

Benchmark

1.1.a

Generated by Doxygen 1.8.4

Mon Mar 17 2014 16:17:55



# Contents

<b>1</b>	<b>Benchmark</b>	<b>1</b>
1.1	Najważniejsze cechy	1
1.2	Opis algorytmu	1
<b>2</b>	<b>Class Index</b>	<b>3</b>
2.1	Class List	3
<b>3</b>	<b>File Index</b>	<b>5</b>
3.1	File List	5
<b>4</b>	<b>Class Documentation</b>	<b>7</b>
4.1	benchmark Class Reference	7
4.1.1	Detailed Description	7
4.1.2	Constructor & Destructor Documentation	8
4.1.2.1	benchmark	8
4.1.3	Member Function Documentation	8
4.1.3.1	czas_start	8
4.1.3.2	czas_stop	8
4.1.3.3	ile_czasu	8
4.1.3.4	wykonaj_algorytm	8
4.1.4	Member Data Documentation	9
4.1.4.1	czas	10
4.1.4.2	elapsedTime	10
4.1.4.3	t1	10
4.1.4.4	t2	10
4.2	element Class Reference	10
4.2.1	Detailed Description	11
4.2.2	Member Data Documentation	11
4.2.2.1	klucz	11
4.2.2.2	nastepny	11
4.3	elementk Class Reference	11
4.3.1	Detailed Description	11

4.3.2	Member Data Documentation . . . . .	12
4.3.2.1	klucz . . . . .	12
4.3.2.2	nastepny . . . . .	12
4.4	kolejka_lista Class Reference . . . . .	12
4.4.1	Detailed Description . . . . .	13
4.4.2	Constructor & Destructor Documentation . . . . .	13
4.4.2.1	kolejka_lista . . . . .	13
4.4.2.2	~kolejka_lista . . . . .	13
4.4.3	Member Function Documentation . . . . .	14
4.4.3.1	dodaj_element . . . . .	14
4.4.3.2	isempty . . . . .	15
4.4.3.3	pokaz_elementy . . . . .	15
4.4.3.4	pop . . . . .	15
4.4.3.5	push . . . . .	16
4.4.3.6	size . . . . .	16
4.4.3.7	wczytaj_dane . . . . .	16
4.4.3.8	wczytaj_dane . . . . .	17
4.4.4	Member Data Documentation . . . . .	17
4.4.4.1	licznik . . . . .	17
4.4.4.2	przod . . . . .	17
4.4.4.3	tyl . . . . .	18
4.5	kolejka_tablica Class Reference . . . . .	18
4.5.1	Detailed Description . . . . .	19
4.5.2	Constructor & Destructor Documentation . . . . .	19
4.5.2.1	kolejka_tablica . . . . .	19
4.5.2.2	~kolejka_tablica . . . . .	19
4.5.3	Member Function Documentation . . . . .	19
4.5.3.1	isEmpty . . . . .	19
4.5.3.2	pokaz_elementy . . . . .	19
4.5.3.3	pop . . . . .	19
4.5.3.4	push . . . . .	20
4.5.3.5	pushx2 . . . . .	20
4.5.3.6	size . . . . .	21
4.5.3.7	wczytaj_dane . . . . .	21
4.5.3.8	wczytaj_dane . . . . .	21
4.5.3.9	wczytaj_danex2 . . . . .	22
4.5.4	Member Data Documentation . . . . .	22
4.5.4.1	dane . . . . .	22
4.5.4.2	danetmp . . . . .	23
4.5.4.3	rozmiar . . . . .	23

4.5.4.4	spr	23
4.6	stos_lista Class Reference	23
4.6.1	Detailed Description	24
4.6.2	Constructor & Destructor Documentation	24
4.6.2.1	stos_lista	24
4.6.2.2	~stos_lista	24
4.6.3	Member Function Documentation	25
4.6.3.1	dodaj_element	25
4.6.3.2	isempty	25
4.6.3.3	pokaz_elementy	25
4.6.3.4	pop	25
4.6.3.5	push	25
4.6.3.6	size	26
4.6.3.7	wczytaj_dane	26
4.6.3.8	wczytaj_dane	26
4.6.4	Member Data Documentation	27
4.6.4.1	licznik	27
4.6.4.2	przod	27
4.6.4.3	tyl	27
4.7	stos_tablica Class Reference	27
4.7.1	Detailed Description	28
4.7.2	Constructor & Destructor Documentation	28
4.7.2.1	stos_tablica	28
4.7.2.2	~stos_tablica	28
4.7.3	Member Function Documentation	28
4.7.3.1	isEmpty	28
4.7.3.2	pokaz_elementy	29
4.7.3.3	pop	29
4.7.3.4	push	29
4.7.3.5	pushx2	29
4.7.3.6	size	29
4.7.3.7	wczytaj_dane	30
4.7.3.8	wczytaj_dane	30
4.7.3.9	wczytaj_danex2	30
4.7.4	Member Data Documentation	30
4.7.4.1	dane	31
4.7.4.2	danetmp	31
4.7.4.3	rozmiar	31
4.7.4.4	spr	31

<b>5</b>	<b>File Documentation</b>	<b>33</b>
5.1	/home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/doc/pages/strona.dox File Reference . . .	33
5.2	/home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/inc/benchmark.hh File Reference . . . . .	33
5.2.1	Detailed Description . . . . .	34
5.3	/home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/inc/kolejka_lista.hh File Reference . . . . .	34
5.3.1	Detailed Description . . . . .	35
5.4	/home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/inc/kolejka_tablica.hh File Reference . . . . .	36
5.4.1	Detailed Description . . . . .	37
5.5	/home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/inc/stos_lista.hh File Reference . . . . .	37
5.5.1	Detailed Description . . . . .	38
5.6	/home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/inc/stos_tablica.hh File Reference . . . . .	38
5.6.1	Detailed Description . . . . .	39
5.7	/home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/src/benchmark.cpp File Reference . . . . .	39
5.7.1	Detailed Description . . . . .	40
5.8	/home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/src/kolejka_lista.cpp File Reference . . . . .	40
5.9	/home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/src/kolejka_tablica.cpp File Reference . . . . .	41
5.10	/home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/src/main.cpp File Reference . . . . .	41
5.10.1	Detailed Description . . . . .	42
5.10.2	Function Documentation . . . . .	42
5.10.2.1	main . . . . .	42
5.11	/home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/src/stos_lista.cpp File Reference . . . . .	42
5.12	/home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/src/stos_tablica.cpp File Reference . . . . .	43
<b>Index</b>		<b>44</b>

# Chapter 1

## Benchmark

### Author

Pawel Zurek

### Date

16.03.2014

### Version

1.1.b

Program umożliwia liczenie czasu trwania operacji wypełniania liczbami stosu / kolejki.

### 1.1 Najważniejsze cechy

Program służy do liczenia czasu wypełnienia liczbami stosów i kolejek za pomocą :

- > list
- > tablic ( po przekroczeniu rozmiaru tablicy, rozmiar powiększany o jeden )
- > tablic ( po przekroczeniu rozmiaru tablicy, rozmiar zwiększany dwukrotnie )

### 1.2 Opis algorytmu

Algorytm w tym zadaniu to 6 petli :

- > wypełnienie stosu za pomocą listy
- > wypełnienie kolejki za pomocą listy
- > wypełnienie stosu za pomocą tablicy ( rozmiar o jeden )
- > wypełnienie stosu za pomocą tablicy ( rozmiar x2 )
- > wypełnienie kolejki za pomocą tablicy ( rozmiar o jeden )
- > wypełnienie kolejki za pomocą tablicy ( rozmiar x2 )

,na którą składa się wypełnienie następującą ilością elementów:

- > 10
- > 100

-> 1000

-> 10000

Czasy sa wyprowadzane na standartowe wyjscie.



## Chapter 2

# Class Index

### 2.1 Class List

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

<a href="#">benchmark</a>	Modeluje pojecie Benchmark . . . . .	7
<a href="#">element</a>	Modeluje pojecie element . . . . .	10
<a href="#">elementk</a>	Modeluje pojecie elementk (zmieniona nazwa, poniewaz <a href="#">stos_lista</a> tez ma klase o nazwie element)	11
<a href="#">kolejka_lista</a>	Modeluje pojecie Kolejka . . . . .	12
<a href="#">kolejka_tablica</a>	Modeluje pojecie Kolejka . . . . .	18
<a href="#">stos_lista</a>	Modeluje pojecie Stos . . . . .	23
<a href="#">stos_tablica</a>	Modeluje pojecie Stos . . . . .	27



## Chapter 3

# File Index

### 3.1 File List

Here is a list of all files with brief descriptions:

/home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/inc/ <a href="#">benchmark.hh</a>	
Definicje funkcji dla klasy benchmark . . . . .	33
/home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/inc/ <a href="#">kolejka_lista.hh</a>	
Definicje funkcji dla klasy Kolejka zdefiniowanej lista . . . . .	34
/home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/inc/ <a href="#">kolejka_tablica.hh</a>	
Definicje funkcji dla klasy Kolejka zdefiniowanej tablica . . . . .	36
/home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/inc/ <a href="#">stos_lista.hh</a>	
Definicje funkcji dla klasy Stos zdefiniowanej lista . . . . .	37
/home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/inc/ <a href="#">stos_tablica.hh</a>	
Definicje funkcji dla klasy Stos zdefiniowanej tablica . . . . .	38
/home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/src/ <a href="#">benchmark.cpp</a>	
Plik zawiera funkcje z klasy benchmark . . . . .	39
/home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/src/ <a href="#">kolejka_lista.cpp</a>	40
/home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/src/ <a href="#">kolejka_tablica.cpp</a>	41
/home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/src/ <a href="#">main.cpp</a>	
Plik zawiera funkcje <a href="#">main()</a> . . . . .	41
/home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/src/ <a href="#">stos_lista.cpp</a>	42
/home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/src/ <a href="#">stos_tablica.cpp</a>	43



## Chapter 4

# Class Documentation

### 4.1 benchmark Class Reference

Modeluje pojecie Benchmark.

```
#include <benchmark.hh>
```

#### Public Member Functions

- [benchmark](#) ()  
*Konstruktor klasy Benchmark.*
- void [wykonaj\\_algorytm](#) ()  
*Funkcja wykonujaca algorytm.*
- void [czas\\_start](#) ()  
*Funkcja pomocnicza mierzenia czasu.*
- void [czas\\_stop](#) ()  
*Funkcja pomocnicza mierzenia czasu.*
- double [ile\\_czasu](#) ()  
*Funkcja obliczania czasu dzialania programu.*

#### Private Attributes

- double [elapsedTime](#)  
*Pole typu double, bedzie uzywane do mierzenia czasu dzialania pojedynczego wypelniania.*
- double [czas](#)  
*Pole typu double, bedzie uzywane do mierzenia calkowitego czasu dzialania programu.*
- timeval [t1](#)  
*Pole typu timeval, pomoc do liczenia czasu dzialania operacji krotkich ( tzn pojedynczego dzialania)*
- timeval [t2](#)

#### 4.1.1 Detailed Description

Modeluje pojecie Benchmark.

