drzewa katalogów
-------------	-------------------------------------

Returns

zwraca liczbę znalezionych plików w węźle

4.1.3.15 PoliczSlowa()

```
size_t PoliczSlowa (
    char * s1 )
```

Funkcja zlicza ilość słów w ścieżce.

Parameters

<i>s1</i>	C-string, docelowo ma przyjmować ścieżkę do katalogu
-----------	--

Returns

zwraca liczbę słów

4.1.3.16 printERROR()

```
void printERROR ( )
```

Funkcja wypisuje na ekran informacje o błędzie.

4.1.3.17 UsunKatalog()

```
int UsunKatalog (
    katalog ** root,
    char * Sciezka )
```

Funkcja usuwa z pamięci węzeł drzewa(katalog) oraz wszystkie lewe poddrzewa(katalogi w nim umieszczone) i doczepione listy plików. Katalog docelowy jest usuwany na podstawie podanej ścieżki.

Parameters

<i>root</i>	wskaźnik na oryginalny wskaźnik struktury katalogów drzewa-BST
<i>Sciezka</i>	C-string zawierający lokalizację katalogu

Returns

zwraca wartość 1, gdy znaleziono katalog do usunięcia, wartość 0 gdy element nie został znaleziony

4.1.3.18 UsunPlik()

```
int UsunPlik (
    plik ** head,
    const char * filename )
```

Funkcja usuwa plik ze listy plików na podstawie nazwy pliku.

Parameters

<i>head</i>	wskaźnik na oryginalny wskaźnik listy plików
<i>filename</i>	C-string zawierający nazwę pliku

Returns

zwraca 1 - w przypadku poprawnego wykonania funkcji, 0 - gdy usunięcie pliku nie było możliwe

4.1.3.19 UtworzArchiwum()

```
bool UtworzArchiwum ( )
```

Funkcja tworzy plik binarny archiwum.

Returns

zwraca prawdę kiedy archiwum zostało utworzone lub fałsz gdy archiwum już istniało

4.1.3.20 WpiszPlikDoArchiwum()

```
void WpiszPlikDoArchiwum (
    plik * head,
    file pfile,
    int liczbaPlikow )
```

Funkcja wpisuje do pliku binarnego(archiwum) zawartość listy plików.

Parameters

<i>head</i>	wskaźnik na listę plików
<i>pfile</i>	wskaźnik na plik binarny
<i>liczbaPlikow</i>	przechowuje liczbę plików do wpisania

4.1.3.21 WyodrebnijPlikNaDysk()

```
bool WyodrebnijPlikNaDysk (
    katalog ** root,
    const char * filename )
```

Funkcja na podstawie nazwy wyodrębnia plik na dysk z archiwum.

Parameters

<i>root</i>	wskaźnik na oryginalny wskaźnik struktury katalogów drzewa-BST
<i>filename</i>	C-string zawierający nazwę pliku

Returns

zwraca prawdę - gdy funkcja zostanie poprawnie wykonana, w momencie gdy nie ma pliku do wyodrębnienia
- fałsz

4.1.3.22 Wypelnij_Katalog()

```
void Wypelnij_Katalog (
    katalog ** root,
    const char * catalogName,
    char * Sciezka )
```

Funkcja wypełnia element struktury katalogów.

Parameters

<i>root</i>	wskaźnik na oryginalny wskaźnik struktury katalogów drzewa-BST
<i>catalogName</i>	C-string zawierający nazwę katalogu
<i>Sciezka</i>	C-string zawierający lokalizację katalogu

4.1.3.23 WypelnijArchiwum()

```
void WypelnijArchiwum (
    katalog * root,
    file pfile )
```

Funkcja wpisuje do pliku binarnego(archiwum) zawartość całej struktury, wszystkie drzewa katalogów i listy plików.

Parameters

<i>root</i>	wskaźnik na drzewo katalogów
<i>pfile</i>	wskaźnik na plik binarny

4.1.3.24 WypelnijPlik()

```
plik* WypelnijPlik (
    plik * pNowy,
    const char * filename,
    char * data_modyfikacji,
    unsigned int rozmiar,
    char * tresc )
```

Funkcja wypełnia strukture pliku danymi.

