

My Project

Generated by Doxygen 1.8.6

Fri May 29 2015 17:15:16

Contents

1	Main Page	1
1.1	Programu	1
2	Hierarchical Index	3
2.1	Class Hierarchy	3
3	Class Index	5
3.1	Class List	5
4	File Index	7
4.1	File List	7
5	Class Documentation	9
5.1	Czasomierz Class Reference	9
5.1.1	Detailed Description	10
5.1.2	Constructor & Destructor Documentation	10
5.1.2.1	Czasomierz	10
5.1.3	Member Function Documentation	10
5.1.3.1	_AktualizujCzas	10
5.1.3.2	_CzasTrwania	10
5.1.3.3	_PojedynczyPomiar	10
5.1.3.4	_Reset	10
5.1.3.5	_RozpocznijPomiar	10
5.1.3.6	_StatusPracy	10
5.1.3.7	_ZakonczPomiar	11
5.1.4	Member Data Documentation	11
5.1.4.1	_Aktualny	11
5.1.4.2	_Koniec	11
5.1.4.3	_Start	11
5.1.4.4	_Status	11
5.2	Graf< Typ > Class Template Reference	11
5.2.1	Detailed Description	12
5.2.2	Constructor & Destructor Documentation	12

5.2.2.1	Graf	12
5.2.2.2	~Graf	13
5.2.3	Member Function Documentation	13
5.2.3.1	_CzyKrawedz	13
5.2.3.2	_DodajKrawedz	13
5.2.3.3	_DodajWartoscWierzcholka	13
5.2.3.4	_IloscKrawedzi	13
5.2.3.5	_UsunKrawedz	14
5.2.3.6	_Wykonaj	14
5.2.3.7	_Zaladuj	14
5.2.3.8	_Zwolnij	14
5.2.3.9	_ZwrocWartoscWierzcholka	14
5.2.3.10	BFS	14
5.2.3.11	DFS	15
5.2.4	Member Data Documentation	15
5.2.4.1	_E	15
5.2.4.2	_EMacierz	15
5.2.4.3	_V	15
5.2.4.4	_VMacierz	15
5.3	IObserwator Class Reference	15
5.3.1	Detailed Description	16
5.3.2	Member Function Documentation	16
5.3.2.1	_Aktualizuj	16
5.4	IObserwowany Class Reference	16
5.4.1	Detailed Description	17
5.4.2	Member Function Documentation	17
5.4.2.1	_DodajObserwator	17
5.4.2.2	_PowiadomObserwatorow	17
5.4.2.3	_UsunObserwator	17
5.5	Itest Class Reference	17
5.5.1	Detailed Description	18
5.5.2	Member Function Documentation	18
5.5.2.1	_Wykonaj	18
5.5.2.2	_Zaladuj	18
5.5.2.3	_Zwolnij	18
5.6	Kolejka< Typ > Class Template Reference	18
5.6.1	Detailed Description	19
5.6.2	Constructor & Destructor Documentation	20
5.6.2.1	Kolejka	20
5.6.2.2	~Kolejka	20

5.6.3	Member Function Documentation	20
5.6.3.1	_Pokaz	20
5.6.3.2	_Pop	20
5.6.3.3	_Push	20
5.6.3.4	_Rozmiar	20
5.6.3.5	_Zwolnij	21
5.6.3.6	CzyPusta	21
5.6.4	Member Data Documentation	21
5.6.4.1	_Ilosc	21
5.6.4.2	_Ostatni	21
5.6.4.3	_Pierwszy	21
5.7	Lista< Typ > Class Template Reference	21
5.7.1	Constructor & Destructor Documentation	22
5.7.1.1	Lista	22
5.7.1.2	~Lista	22
5.7.2	Member Function Documentation	22
5.7.2.1	_Pokaz	22
5.7.2.2	_Pop	22
5.7.2.3	_Push	23
5.7.2.4	_Rozmiar	23
5.7.2.5	_Zwolnij	23
5.7.3	Member Data Documentation	23
5.7.3.1	_Ilosc	23
5.7.3.2	Glowa	23
5.8	StosTab< Typ > Class Template Reference	23
5.8.1	Constructor & Destructor Documentation	24
5.8.1.1	StosTab	24
5.8.1.2	StosTab	24
5.8.1.3	~StosTab	24
5.8.2	Member Function Documentation	25
5.8.2.1	_Pokaz	25
5.8.2.2	_Pop	25
5.8.2.3	_Push	25
5.8.2.4	_Rozmiar	25
5.8.2.5	_Zwolnij	25
5.8.2.6	CzyPusty	26
5.8.3	Member Data Documentation	26
5.8.3.1	_L	26
5.8.3.2	_RozmiarL	26
5.8.3.3	_RozmiarT	26