Klasa sluzy do przeprowadzenia Benchmarku programu, tzn : -> wczytania 4 zestawu danych o ilosci elementow :

-> 10

-> 100

-> 1000

-> 10000

Na czterech klasach :

-> [stos\\_lista](#)

-> [stos\\_tablica](#)

-> [kolejka\\_lista](#)

-> [kolejka\\_tablica](#)

-> Na koniec zostana wyswietlone czasy kazdej akcji z osobna

Definition at line 42 of file benchmark.hh.

## 4.1.2 Constructor & Destructor Documentation

### 4.1.2.1 `benchmark::benchmark( )` `[inline]`

Konstruktor klasy Benchmark.

Konstruktor jest bezparametryczny, inicjalizuje wszystkie skladowe klasy wartosciami zerowymi.

Definition at line 68 of file benchmark.hh.

## 4.1.3 Member Function Documentation

### 4.1.3.1 `void benchmark::czas_start( )`

Funkcja pomocnicza mierzenia czasu.

Funkcja zaczyna liczyc czas od momentu wywolania tej metody Sluzy do liczenia czasu wykonywania pojedynczego wypelniania stosu/kolejki

Definition at line 14 of file benchmark.cpp.

### 4.1.3.2 `void benchmark::czas_stop( )`

Funkcja pomocnicza mierzenia czasu.

Funkcja konczy liczyc czas od momentu wywolania tej metody Sluzy do liczenia czasu wykonywania pojedynczego wypelniania stosu/kolejki

Definition at line 17 of file benchmark.cpp.

### 4.1.3.3 `double benchmark::ile_czasu( )`

Funkcja obliczania czasu dzialania programu.

Funkcja podaje czas wykonywania pojedynczego wypelniania stosu/kolejki

#### Returns

elapsedTime -> zmienna typu double ( wynik obliczen )

Definition at line 20 of file benchmark.cpp.

### 4.1.3.4 `void benchmark::wykonaj_algorytm( )`

Funkcja wykonujaca algorytm.

Działanie funkcji:

-> stworzenie obiektow klas :

\*stos\_lista

\*stos\_tablica

\*kolejka\_lista

\*kolejka\_tablica

-> wypelniana jest wartosciami z plikow z :

\* 10 elementami

\* 100 elementami

\* 1000 elementami

\* 10000 elementami

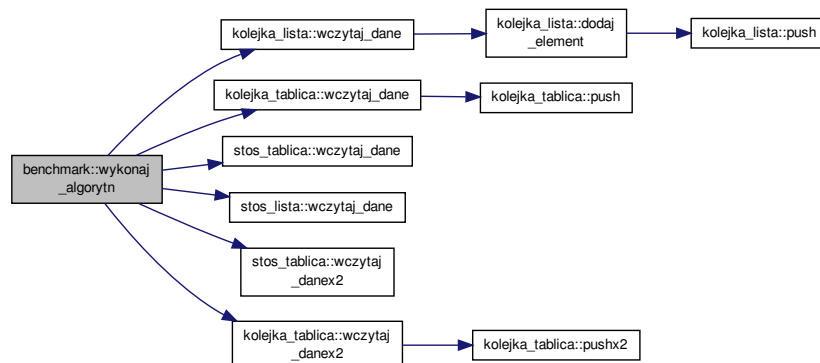
-> wyswietlenie wynikow :

- czasu dzialania po kolei kazdego wypelnienia z osobna

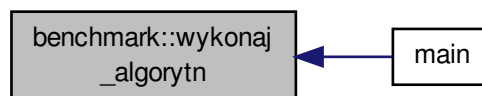
- czasu dzialania calego algorytmu

Definition at line 27 of file benchmark.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



#### 4.1.4 Member Data Documentation

#### 4.1.4.1 `double benchmark::czas` [private]

Pole typu `double`, będzie używane do mierzenia całkowitego czasu działania programu.

Definition at line 52 of file `benchmark.hh`.

#### 4.1.4.2 `double benchmark::elapsedTime` [private]

Pole typu `double`, będzie używane do mierzenia czasu działania pojedynczego wypełniania.

Definition at line 48 of file `benchmark.hh`.

#### 4.1.4.3 `timeval benchmark::t1` [private]

Pole typu `timeval`, pomoc do liczenia czasu działania operacji krótkich ( tzn pojedynczego działania)

Definition at line 56 of file `benchmark.hh`.

#### 4.1.4.4 `timeval benchmark::t2` [private]

Definition at line 56 of file `benchmark.hh`.

The documentation for this class was generated from the following files:

- `/home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/inc/benchmark.hh`
- `/home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/src/benchmark.cpp`

## 4.2 `element` Class Reference

Modeluje pojecie `element`.

```
#include <stos_lista.hh>
```

Collaboration diagram for `element`:



### Public Attributes

- `element * nastepny`

*Pole typu `element*`, wskaźnik używany do łączenia elementów między sobą w stos.*

- `int klucz`

*Pole typu `int`, przechowuje informacje o wartości danego elementu.*



### 4.2.1 Detailed Description

Modeluje pojecie element.

Element jest klasa, ktora bedzie wpisywana do stosu Klasa ta zawiera informacje o danej wartosci elementu ( klucz ) oraz wskaznik na nastepny element.

Definition at line 29 of file stos\_lista.hh.

### 4.2.2 Member Data Documentation

#### 4.2.2.1 int element::klucz

Pole typu int, przechowuje informacje o wartosci danego elementu.

Definition at line 39 of file stos\_lista.hh.

#### 4.2.2.2 element\* element::nastepny

Pole typu element\*, wskaznik uzywany do laczenia elementow miedzy soba w stos.

Definition at line 35 of file stos\_lista.hh.

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

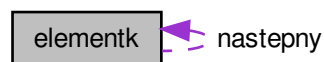
- [/home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/inc/stos\\_lista.hh](#)

## 4.3 elementk Class Reference

Modeluje pojecie elementk (zmieniona nazwa, poniewaz [stos\\_lista](#) tez ma klase o nazwie element).

```
#include <kolejka_lista.hh>
```

Collaboration diagram for elementk:



### Public Attributes

- [elementk](#) \* [nastepny](#)  
*Pole typu element\*, wskaznik uzywany do laczenia elementow miedzy soba w stos.*
- int [klucz](#)  
*Pole typu int, przechowuje informacje o wartosci danego elementu.*

### 4.3.1 Detailed Description

Modeluje pojecie elementk (zmieniona nazwa, poniewaz [stos\\_lista](#) tez ma klase o nazwie element).

Element jest klasa, która będzie wpisywana do stosu. Klasa ta zawiera informacje o danej wartości elementu (klucz) oraz wskaźnik na następny element.

Definition at line 30 of file kolejka\_lista.hh.

### 4.3.2 Member Data Documentation

#### 4.3.2.1 int elementk::klucz

Pole typu int, przechowuje informacje o wartości danego elementu.

Definition at line 40 of file kolejka\_lista.hh.

#### 4.3.2.2 elementk\* elementk::nastepny

Pole typu elementk\*, wskaźnik używany do łączenia elementów między sobą w stos.

Definition at line 36 of file kolejka\_lista.hh.

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

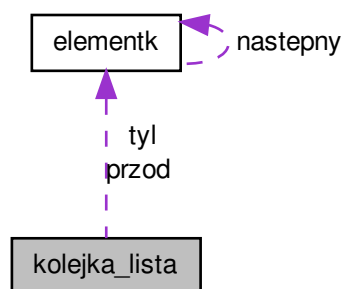
- [/home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/inc/kolejka\\_lista.hh](/home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/inc/kolejka_lista.hh)

## 4.4 kolejka\_lista Class Reference

Modeluje pojęcie Kolejka.

```
#include <kolejka_lista.hh>
```

Collaboration diagram for kolejka\_lista:



### Public Member Functions

- [kolejka\\_lista \(\)](#)  
*Konstruktor klasy kolejka.*
- [~kolejka\\_lista \(\)](#)  
*Destruktor klasy Kolejka.*
- void [wczytaj\\_dane \(\)](#)  
*Funkcja wczytywania.*

- void `wczytaj_dane` (string nazwa)  
*Funkcja wczytywania.*
- void `pokaz_elementy` ()  
*Funkcja wyswietlajaca.*
- bool `isempty` ()  
*Funkcja sprawdzania pustosci kolejki.*
- unsigned `size` ()  
*Funkcja sprawdzania rozmiar kolejki.*
- `elementk` \* `push` (`elementk` \*p)  
*Funkcja dodajaca element.*
- void `dodaj_element` (int a)  
*Funkcja dodajaca element.*
- `elementk` \* `pop` ()  
*Funkcja usuwajaca ostatni element ze stosu.*

### Private Attributes

- `elementk` \* `przod`  
*Pole typu element, wskazuje na pierwszy element listy.*
- `elementk` \* `tyl`  
*Pole typu element, wskazuje na ostatni element listy.*
- unsigned `licznik`  
*Pole typu int, bedzie uzywane jako rozmiar kolejki.*

#### 4.4.1 Detailed Description

Modeluje pojecie Kolejka.