Parameters

<i>pNowy</i>	wskaźnik na liste plików w danym węźle drzewa-BST
<i>filename</i>	C-string zawierający nazwę pliku

Parameters

<i>data_modyfikacji</i>	C-string zawierający datę odczytania pliku
<i>rozmiar</i>	przechowuje liczbę elementów sczytanych z pliku
<i>tresc</i>	C-string przechowujący zawartość odczytanego pliku

Returns

zwraca wskaźnik na utworzony element pliku

4.1.3.25 Wyszwietl_Plik()

```
void Wyszwietl_Plik (
    plik * head,
    int wciecie )
```

Funkcja wyświetla informacje zawarte liście plików.

Parameters

<i>head</i>	oryginalny wskaźnik na listę plików
<i>wciecie</i>	liczba określająca z jak wielkim wcięciem mają zostać wypisane informacje

4.1.3.26 WyszwietlKatalogi()

```
void WyszwietlKatalogi (
    katalog * root,
    int wciecie )
```

Funkcja wypisuje na ekran wszystkie informacje zawarte w całej strukturze.

Parameters

<i>root</i>	wskaźnik na drzewo katalogów
<i>wciecie</i>	liczba określająca z jak wielkim wcięciem mają zostać wypisane informacje

4.1.3.27 Wytnij_Koniec()

```
void Wytnij_Koniec (
    char * Sciezka,
    char ** wyrazWyciety )
```

Funkcja wydziela z podawanej ścieżki wyraz ostatni, nie modyfikując oryginału.

Parameters

<i>Sciezka</i>	C-string, z którego wydzielany jest ostatni wyraz
<i>wyrazWyciety</i>	wskaźnik na C-string, do którego jest wpisywany wydzielony wyraz

4.1.3.28 Wytnij_poczatek()

```
void Wytnij_poczatek (
    char * sciezka,
    char ** wyrazWydzielony,
    char ** nowaSciezka )
```

Funkcja dzieli podawaną ścieżkę na pierwszy wyraz oraz resztę ścieżki nie modyfikując oryginału.

Parameters

<i>sciezka</i>	C-string, z którego wycinany jest wyraz
<i>wyrazWydzielony</i>	wskaźnik na C-string, do którego wpisywany jest wyraz wycięty
<i>nowaSciezka</i>	wskaźnik na C-string, do którego jest wpisywana reszta ścieżki

4.1.3.29 ZapiszStrukturaWArchiwum()

```
bool ZapiszStrukturaWArchiwum (
    katalog * root )
```

Funkcja otwiera plik binarny i wykonuje funkcje WypelnijArchiwum.

See also

[WypelnijArchiwum\(katalog *root, file pfile\)](#)

Parameters

<i>root</i>	wskaźnik na drzewo katalogów
-------------	------------------------------

Returns

zwraca prawdę jeśli funkcja wykonała się poprawnie, fałsz - nie istnieje struktura do wpisania

4.1.3.30 ZnajdzKatalog()

```
katalog* ZnajdzKatalog (
    katalog * root,
    char * Sciezka )
```

Funkcja wyszukuje katalog na podstawie podanej ścieżki do katalogu.

Parameters

<i>root</i>	wskaźnik na drzewo katalogów
<i>Sciezka</i>	C-string zawierający lokalizację katalogu

Returns

zwraca wskaźnik na znaleziony katalog

4.1.3.31 ZnajdzPlik()

```
plik* ZnajdzPlik (
    katalog * root,
    const char * filename )
```

Funkcja znajduje plik w liście plików na podstawie nazwy.

Parameters

<i>root</i>	wskaźnik na drzewo katalogów
<i>filename</i>	C-string zawierający nazwę pliku

Returns

zwraca wskaźnik na znaleziony element

4.1.3.32 ZwolnijStukture()

```
void ZwolnijStukture (
    katalog ** root )
```

Funkcja usuwa z pamięci całą strukturę.