5.9	Struktury< Typ > Class Template Reference	26
5.9.1	Detailed Description	27
5.9.2	Member Function Documentation	27
5.9.2.1	_Pokaz	27
5.9.2.2	_Pop	27
5.9.2.3	_Push	27
5.9.2.4	_Rozmiar	27
5.9.2.5	_Zwolnij	28
5.10	StrukturyBenchmark< Typ > Class Template Reference	28
5.10.1	Detailed Description	29
5.10.2	Constructor & Destructor Documentation	29
5.10.2.1	StrukturyBenchmark	29
5.10.2.2	~StrukturyBenchmark	29
5.10.2.3	StrukturyBenchmark	29
5.10.3	Member Function Documentation	29
5.10.3.1	_DodajObserwator	29
5.10.3.2	_PowiadomObserwatorow	29
5.10.3.3	_UsunObserwator	29
5.10.3.4	_WykonajTest	30
5.10.4	Member Data Documentation	30
5.10.4.1	_IloscPowt	30
5.10.4.2	_IloscProb	30
5.10.4.3	_TablicaRozmiarow	30
5.10.4.4	Obserwatorzy	30
5.11	Kolejka< Typ >::Wezel Struct Reference	30
5.11.1	Detailed Description	31
5.11.2	Member Data Documentation	31
5.11.2.1	_Nast	31
5.11.2.2	_Wartosc	31
5.12	Lista< Typ >::Wezel Struct Reference	31
5.12.1	Detailed Description	31
5.12.2	Member Data Documentation	31
5.12.2.1	_Nast	31
5.12.2.2	_Wartosc	31
5.13	Wyniki Class Reference	32
5.13.1	Constructor & Destructor Documentation	32
5.13.1.1	Wyniki	32
5.13.1.2	Wyniki	32
5.13.1.3	~Wyniki	32
5.13.2	Member Function Documentation	32

5.13.2.1	_Aktualizuj	32
5.13.2.2	_ZapiszWyniki	32
5.13.3	Member Data Documentation	32
5.13.3.1	_IloscPowtorzen	32
5.13.3.2	_IloscProb	33
5.13.3.3	_TablicaRozmiarow	33
5.13.3.4	_TablicaWynikow	33
5.13.3.5	Stoper	33
6	File Documentation	35
6.1	/home/bartolomeo/209255/prj/doc/strona-glowna.dox File Reference	35
6.2	/home/bartolomeo/209255/prj/inc/Czasomierz.hh File Reference	35
6.3	/home/bartolomeo/209255/prj/inc/Graf.hh File Reference	35
6.4	/home/bartolomeo/209255/prj/inc/IObserwator.hh File Reference	35
6.5	/home/bartolomeo/209255/prj/inc/IObserwowany.hh File Reference	36
6.6	/home/bartolomeo/209255/prj/inc/Itest.hh File Reference	36
6.7	/home/bartolomeo/209255/prj/inc/Kolejka.hh File Reference	36
6.8	/home/bartolomeo/209255/prj/inc/Lista.hh File Reference	36
6.9	/home/bartolomeo/209255/prj/inc/StosTab.hh File Reference	37
6.10	/home/bartolomeo/209255/prj/inc/Struktury.hh File Reference	37
6.11	/home/bartolomeo/209255/prj/inc/StrukturyBenchmark.hh File Reference	37
6.12	/home/bartolomeo/209255/prj/inc/Wyniki.hh File Reference	37
6.13	/home/bartolomeo/209255/prj/src/Czasomierz.cpp File Reference	38
6.14	/home/bartolomeo/209255/prj/src/Graf.cpp File Reference	38
6.15	/home/bartolomeo/209255/prj/src/Main.cpp File Reference	38
6.15.1	Detailed Description	38
6.15.2	Function Documentation	38
6.15.2.1	main	38
6.15.3	Variable Documentation	38
6.15.3.1	ILOSC_POW	38
6.15.3.2	ILOSC_PROB	38
6.16	/home/bartolomeo/209255/prj/src/Wyniki.cpp File Reference	38
	Index	39