Stos jest klasa zawierajaca dynamicznie zaalokowane liste

Definition at line 52 of file `kolejka_lista.hh`.

#### 4.4.2 Constructor & Destructor Documentation

##### 4.4.2.1 `kolejka_lista::kolejka_lista ( )` `[inline]`

Konstruktor klasy kolejka.

Konstruktor jest bezparametryczny, inicjalizuje wszystkie skladowe klasy wartosciami zerowymi.

Definition at line 78 of file `kolejka_lista.hh`.

##### 4.4.2.2 `kolejka_lista::~~kolejka_lista ( )` `[inline]`

Destruktor klasy Kolejka.

Usuwa dynamicznie zaalokowana liste

Definition at line 85 of file `kolejka_lista.hh`.

### 4.4.3 Member Function Documentation

#### 4.4.3.1 void kolejka\_lista::dodaj\_element ( int a )

Funkcja dodająca element.

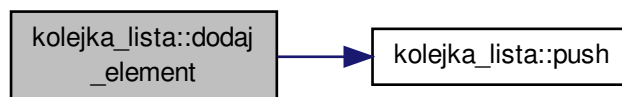
Funkcja dodaje do stosu element podany przez użytkownika poprzez wywołanie funkcji [push\(\)](#).

## Parameters

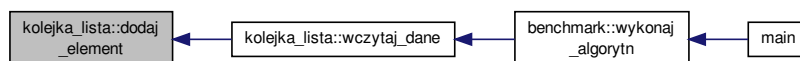
<code>a</code>	- pole typu int przechowuje wartosc jaka ma byc wpisana
----------------	---

Definition at line 61 of file kolejka\_lista.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



#### 4.4.3.2 bool kolejka\_lista::isempty ( )

Funkcja sprawdzania pustosci kolejki.

Funkcja sprawdza czy kolejka jest pusta

## Returns

1 -> Kolejka pusta 0 -> kolejka nie pusta

Definition at line 6 of file kolejka\_lista.cpp.

#### 4.4.3.3 void kolejka\_lista::pokaz\_elementy ( )

Funkcja wyswietlajaca.

Funkcja wyswietla aktualny stan kolejki

Definition at line 44 of file kolejka\_lista.cpp.

#### 4.4.3.4 elementk \* kolejka\_lista::pop ( )

Funkcja usuwajaca ostatni element ze stosu.

Funkcja usuwa z listy ostatni element i zwraca jego adres.

Jeśli lista jest pusta, to pole tyl zawiera NULL. W takim przypadku zwracamy NULL i kończymy.

W przeciwnym razie zapamiętujemy adres ostatniego elementu listy w p. Jeśli lista zawiera tylko jeden element , to pola przod i tyl zawierają ten sam adres. W takim przypadku lista staje się pusta po odłączeniu ostatniego elementu, dlatego wpisujemy w nie adres pusty NULL.

Jeśli lista zawiera więcej niż jeden element, to przechodzimy kolejno przez wszystkie elementy listy ustawiając pole `tyl` na adres poprzednika ostatniego elementu. W poprzedniku zerujemy następnik.

Zmniejszamy licznik i zwracamy zapamiętany w `p` adres ostatniego elementu.

Definition at line 28 of file `kolejka_lista.cpp`.

#### 4.4.3.5 `elementk * kolejka_lista::push ( elementk * p )`

Funkcja dodająca element.

Funkcja wstawia nowy element `p` na koniec kolejki i zwraca jego adres.

Jeśli lista zawiera jakieś elementy, to `tyl` zawsze wskazuje ostatni element listy. W pole `nastepny` ostatniego elementu wpisujemy w takim przypadku adres wstawianego elementu. W efekcie zostanie on dołączony do końca listy. W polu `nastepny` nowego elementu umieszczamy adres zerowy `NULL`, ponieważ jest on teraz ostatnim elementem i nie posiada następnika.

Lista mogła być pusta. W takim przypadku wstawiony element jest jednocześnie pierwszym i ostatnim. Dlatego do pola `przod` wpisujemy adres pobrany z pola `tyl`.

Po wstawieniu elementu zwiększamy licznik i zwracamy adres końca kolejki.

##### Parameters

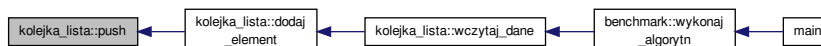
<code>p</code>	- pole typu <code>element</code> przechowuje wartość jaką ma być wpisana
----------------	--

##### Returns

adres nowego elementu.

Definition at line 19 of file `kolejka_lista.cpp`.

Here is the caller graph for this function:



#### 4.4.3.6 `unsigned kolejka_lista::size ( )`

Funkcja sprawdzania rozmiar kolejki.

Funkcja podaje aktualny rozmiar kolejki

##### Returns

rozmiar -> rozmiar kolejki

Definition at line 14 of file `kolejka_lista.cpp`.

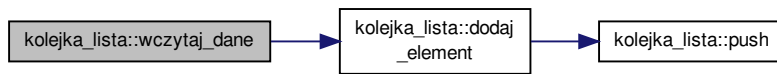
#### 4.4.3.7 `void kolejka_lista::wczytaj_dane ( )`

Funkcja wczytywania.

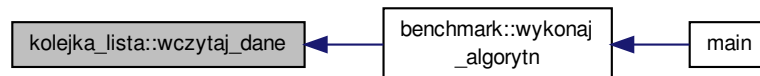
Funkcja wczytuje wartości do stosu z podanego pliku przez użytkownika.

Definition at line 68 of file `kolejka_lista.cpp`.

Here is the call graph for this function:



Here is the caller graph for this function:



#### 4.4.3.8 void kolejka\_lista::wczytaj\_dane ( string nazwa )

Funkcja wczytywania.

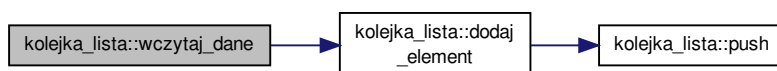
Funkcja wczytuje wartosci do tabeli po przez wpisanie nazwy jako argument metody Wykorzystuje metode push jako funkcje wpisujaca do kolejki

##### Parameters

<i>nazwa</i>	-> zmienna typu string, przechowuje nazwe otwieranego pliku
--------------	---

Definition at line 85 of file kolejka\_lista.cpp.

Here is the call graph for this function:



### 4.4.4 Member Data Documentation

#### 4.4.4.1 unsigned kolejka\_lista::licznik [private]

Pole typu int, bedzie uzywane jako rozmiar kolejki.

Definition at line 67 of file kolejka\_lista.hh.

#### 4.4.4.2 elementk\* kolejka\_lista::przod [private]

Pole typu element, wskazuje na pierwszy element listy.

Definition at line 58 of file kolejka\_lista.hh.

#### 4.4.4.3 elementk\* kolejka\_lista::tyl [private]

Pole typu element, wskazuje na ostatni element listy.

Definition at line 62 of file kolejka\_lista.hh.

The documentation for this class was generated from the following files:

- [/home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/inc/kolejka\\_lista.hh](#)
- [/home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/src/kolejka\\_lista.cpp](#)

## 4.5 kolejka\_tablica Class Reference

Modeluje pojecie Kolejka.

```
#include <kolejka_tablica.hh>
```

### Public Member Functions

- [kolejka\\_tablica \(\)](#)  
*Konstruktor klasy kolejka.*
- [~kolejka\\_tablica \(\)](#)  
*Destruktor klasy kolejka.*
- void [wczytaj\\_dane \(\)](#)  
*Funkcja wczytywania.*
- void [wczytaj\\_dane \(string nazwa\)](#)  
*Funkcja wczytywania.*
- void [wczytaj\\_danex2 \(string nazwa\)](#)  
*Funkcja wczytywania.*
- void [pokaz\\_elementy \(\)](#)  
*Funkcja wyswietlajaca.*
- bool [isEmpty \(\)](#)  
*Funkcja sprawdzania pustosci kolejki.*
- unsigned [size \(\)](#)  
*Funkcja sprawdzania rozmiar kolejki.*
- void [push \(int element\)](#)  
*Funkcja dodajaca element.*
- void [pushx2 \(int element\)](#)  
*Funkcja dodajaca element.*
- void [pop \(\)](#)  
*Funkcja usuwajaca ostatni element ze kolejki.*

### Private Attributes

- int \* [dane](#)  
*Pole typu int, bedzie uzywane jako kolejki z danymi.*
- int \* [danetmp](#)  
*Pole typu int, bedzie uzywane jako kolejko z danymi sprawdzajacymi, pomocniczymi.*
- int [rozmiar](#)  
*Pole typu int, bedzie uzywane jako rozmiar tabeli.*
- int [spr](#)  
*Pole typu int, bedzie uzywane jako pomocnicza wartosc jako rozmiar kolejki.*



### 4.5.1 Detailed Description

Modeluje pojecie Kolejka.

Kolejka\_tab jest klasa zawierajaca dynamicznie zaalokowana tablice i tablice pomocnicza przy dodawaniu i usuwaniu elementow z kolejki.