Parameters

<i>root</i>	wskaźnik na oryginalny wskaźnik struktury katalogów drzewa-BST
-------------	--

Index

ARCHIWUM

Struktury.h, [8](#)

catalog

Struktury.h, [8](#)

czas

Struktury.h, [10](#)

CzytajParametry

Struktury.h, [10](#)

DATA_FORMAT

Struktury.h, [8](#)

DodajKatalog

Struktury.h, [11](#)

DodajPlik

Struktury.h, [11](#)

Free2DTab

Struktury.h, [12](#)

iloscZnakow

Struktury.h, [9](#)

Init2DTab

Struktury.h, [12](#)

katalog, [5](#)

Struktury.h, [9](#)

KONIEC

Struktury.h, [9](#)

LiczbaNawigacji

Struktury.h, [9](#)

MAX_SIZE_FILENAME

Struktury.h, [9](#)

MemoryRemove

Struktury.h, [12](#)

Menu

Struktury.h, [13](#)

NowaSciezka

Struktury.h, [9](#)

OdczytajPlik

Struktury.h, [13](#)

OdczytajPlikiKataloguZArchiwum

Struktury.h, [13](#)

OdczytajStrukture

Struktury.h, [14](#)

plik, [6](#)

Struktury.h, [10](#)

PlikDoKatalogu

Struktury.h, [14](#)

PobierzArchiwum

Struktury.h, [14](#)

PoliczPliki

Struktury.h, [15](#)

PoliczSlova

Struktury.h, [15](#)

printERROR

Struktury.h, [16](#)

Struktury.h, [7](#)

ARCHIWUM, [8](#)

catalog, [8](#)

czas, [10](#)

CzytajParametry, [10](#)

DATA_FORMAT, [8](#)

DodajKatalog, [11](#)

DodajPlik, [11](#)

Free2DTab, [12](#)

iloscZnakow, [9](#)

Init2DTab, [12](#)

katalog, [9](#)

KONIEC, [9](#)

LiczbaNawigacji, [9](#)

MAX_SIZE_FILENAME, [9](#)

MemoryRemove, [12](#)

Menu, [13](#)

NowaSciezka, [9](#)

OdczytajPlik, [13](#)

OdczytajPlikiKataloguZArchiwum, [13](#)

OdczytajStrukture, [14](#)

plik, [10](#)

PlikDoKatalogu, [14](#)

PobierzArchiwum, [14](#)

PoliczPliki, [15](#)

PoliczSlova, [15](#)

printERROR, [16](#)

UsunKatalog, [16](#)

UsunPlik, [16](#)

UtworzArchiwum, [17](#)

WpiszPlikDoArchiwum, [17](#)

wyciety, [9](#)

WyodrebnijPlikNaDysk, [17](#)

Wypelnij_Katalog, [18](#)

WypelnijArchiwum, [18](#)

WypelnijPlik, [18](#)

Wyswietl_Plik, [19](#)

WyswietlKatalogi, [19](#)

Wytnij_Koniec, [19](#)
Wytnij_poczatek, [21](#)
ZapiszStruktureWArchiwum, [21](#)
ZnajdzKatalog, [21](#)
ZnajdzPlik, [22](#)
ZwolnijStukture, [22](#)

UsunKatalog
 Struktury.h, [16](#)
UsunPlik
 Struktury.h, [16](#)
UtworzArchiwum
 Struktury.h, [17](#)

WpiszPlikDoArchiwum
 Struktury.h, [17](#)
wyciety
 Struktury.h, [9](#)
WyodrebnijPlikNaDysk
 Struktury.h, [17](#)
Wypelnij_Katalog
 Struktury.h, [18](#)
WypelnijArchiwum
 Struktury.h, [18](#)
WypelnijPlik
 Struktury.h, [18](#)
Wyswietl_Plik
 Struktury.h, [19](#)
WyswietlKatalogi
 Struktury.h, [19](#)
Wytnij_Koniec
 Struktury.h, [19](#)
Wytnij_poczatek
 Struktury.h, [21](#)

ZapiszStruktureWArchiwum
 Struktury.h, [21](#)
ZnajdzKatalog
 Struktury.h, [21](#)
ZnajdzPlik
 Struktury.h, [22](#)
ZwolnijStukture
 Struktury.h, [22](#)