Chapter 1

Main Page

Author

Bartłomiej Ankowski

Date

16.04.2015

Version

0.3

1.1 Programu

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Czasomierz	9
IObservator	15
Wyniki	32
IObservowany	16
StrukturyBenchmark< Typ >	28
Itest	17
Graf< Typ >	11
Struktury< Typ >	26
Kolejka< Typ >	18
Lista< Typ >	21
StosTab< Typ >	23
Kolejka< Typ >::Wezel	30
Lista< Typ >::Wezel	31

Chapter 3

Class Index

3.1 Class List

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

Czasomierz	
Modeluje pojecie Czasomierza	9
Graf< Typ >	
Modeluje pojecie grafu	11
IObserwator	
Modeluje pojecie interfejsu dla obserwatora	15
IObserwowany	
Interfejs dla Obserwatora	16
Itest	
Modeluje pojecie Interfejsu Testujacego	17
Kolejka< Typ >	
Modeluje pojecie Kolejki	18
Lista< Typ >	21
StosTab< Typ >	23
Struktury< Typ >	
Modeluje pojecie Struktury danych, klasa bazowa dla Stosu,Kolejki i Listy,zarowno w implemene- tacji wskaznikowej jak i tablicowej	26
StrukturyBenchmark< Typ >	28
Kolejka< Typ >::Wezel	30
Lista< Typ >::Wezel	31
Wyniki	32

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

/home/bartolomeo/209255/prj/inc/Czasomierz.hh	35
/home/bartolomeo/209255/prj/inc/Graf.hh	35
/home/bartolomeo/209255/prj/inc/IObserwator.hh	35
/home/bartolomeo/209255/prj/inc/IObserwowany.hh	36
/home/bartolomeo/209255/prj/inc/Itest.hh	36
/home/bartolomeo/209255/prj/inc/Kolejka.hh	36
/home/bartolomeo/209255/prj/inc/Lista.hh	36
/home/bartolomeo/209255/prj/inc/StosTab.hh	37
/home/bartolomeo/209255/prj/inc/Struktury.hh	37
/home/bartolomeo/209255/prj/inc/StrukturyBenchmark.hh	37
/home/bartolomeo/209255/prj/inc/Wyniki.hh	37
/home/bartolomeo/209255/prj/src/Czasomierz.cpp	38
/home/bartolomeo/209255/prj/src/Graf.cpp	38
/home/bartolomeo/209255/prj/src/Main.cpp	
Funkcja glowna programu	38
/home/bartolomeo/209255/prj/src/Wyniki.cpp	38

Chapter 5

Class Documentation

5.1 Czasomierz Class Reference

Modeluje pojecie Czasomierza.

```
#include <Czasomierz.hh>
```

Public Member Functions

- [Czasomierz \(\)](#)
Konstruktor obiektu.
- void [_RozpocznijPomiar \(\)](#)
Metoda zaczynajaca pomiar.
- void [_ZakonczPomiar \(\)](#)
Metoda konczaca pomiar.
- void [_AktualizujCzas \(\)](#)
Metoda Aktualizujaca czas danej proby.
- long double [_CzasTrwania \(\) const](#)
Metoda zwracajaca calkowity czas Proby.
- void [_Reset \(\)](#)
Metoda resetujaca [Czasomierz](#).
- bool [_StatusPracy \(\) const](#)
- double [_PojedynczyPomiar \(\) const](#)
Metoda zwracajaca czas pojedynczego pomiaru.