Definition at line 29 of file kolejka\_tablica.hh.

### 4.5.2 Constructor & Destructor Documentation

#### 4.5.2.1 kolejka\_tablica::kolejka\_tablica ( ) [inline]

Konstruktor klasy kolejka.

Konstruktor jest bezparametryczny, inicjalizuje wszystkie skladowe klasy wartosciami zerowymi.

Definition at line 59 of file kolejka\_tablica.hh.

#### 4.5.2.2 kolejka\_tablica::~~kolejka\_tablica ( ) [inline]

Destruktor klasy kolejka.

Usuwa dynamicznie zaalokawana tablice

Definition at line 66 of file kolejka\_tablica.hh.

### 4.5.3 Member Function Documentation

#### 4.5.3.1 bool kolejka\_tablica::isEmpty ( )

Funkcja sprawdzania pustosci kolejki.

Funkcja sprawdza czy kolejka jest pusty

#### Returns

1 -> Kolejka pusta 0 -> Kolejka nie pusta

Definition at line 4 of file kolejka\_tablica.cpp.

#### 4.5.3.2 void kolejka\_tablica::pokaz\_elementy ( )

Funkcja wyswietlajaca.

Funkcja wyswietla aktualny stan kolejki

Definition at line 94 of file kolejka\_tablica.cpp.

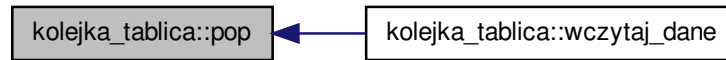
#### 4.5.3.3 void kolejka\_tablica::pop ( )

Funkcja usuwajaca ostatni element ze kolejki.

Funkcja usuwa ostatni element znajdujacy sie na kolejce oraz zmniejsza jego rozmiar o jeden.

Definition at line 75 of file kolejka\_tablica.cpp.

Here is the caller graph for this function:



#### 4.5.3.4 void kolejka\_tablica::push ( int *element* )

Funkcja dodająca element.

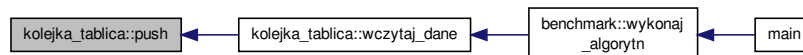
Funkcja dodaje do stosu element podany przez uzytkownika Funkcja dodaje element poprzez zwiekszenie rozmiaru tablicy o jeden.

##### Parameters

<i>element</i>	- pole typu int przechowuje wartosc jaka ma byc wpisana
----------------	---

Definition at line 17 of file kolejka\_tablica.cpp.

Here is the caller graph for this function:



#### 4.5.3.5 void kolejka\_tablica::pushx2 ( int *element* )

Funkcja dodająca element.

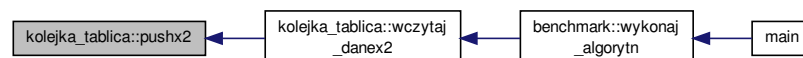
Funkcja dodaje do stosu element podany przez uzytkownika Funkcja dodaje element poprzez zwiekszenie rozmiaru tablicy o dwa razy niz aktualny stan.

##### Parameters

<i>element</i>	- pole typu int przechowuje wartosc jaka ma byc wpisana
----------------	---

Definition at line 44 of file kolejka\_tablica.cpp.

Here is the caller graph for this function:



## 4.5.3.6 unsigned kolejka\_tablica::size ( )

Funkcja sprawdzania rozmiar kolejki.

Funkcja podaje aktualny rozmiar kolejki

## Returns

rozmiar -> rozmiar kolejki

Definition at line 12 of file kolejka\_tablica.cpp.

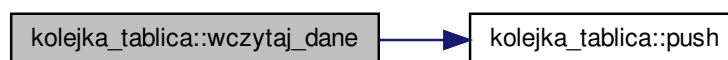
## 4.5.3.7 void kolejka\_tablica::wczytaj\_dane ( )

Funkcja wczytywania.

Funkcja wczytuje wartosci do tabeli z podanego pliku przez uzytkownika.

Definition at line 107 of file kolejka\_tablica.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



## 4.5.3.8 void kolejka\_tablica::wczytaj\_dane ( string nazwa )

Funkcja wczytywania.

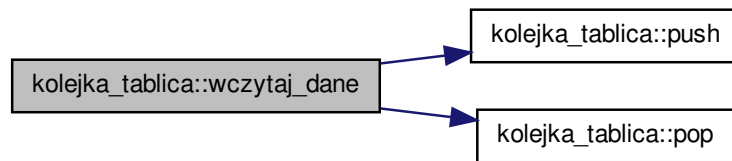
Funkcja wczytuje wartosci do tabeli po przez wpisanie nazwy jako argument metody Wykorzystuje metode push jako funkcje wpisujaca do kolejki

## Parameters

<i><b>nazwa</b></i>	-> zmienna typu string, przechowuje nazwe otwieranego pliku
---------------------	---

Definition at line 124 of file kolejka\_tablica.cpp.

Here is the call graph for this function:



#### 4.5.3.9 void kolejka\_tablica::wczytaj\_danex2 ( string nazwa )

Funkcja wczytywania.

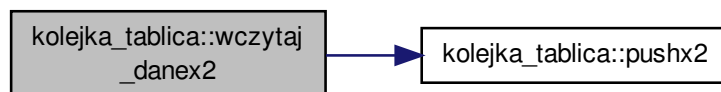
Funkcja wczytuje wartosci do tabeli po przez wpisanie nazwy jako argument metody Wykorzystuje metode pushx2 jako funkcje wpisujaca do kolejki

##### Parameters

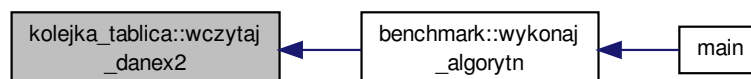
<i><code>nazwa</code></i>	-> zmienna typu string, przechowuje nazwe otwieranego pliku
---------------------------	---

Definition at line 140 of file `kolejka_tablica.cpp`.

Here is the call graph for this function:



Here is the caller graph for this function:



## 4.5.4 Member Data Documentation

### 4.5.4.1 int\* kolejka\_tablica::dane [private]

Pole typu int, bedzie uzywane jako kolejki z danymi.

Definition at line 35 of file kolejka\_tablica.hh.

#### 4.5.4.2 `int* kolejka_tablica::danetmp` `[private]`

Pole typu `int`, bedzie uzywane jako kolejko z danymi sprawdzajacymi, pomocniczymi.

Definition at line 39 of file kolejka\_tablica.hh.

#### 4.5.4.3 `int kolejka_tablica::rozmiar` `[private]`

Pole typu `int`, bedzie uzywane jako rozmiar tabeli.

Definition at line 44 of file kolejka\_tablica.hh.

#### 4.5.4.4 `int kolejka_tablica::spr` `[private]`

Pole typu `int`, bedzie uzywane jako pomocnicza wartosc jako rozmiar kolejki.

Definition at line 48 of file kolejka\_tablica.hh.

The documentation for this class was generated from the following files:

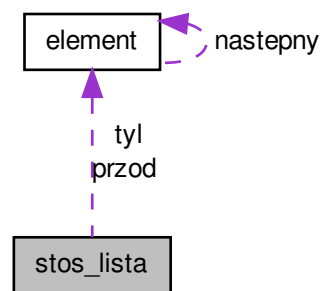
- [/home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/inc/kolejka\\_tablica.hh](#)
- [/home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/src/kolejka\\_tablica.cpp](#)

## 4.6 stos\_lista Class Reference

Modeluje pojecie Stos.

```
#include <stos_lista.hh>
```

Collaboration diagram for `stos_lista`:



### Public Member Functions

- [stos\\_lista](#) ()  
*Konstruktor klasy stos.*
- [~stos\\_lista](#) ()

- Destruktor klasy stos.*
- void [wczytaj\\_dane](#) ()  
*Funkcja wczytywania.*
- void [wczytaj\\_dane](#) (string nazwa)  
*Funkcja wczytywania.*
- void [pokaz\\_elementy](#) ()  
*Funkcja wyswietlajaca.*
- bool [isempty](#) ()  
*Funkcja sprawdzania pustosci stosu.*
- unsigned [size](#) ()  
*Funkcja sprawdzania rozmiar stosu.*
- [element](#) \* [push](#) ([element](#) \*p)  
*Funkcja dodajaca element.*
- void [dodaj\\_element](#) (int a)  
*Funkcja dodajaca element.*
- [element](#) \* [pop](#) ()  
*Funkcja usuwajaca ostatni element ze stosu.*

## Private Attributes

- [element](#) \* [przod](#)  
*Pole typu element, wskazuje na pierwszy element listy.*
- [element](#) \* [tyl](#)  
*Pole typu element, wskazuje na ostatni element listy.*
- unsigned [licznik](#)  
*Pole typu int, bedzie uzywane jako rozmiar stosu.*

## 4.6.1 Detailed Description

Modeluje pojecie Stos.

Stos jest klasa zawierajaca dynamicznie zaalokowane lista

Definition at line 53 of file `stos_lista.hh`.

## 4.6.2 Constructor & Destructor Documentation

### 4.6.2.1 `stos_lista::stos_lista ( )` `[inline]`

Konstruktor klasy stos.

Konstruktor jest bezparametryczny, inicjalizuje wszystkie składowe klasy wartosciami zerowymi.

Definition at line 79 of file `stos_lista.hh`.

### 4.6.2.2 `stos_lista::~~stos_lista ( )` `[inline]`

Destruktor klasy stos.

Usuwa dynamicznie zaalokowana liste

Definition at line 86 of file `stos_lista.hh`.

### 4.6.3 Member Function Documentation

#### 4.6.3.1 void stos\_lista::dodaj\_element ( int a )

Funkcja dodająca element.

Funkcja dodaje do stosu element podany przez użytkownika poprzez wywołanie funkcji [push\(\)](#).

##### Parameters

a	- pole typu int przechowuje wartosc jaka ma byc wpisana
---	---

Definition at line 61 of file stos\_lista.cpp.

#### 4.6.3.2 bool stos\_lista::isempty ( )

Funkcja sprawdzania pustosci stosu.

Funkcja sprawdza czy stos jest pusty

##### Returns

- 1 -> Stos pusty
- 0 -> Stos nie pusty

Definition at line 7 of file stos\_lista.cpp.

#### 4.6.3.3 void stos\_lista::pokaz\_elementy ( )

Funkcja wyswietlajaca.

Funkcja wyswietla aktualny stan stosu

Definition at line 44 of file stos\_lista.cpp.

#### 4.6.3.4 element \* stos\_lista::pop ( )

Funkcja usuwajaca ostatni element ze stosu.

Funkcja usuwa z listy pierwszy element i zwraca jego adres. Element zostaje jedynie odłączony od listy, lecz wciąż zajmuje swój obszar pamięci.

Na początku sprawdzamy, czy lista zawiera jakiś element. Jeśli nie, zwracamy NULL. W przeciwnym razie zapamiętujemy w zmiennej p adres pierwszego elementu. W polu przod umieszczamy adres następnego elementu listy. W ten sposób element p zostaje odłączony logicznie od listy.

Jeśli po odłączeniu pole przod zawiera adres NULL, to lista jest obecnie pusta i NULL wpisujemy również do pola tyl.

Zmniejszamy o jeden licznik licznik i zwracamy adres usuniętego z listy elementu.

Definition at line 28 of file stos\_lista.cpp.

#### 4.6.3.5 element \* stos\_lista::push ( element \* p )

Funkcja dodająca element.

Funkcja wstawia nowy element p na początek listy i zwraca jego adres.

Najpierw wpisuje w pole następnego elementu p adres pierwszego elementu listy. W efekcie pierwszy element listy staje się elementem następnym dla p. W polu przod zostaje umieszczony adres elementu p, czyli staje się on pierwszym elementem na liście.

Lista mogła być pusta. Jeśli pole `tyl` zawiera `NULL`, to wstawiony element jest jedynym jak dotąd elementem listy. Zatem jest on zarówno elementem pierwszym jak i ostatnim. Dlatego w polu `tyl` umieszczamy w tym przypadku adres pierwszego elementu pobrany z pola `przod`.

Po wstawieniu elementu zwiększamy licznik `licznik` i zwracamy adres pierwszego elementu listy.

#### Parameters

<i>p</i>	- pole typu <code>element</code> przechowuje wartość jaką ma być wpisana
----------	--

#### Returns

adres nowego elementu.

Definition at line 20 of file `stos_lista.cpp`.

#### 4.6.3.6 unsigned `stos_lista::size` ( )

Funkcja sprawdzania rozmiar stosu.

Funkcja podaje aktualny rozmiar stosu

#### Returns

`rozmiar` -> rozmiar stosu

Definition at line 15 of file `stos_lista.cpp`.

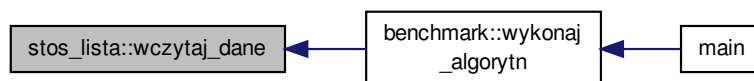
#### 4.6.3.7 void `stos_lista::wczytaj_dane` ( )

Funkcja wczytywania.

Funkcja wczytuje wartości do stosu z podanego pliku przez użytkownika.

Definition at line 68 of file `stos_lista.cpp`.

Here is the caller graph for this function:



#### 4.6.3.8 void `stos_lista::wczytaj_dane` ( string *nazwa* )

Funkcja wczytywania.

Funkcja wczytuje wartości do tabeli po przez wpisanie nazwy jako argument metody. Wykorzystuje metodę `push` jako funkcję wpisywania do stosu.



## Parameters

<i>nazwa</i>	-> zmienna typu string, przechowuje nazwe otwieranego pliku
--------------	---

Definition at line 85 of file stos\_lista.cpp.

#### 4.6.4 Member Data Documentation

##### 4.6.4.1 unsigned stos\_lista::licznik [private]

Pole typu int, bedzie uzywane jako rozmiar stosu.

Definition at line 68 of file stos\_lista.hh.

##### 4.6.4.2 element\* stos\_lista::przod [private]

Pole typu element, wskazuje na pierwszy element listy.

Definition at line 59 of file stos\_lista.hh.

##### 4.6.4.3 element\* stos\_lista::tyl [private]

Pole typu element, wskazuje na ostatni element listy.

Definition at line 63 of file stos\_lista.hh.

The documentation for this class was generated from the following files:

- [/home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/inc/stos\\_lista.hh](#)
- [/home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/src/stos\\_lista.cpp](#)

## 4.7 stos\_tablica Class Reference

Modeluje pojecie Stos.

```
#include <stos_tablica.hh>
```

### Public Member Functions

- [stos\\_tablica](#) ()  
*Konstruktor klasy stos.*
- [~stos\\_tablica](#) ()  
*Destruktor klasy stos.*
- void [wczytaj\\_dane](#) ()  
*Funkcja wczytywania.*
- void [wczytaj\\_dane](#) (string nazwa)  
*Funkcja wczytywania.*
- void [wczytaj\\_danex2](#) (string nazwa)  
*Funkcja wczytywania.*
- void [pokaz\\_elementy](#) ()  
*Funkcja wyswietlajaca.*
- bool [isEmpty](#) ()  
*Funkcja sprawdzania pustosci stosu.*
- unsigned [size](#) ()

*Funkcja sprawdzania rozmiar stosu.*

- void `push` (int `element`)

*Funkcja dodajaca element.*

- void `pushx2` (int `element`)

*Funkcja dodajaca element.*

- void `pop` ()

*Funkcja usuwajaca ostatni element ze stosu.*

## Private Attributes

- int \* `dane`

*Pole typu int, bedzie uzywane jako stosu z danymi.*

- int \* `danetmp`

*Pole typu int, bedzie uzywane jako stosu z danymi sprawdzajacymi, pomocniczymi.*

- int `rozmiar`

*Pole typu int, bedzie uzywane jako rozmiar tabeli.*

- int `spr`

*Pole typu int, bedzie uzywane jako pomocnicza wartosc jako rozmiar stosu.*

### 4.7.1 Detailed Description

Modeluje pojecie Stos.

Stos jest klasa zawierajaca dynamicznie zaalokowane 2 tablice Pierwsza z nich to tablica z danymi, na ktorych beda wykonywane operacje. Druga z nich sluzi jako tablica do przechowywania tymczasowych danych.

Definition at line 30 of file `stos_tablica.hh`.

### 4.7.2 Constructor & Destructor Documentation

#### 4.7.2.1 `stos_tablica::stos_tablica ( )` `[inline]`

Konstruktor klasy `stos`.

Konstruktor jest bezparametryczny, inicjalizuje wszystkie skladowe klasy wartosciami zerowymi.

Definition at line 59 of file `stos_tablica.hh`.

#### 4.7.2.2 `stos_tablica::~~stos_tablica ( )` `[inline]`

Destruktor klasy `stos`.

Usuwa dynamicznie zaalokowana tablice

Definition at line 66 of file `stos_tablica.hh`.

### 4.7.3 Member Function Documentation

#### 4.7.3.1 `bool stos_tablica::isEmpty ( )`

Funkcja sprawdzania pustosci stosu.

Funkcja sprawdza czy stos jest pusty

**Returns**

1 -> Stos pusty 0 -> Stos nie pusty

Definition at line 5 of file stos\_tablica.cpp.

**4.7.3.2 void stos\_tablica::pokaz\_elementy ( )**

Funkcja wyswietlajaca.

Funkcja wyswietla aktualny stan stosu

Definition at line 93 of file stos\_tablica.cpp.

**4.7.3.3 void stos\_tablica::pop ( )**

Funkcja usuwajaca ostatni element ze stosu.

Funkcja usuwa ostatni element znajdujacy sie na stosie oraz zmniejsza jego rozmiar o jeden.

Definition at line 73 of file stos\_tablica.cpp.

**4.7.3.4 void stos\_tablica::push ( int *element* )**

Funkcja dodajaca element.

Funkcja dodaje do stosu element podany przez uzytkownika Funkcja dodaje element poprzez zwiekszenie rozmiaru tablicy o jeden.

**Parameters**

<i>element</i>	- pole typu int przechowuje wartosc jaka ma byc wpisana
----------------	---

Definition at line 18 of file stos\_tablica.cpp.

**4.7.3.5 void stos\_tablica::pushx2 ( int *element* )**

Funkcja dodajaca element.

Funkcja dodaje do stosu element podany przez uzytkownika Funkcja dodaje element poprzez zwiekszenie rozmiaru tablicy o dwa razy niz aktualny stan.

**Parameters**

<i>element</i>	- pole typu int przechowuje wartosc jaka ma byc wpisana
----------------	---

Definition at line 42 of file stos\_tablica.cpp.

**4.7.3.6 unsigned stos\_tablica::size ( )**

Funkcja sprawdzania rozmiar stosu.

Funkcja podaje aktualny rozmiar stosu

**Returns**

rozmiar -> rozmiar stosu

Definition at line 13 of file stos\_tablica.cpp.

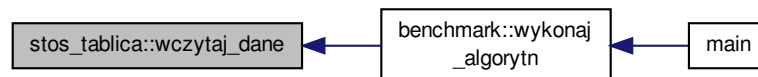
#### 4.7.3.7 void stos\_tablica::wczytaj\_dane ( )

Funkcja wczytywania.

Funkcja wczytuje wartosci do tabeli z podanego pliku przez uzytkownika.

Definition at line 106 of file stos\_tablica.cpp.