Private Attributes

- clock_t [_Start](#)
Pole klasy [Czasomierz](#).
- clock_t [_Koniec](#)
Pole klasy [Czasomierz](#).
- long double [_Aktualny](#)
Pole klasy [Czasomierz](#).
- bool [_Status](#)
Pole klasy [Czasomierz](#).

5.1.1 Detailed Description

Modeluje pojecie Czasomierza.

Klasa ma za zadanie mierzyc czas wykonywania danego algorytmu oraz przechowywac informacje o srednich czasach pomiarow dla wybranej ilosci danych przy okreslonej liczbie powtorzen testu

5.1.2 Constructor & Destructor Documentation

5.1.2.1 Czasomierz::Czasomierz ()

Konstruktor obiektu.

Konstruktor klasy [Czasomierz](#), zerujacy wszystkie pola tej klasy

5.1.3 Member Function Documentation

5.1.3.1 void Czasomierz::_AktualizujCzas ()

Metoda Aktualizujaca czas danej proby.

Metoda ma za zadanie zaktualizowac czas danej proby, za sprawa dodadania do poprzedniej sumy pomiarow, czas ktory przypada na najnowszy pomiar

5.1.3.2 long double Czasomierz::_CzasTrwania () const

Metoda zwracajaca calkowity czas Proby.

Metoda ma za zadanie zwrocic calkowity czas przypadajacy na dana probe

5.1.3.3 double Czasomierz::_PojedynczyPomiar () const

Metoda zwracajaca czas pojedynczego pomiaru.

Metoda ma za zadanie zwrocic czas trwania pojedynczego testu

5.1.3.4 void Czasomierz::_Reset ()

Metoda resetujaca [Czasomierz](#).

Metoda ma za zadanie wyzerowac pola przechowujace informacje o czasie systemowym zapisanym na poczatku i na koncu pomiaru. Zmienia status czasomierza na wolny

5.1.3.5 void Czasomierz::_RozpocznijPomiar ()

Metoda zaczynajaca pomiar.

Metoda ma za zadanie odczytac czas systemowy w momencie jej wywolania i zapisania tej informacji w odpowiednim polu klasy. Zmienia rowniez status pracy czasomierza na zajety.

5.1.3.6 bool Czasomierz::_StatusPracy () const

informujaca o stanie czasomierza

Metoda ma za zadanie poinformowanie w jakim stanie znajduje sie obecnie czasomierz

5.1.3.7 void Czasomierz::_ZakoncZPomiar ()

Metoda konczaca pomiar.

Metoda ma za zadanie odczytac czas systemowy w momencie jej wywolania i zapisania tej informacji w odpowiednim polu klasy. Zmienia status pracy czasomierza na wolny

5.1.4 Member Data Documentation

5.1.4.1 long double Czasomierz::_Aktualny [private]

Pole klasy [Czasomierz](#).

Pole przechowuje czas w ms, odpowiadajacy czasowi aktualnego pomiaru dla danej ilosci danych

5.1.4.2 clock_t Czasomierz::_Koniec [private]

Pole klasy [Czasomierz](#).

Pole przechowuje informacje, ktora jest czas systemu w chwili zakonczenia pomiaru

5.1.4.3 clock_t Czasomierz::_Start [private]

Pole klasy [Czasomierz](#).

Pole przechowuje informacje, ktora jest czas systemu w chwili uruchomienia Stopera

5.1.4.4 bool Czasomierz::_Status [private]

Pole klasy [Czasomierz](#).