Here is the caller graph for this function:



#### 4.7.3.8 void stos\_tablica::wczytaj\_dane ( string nazwa )

Funkcja wczytywania.

Funkcja wczytuje wartosci do tabeli po przez wpisanie nazwy jako argument metody Wykorzystuje metode push jako funkcje wpisujaca do stosu

Parameters

<i><code>nazwa</code></i>	-> zmienna typu string, przechowuje nazwe otwieranego pliku
---------------------------	---

Definition at line 123 of file stos\_tablica.cpp.

#### 4.7.3.9 void stos\_tablica::wczytaj\_danex2 ( string nazwa )

Funkcja wczytywania.

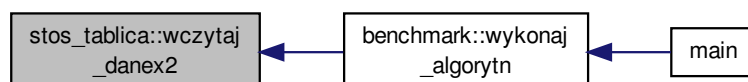
Funkcja wczytuje wartosci do tabeli po przez wpisanie nazwy jako argument metody Wykorzystuje metode pushx2 jako funkcje wpisujaca do stosu

Parameters

<i><code>nazwa</code></i>	-> zmienna typu string, przechowuje nazwe otwieranego pliku
---------------------------	---

Definition at line 137 of file stos\_tablica.cpp.

Here is the caller graph for this function:



### 4.7.4 Member Data Documentation

#### 4.7.4.1 `int* stos_tablica::dane` `[private]`

Pole typu `int`, bedzie uzywane jako stosu z danymi.

Definition at line 35 of file `stos_tablica.hh`.

#### 4.7.4.2 `int* stos_tablica::danetmp` `[private]`

Pole typu `int`, bedzie uzywane jako stosu z danymi sprawdzajacymi, pomocniczymi.

Definition at line 39 of file `stos_tablica.hh`.

#### 4.7.4.3 `int stos_tablica::rozmiar` `[private]`

Pole typu `int`, bedzie uzywane jako rozmiar tabeli.

Definition at line 44 of file `stos_tablica.hh`.

#### 4.7.4.4 `int stos_tablica::spr` `[private]`

Pole typu `int`, bedzie uzywane jako pomocnicza wartosc jako rozmiar stosu.

Definition at line 48 of file `stos_tablica.hh`.

The documentation for this class was generated from the following files:

- `/home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/inc/stos_tablica.hh`
- `/home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/src/stos_tablica.cpp`



## Chapter 5

# File Documentation

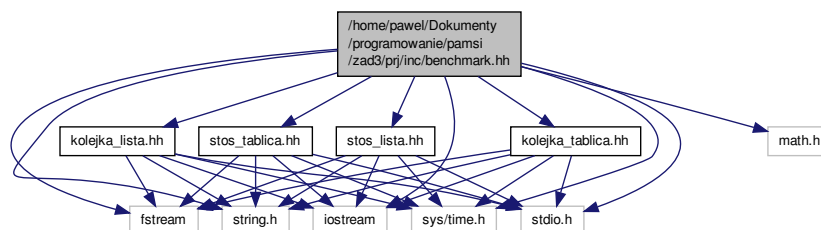
### 5.1 /home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/doc/pages/strona.dox File Reference

### 5.2 /home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/inc/benchmark.hh File Reference

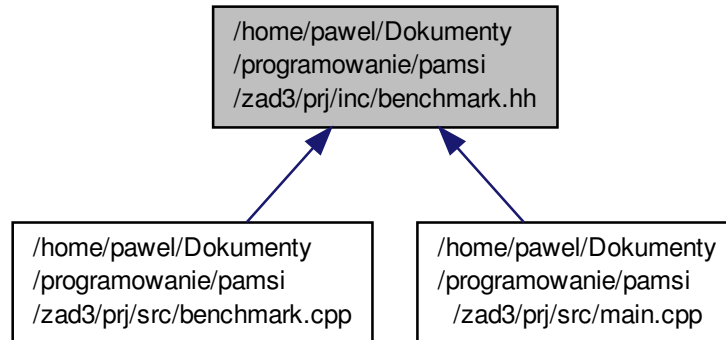
Definicje funkcji dla klasy benchmark.

```
#include <fstream>
#include <iostream>
#include <math.h>
#include <string.h>
#include <stdio.h>
#include <sys/time.h>
#include "kolejka_tablica.hh"
#include "kolejka_lista.hh"
#include "stos_tablica.hh"
#include "stos_lista.hh"
```

Include dependency graph for benchmark.hh:



This graph shows which files directly or indirectly include this file:



## Classes

- class [benchmark](#)

*Modeluje pojecie Benchmark.*

### 5.2.1 Detailed Description

Definicje funkcji dla klasy benchmark.

Definition in file [benchmark.hh](#).

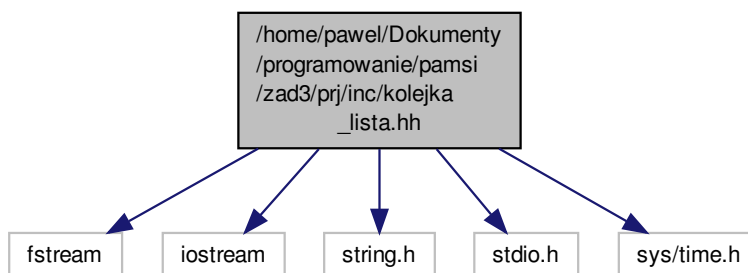
## 5.3 /home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/inc/kolejka\_lista.hh File Reference

Definicje funkcji dla klasy Kolejka zdefiniowanej lista.

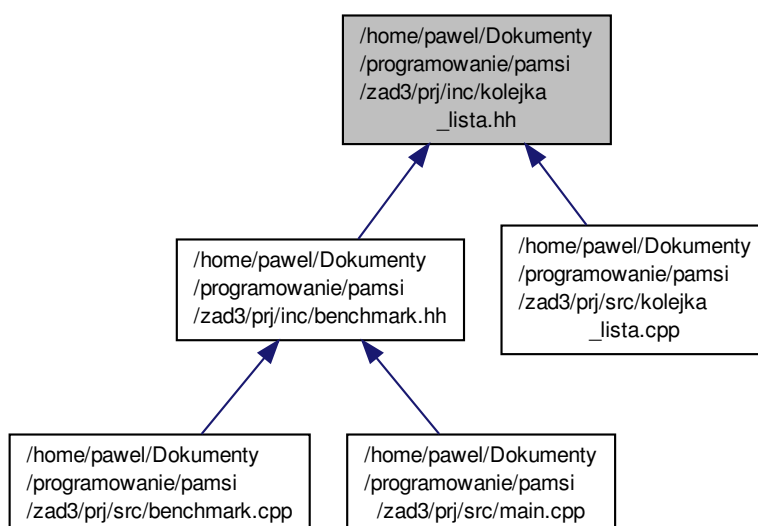
```
#include <fstream>
#include <iostream>
#include <string.h>
#include <stdio.h>
#include <sys/time.h>
```



Include dependency graph for `kolejka_lista.hh`:



This graph shows which files directly or indirectly include this file:



## Classes

- class [elementk](#)  
*Modeluje pojecie elementk (zmieniona nazwa, poniewaz [stos\\_lista](#) tez ma klase o nazwie element).*
- class [kolejka\\_lista](#)  
*Modeluje pojecie Kolejka.*

### 5.3.1 Detailed Description

Definicje funkcji dla klasy Kolejka zdefiniowanej lista.

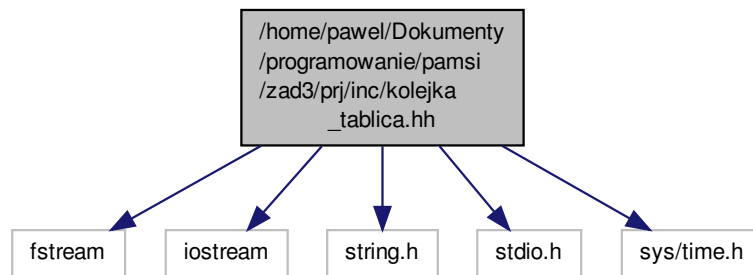
Definition in file [kolejka\\_lista.hh](#).

## 5.4 /home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/inc/kolejka\_tablica.hh File Reference

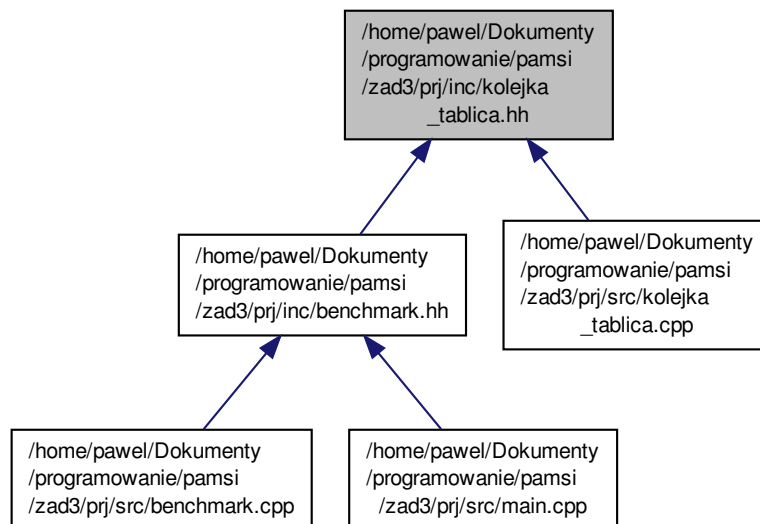
Definicje funkcji dla klasy Kolejka zdefiniowanej tablica.

```
#include <fstream>
#include <iostream>
#include <string.h>
#include <stdio.h>
#include <sys/time.h>
```

Include dependency graph for kolejka\_tablica.hh:



This graph shows which files directly or indirectly include this file:



### Classes

- class [kolejka\\_tablica](#)

*Modeluje pojecie Kolejka.*

### 5.4.1 Detailed Description

Definicje funkcji dla klasy Kolejka zdefiniowanej tablica.

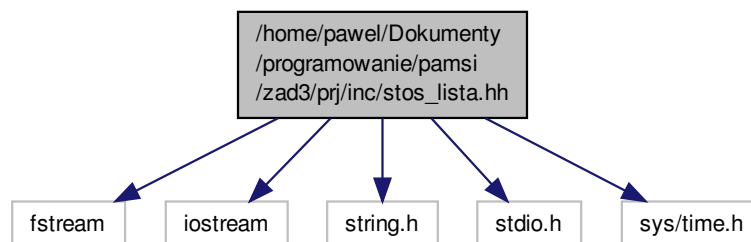
Definition in file [kolejka\\_tablica.hh](#).

## 5.5 /home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/inc/stos\_lista.hh File Reference

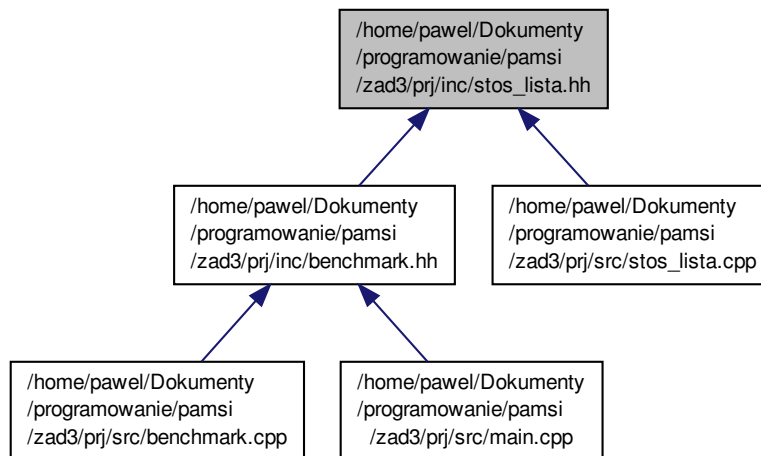
Definicje funkcji dla klasy Stos zdefiniowanej lista.

```
#include <fstream>
#include <iostream>
#include <string.h>
#include <stdio.h>
#include <sys/time.h>
```

Include dependency graph for stos\_lista.hh:



This graph shows which files directly or indirectly include this file:



## Classes

- class [element](#)

*Modeluje pojecie element.*

- class [stos\\_lista](#)

*Modeluje pojecie Stos.*

### 5.5.1 Detailed Description

Definicje funkcji dla klasy Stos zdefiniowanej lista.

Definition in file [stos\\_lista.hh](#).

## 5.6 /home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/inc/stos\_tablica.hh File Reference

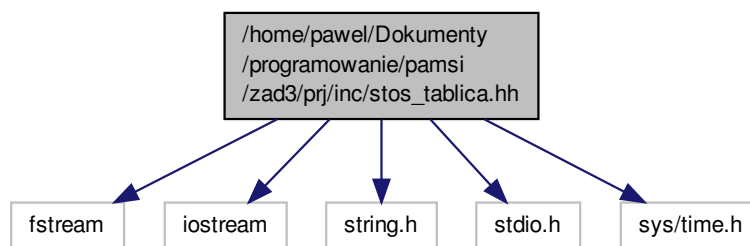
Definicje funkcji dla klasy Stos zdefiniowanej tablica.

```

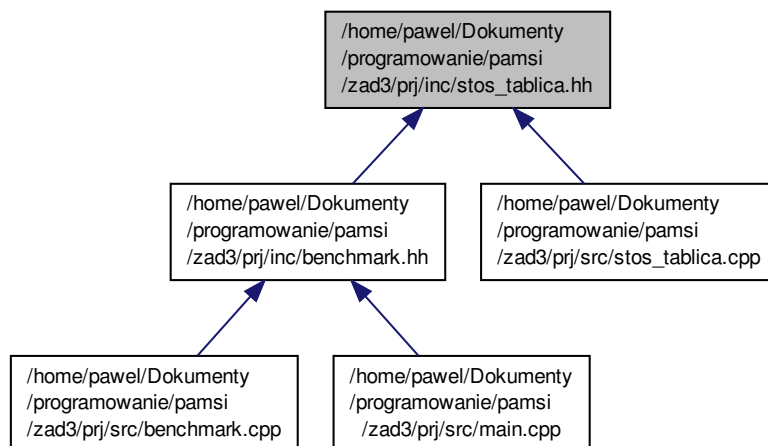
#include <fstream>
#include <iostream>
#include <string.h>
#include <stdio.h>
#include <sys/time.h>

```

Include dependency graph for stos\_tablica.hh:



This graph shows which files directly or indirectly include this file:



## Classes

- class `stos_tablica`  
*Modeluje pojecie Stos.*

### 5.6.1 Detailed Description

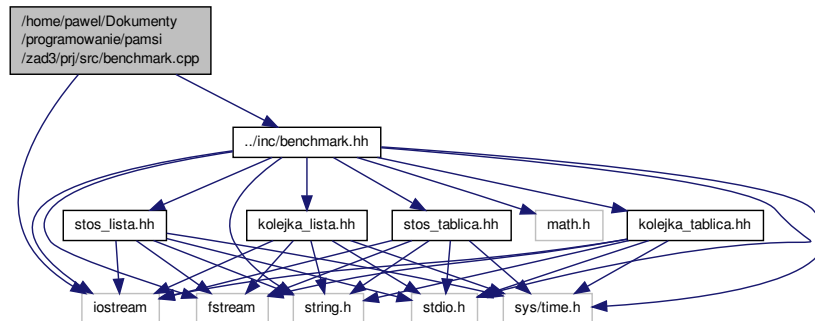
Definicje funkcji dla klasy Stos zdefiniowanej tablica.

Definition in file `stos_tablica.hh`.

## 5.7 /home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/src/benchmark.cpp File Reference

Plik zawiera funkcje z klasy benchmark.

```
#include "../inc/benchmark.hh"
#include <iostream>
Include dependency graph for benchmark.cpp:
```



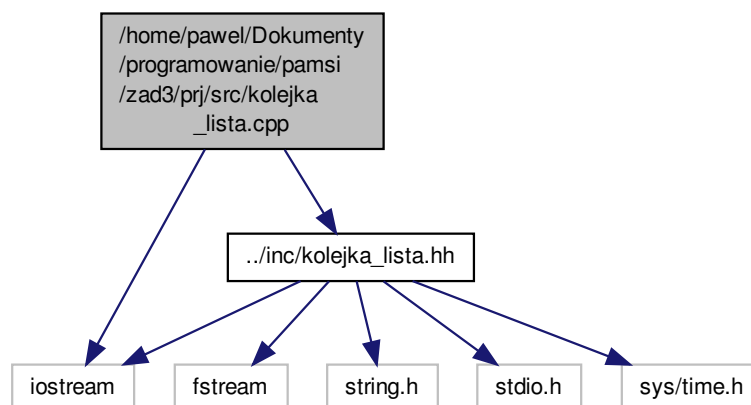
### 5.7.1 Detailed Description

Plik zawiera funkcje z klasy benchmark.

Definition in file [benchmark.cpp](#).

## 5.8 /home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/src/kolejka\_lista.cpp File Reference

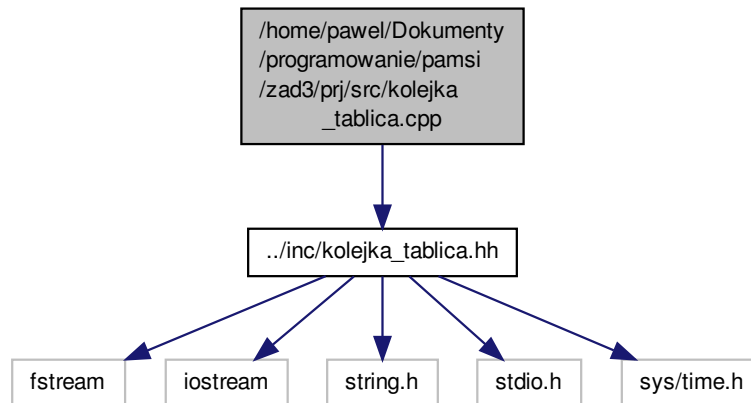
```
#include "../inc/kolejka_lista.hh"
#include <iostream>
Include dependency graph for kolejka_lista.cpp:
```



## 5.9 /home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/src/kolejka\_tablica.cpp File Reference

```
#include "../inc/kolejka_tablica.hh"
```

Include dependency graph for kolejka\_tablica.cpp:



## 5.10 /home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/src/main.cpp File Reference

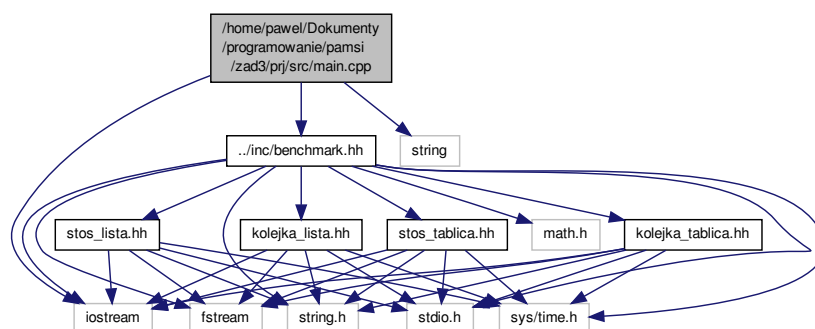
Plik zawiera funkcje `main()`

```
#include <iostream>
```

```
#include <string>
```

```
#include "../inc/benchmark.hh"
```

Include dependency graph for main.cpp:



### Functions

- `int main()`

Aby uruchomic prosty interfejs graiczny nalezy odkomentowac podane nizej linijki kodu i usunac linie " benchmark B; i B.petla(); ".

### 5.10.1 Detailed Description

Plik zawiera funkcje `main()`

Definition in file `main.cpp`.

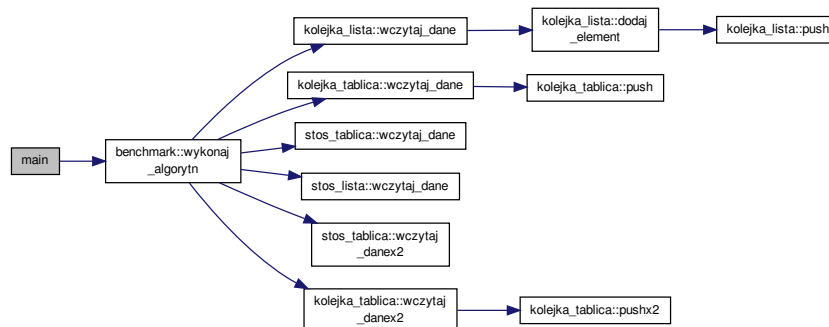
### 5.10.2 Function Documentation

#### 5.10.2.1 `int main ( )`

Aby uruchomic prosty interfejs graiczny nalezy odkomentowac podane nizej linijki kodu i usunac linie " benchmark B; i B.petla(); ".

Definition at line 20 of file `main.cpp`.

Here is the call graph for this function:

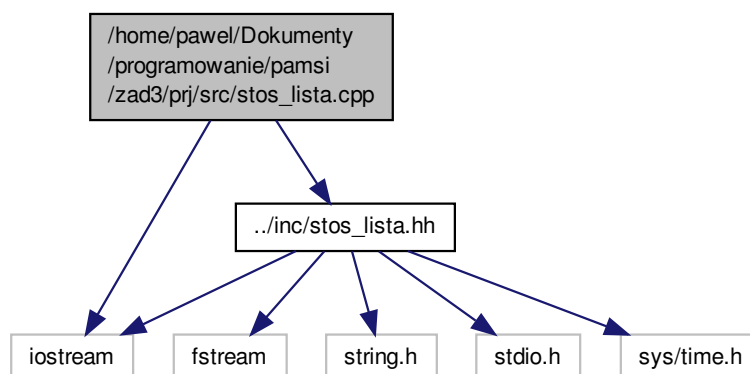


## 5.11 /home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/src/stos\_lista.cpp File Reference

```
#include "../inc/stos_lista.hh"
#include <iostream>
```



Include dependency graph for stos\_lista.cpp:

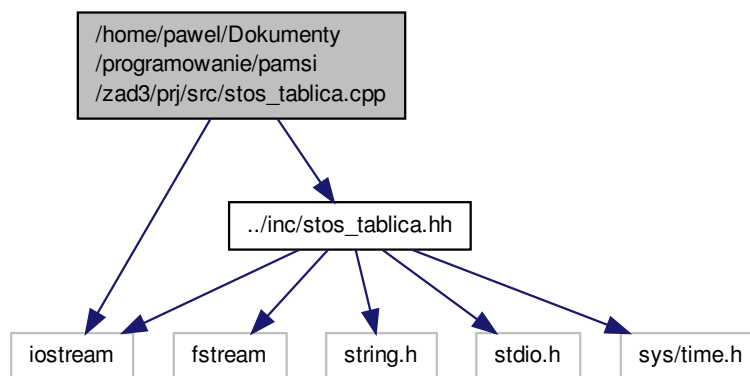


## 5.12 /home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/src/stos\_tablica.cpp File Reference

```
#include "../inc/stos_tablica.hh"
```

```
#include <iostream>
```

Include dependency graph for stos\_tablica.cpp:



# Index

~kolejka\_lista  
    kolejka\_lista, 13  
~kolejka\_tablica  
    kolejka\_tablica, 19  
~stos\_lista  
    stos\_lista, 24  
~stos\_tablica  
    stos\_tablica, 28  
/home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/doc/paginas/stos\_lista.dox, 33  
/home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/inc/elapsedTime.hh, 33  
/home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/inc/kolejka\_lista.hh, 34  
/home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/inc/kolejka\_tablica.hh, 36  
/home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/inc/stos\_lista.hh, 37  
/home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/inc/stos\_tablica.hh, 38  
/home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/src/benchmark.cpp, 39  
/home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/src/kolejka\_lista.cpp, 40  
/home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/src/kolejka\_tablica.cpp, 41  
/home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/src/main.cpp, 41  
/home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/src/stos\_lista.cpp, 42  
/home/pawel/Dokumenty/programowanie/pamsi/zad3/prj/src/stos\_tablica.cpp, 43  
  
benchmark, 7  
    benchmark, 8  
    czas, 9  
    czas\_start, 8  
    czas\_stop, 8  
    elapsedTime, 10  
    ile\_czasu, 8  
    t1, 10  
    t2, 10  
    wykonaj\_algorytm, 8  
  
czas  
    benchmark, 9  
czas\_start  
    benchmark, 8  
czas\_stop  
    benchmark, 8  
  
dane  
    kolejka\_tablica, 22  
    stos\_tablica, 30  
danetmp  
    kolejka\_tablica, 23  
    stos\_tablica, 31  
dodaj\_element  
    kolejka\_lista, 14  
    stos\_lista, 25  
elapsedTime  
    benchmark, 10  
element  
    klucz, 11  
    nastepny, 11  
elementk  
    klucz, 12  
    nastepny, 12  
ile\_czasu  
    benchmark, 8  
isEmpty  
    kolejka\_tablica, 19  
    stos\_tablica, 28  
isEmpty  
    kolejka\_lista, 15  
    stos\_lista, 25  
klucz  
    element, 11  
    elementk, 12  
kolejka\_lista, 12  
    ~kolejka\_lista, 13  
    dodaj\_element, 14  
    isEmpty, 15  
    kolejka\_lista, 13  
    kolejka\_lista, 13  
    licznik, 17  
    pokaz\_elementy, 15  
    pop, 15  
    przod, 17  
    push, 16  
    size, 16  
    tyl, 18  
    wczytaj\_dane, 16, 17  
kolejka\_tablica, 18  
    ~kolejka\_tablica, 19  
    dane, 22  
    danetmp, 23  
    isEmpty, 19

- kolejka\_tablica, [19](#)
- kolejka\_tablica, [19](#)
- pokaz\_elementy, [19](#)
- pop, [19](#)
- push, [20](#)
- pushx2, [20](#)
- rozmiar, [23](#)
- size, [20](#)
- spr, [23](#)
- wczytaj\_dane, [21](#)
- wczytaj\_danex2, [22](#)
- licznik
  - kolejka\_lista, [17](#)
  - stos\_lista, [27](#)
- main
  - main.cpp, [42](#)
- main.cpp
  - main, [42](#)
- nastepny
  - element, [11](#)
  - elementk, [12](#)
- pokaz\_elementy
  - kolejka\_lista, [15](#)
  - kolejka\_tablica, [19](#)
  - stos\_lista, [25](#)
  - stos\_tablica, [29](#)
- pop
  - kolejka\_lista, [15](#)
  - kolejka\_tablica, [19](#)
  - stos\_lista, [25](#)
  - stos\_tablica, [29](#)
- przod
  - kolejka\_lista, [17](#)
  - stos\_lista, [27](#)
- push
  - kolejka\_lista, [16](#)
  - kolejka\_tablica, [20](#)
  - stos\_lista, [25](#)
  - stos\_tablica, [29](#)
- pushx2
  - kolejka\_tablica, [20](#)
  - stos\_tablica, [29](#)
- rozmiar
  - kolejka\_tablica, [23](#)
  - stos\_tablica, [31](#)
- size
  - kolejka\_lista, [16](#)
  - kolejka\_tablica, [20](#)
  - stos\_lista, [26](#)
  - stos\_tablica, [29](#)
- spr
  - kolejka\_tablica, [23](#)
  - stos\_tablica, [31](#)
- stos\_lista, [23](#)
- ~stos\_lista, [24](#)
- dodaj\_element, [25](#)
- isempty, [25](#)
- licznik, [27](#)
- pokaz\_elementy, [25](#)
- pop, [25](#)
- przod, [27](#)
- push, [25](#)
- size, [26](#)
- stos\_lista, [24](#)
- stos\_lista, [24](#)
- tyl, [27](#)
- wczytaj\_dane, [26](#)
- stos\_tablica, [27](#)
- ~stos\_tablica, [28](#)
- dane, [30](#)
- danetmp, [31](#)
- isEmpty, [28](#)
- pokaz\_elementy, [29](#)
- pop, [29](#)
- push, [29](#)
- pushx2, [29](#)
- rozmiar, [31](#)
- size, [29](#)
- spr, [31](#)
- stos\_tablica, [28](#)
- stos\_tablica, [28](#)
- wczytaj\_dane, [29](#), [30](#)
- wczytaj\_danex2, [30](#)
- t1
  - benchmark, [10](#)
- t2
  - benchmark, [10](#)
- tyl
  - kolejka\_lista, [18](#)
  - stos\_lista, [27](#)
- wczytaj\_dane
  - kolejka\_lista, [16](#), [17](#)
  - kolejka\_tablica, [21](#)
  - stos\_lista, [26](#)
  - stos\_tablica, [29](#), [30](#)
- wczytaj\_danex2
  - kolejka\_tablica, [22](#)
  - stos\_tablica, [30](#)
- wykonaj\_algorytm
  - benchmark, [8](#)