Pole przechowuje informacje o stanie Czasomierza, tzn czy obecnie odmierza czas czy nie

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

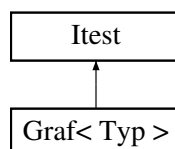
- [/home/bartolomeo/209255/prj/inc/Czasomierz.hh](#)
- [/home/bartolomeo/209255/prj/src/Czasomierz.cpp](#)

5.2 Graf< Typ > Class Template Reference

Modeluje pojecie grafu.

```
#include <Graf.hh>
```

Inheritance diagram for Graf< Typ >:



Public Member Functions

- void [_Wykonaj](#) (unsigned int n) const

- Metoda wywołująca testowany algorytm.*
- void [_Zwolnij](#) ()
Metoda zwalnijająca pamięć.
- void [_Zaladuj](#) (unsigned int n)
Metoda ładująca dane.
- [Graf](#) (unsigned int n)
Konstruktor obiektu.
- virtual [~Graf](#) ()
Destruktor obiektu.
- void [_DodajKrawedz](#) (unsigned int i, unsigned int j)
Metoda dodająca Krawedz.
- void [_UsunKrawedz](#) (const unsigned int i, const unsigned int j)
Metoda usuwająca połączenie.
- bool [_CzyKrawedz](#) (const unsigned int i, const unsigned int j) const
Metoda sprawdzająca status połączenia.
- void [BFS](#) (const int i, const int Szukany) const
Algorytm BFS.
- void [DFS](#) (const int x, const int Wymagany) const
Algorytm BFS.
- void [_DodajWartoscWierzcholka](#) (const unsigned int indeks, const Typ Wartosc)
Metoda dodająca wartość do danego wierzchołka.
- Typ [_ZwrocWartoscWierzcholka](#) (const unsigned int indeks) const
Metoda zwracająca wartość z wierzchołka.
- unsigned int [_IloscKrawedzi](#) () const
Metoda zwracająca ilość krawędzi.

Private Attributes

- unsigned int [_V](#)
Pole klasy [Graf](#).
- unsigned int [_E](#)
Pole klasy [Graf](#).
- bool ** [_EMacierz](#)
Pole klasy [Graf](#).
- Typ * [_VMacierz](#)
Pole klasy [Graf](#).

5.2.1 Detailed Description

```
template<class Typ>class Graf< Typ >
```

Modeluje pojecie grafu.

Klasa modeluje pojecie Grafu, przedstawionego jako Macierz sasiedztw, która przechowuje informacje na temat polaczen miedzy wierzchołkami

5.2.2 Constructor & Destructor Documentation

5.2.2.1 `template<class Typ > Graf< Typ >::Graf (unsigned int n)`

Konstruktor obiektu.

Konstruktor ma za zadanie zaalokowac pamiec potrzebna dla macerzy sasiedztw i wstepnie wypelnic ja wartosciami false

Parameters

<i>in</i>	<i>n</i>	- wielkosc tworzonego grafu
-----------	----------	-----------------------------

5.2.2.2 `template<class Typ> Graf< Typ>::~~Graf () [virtual]`

Destruktor obiektu.

Destruktor ma za zadanie uwolnic pamiec przeznaczona na realizacje grafu

5.2.3 Member Function Documentation

5.2.3.1 `template<class Typ> bool Graf< Typ>::_CzyKrawedz (const unsigned int i, const unsigned int j) const`

Metoda sprawdzajaca status polaczenia.

Metoda ma za zadanie sprawdzic status polaczenia miedzy wierzchokami zadanymi poprzez argument

Parameters

<i>in</i>	<i>i</i>	- numer wierzcholka
<i>in</i>	<i>j</i>	- numer wierzcholka

5.2.3.2 `template<class Typ> void Graf< Typ>::_DodajKrawedz (unsigned int i, unsigned int j)`

Metoda dodajaca Krawedz.

Metoda ma za zadanie zmienic status polaczen zadanых poprzez argument wierzcholkow na polaczony

Parameters

<i>in</i>	<i>i</i>	- numer wierzcholka,ktory ma zostac polaczony z drugim argumentem
<i>in</i>	<i>j</i>	- numer wierzcholka, ktory ma zostac polaczony z pierwszym argumentem

5.2.3.3 `template<class Typ> void Graf< Typ>::_DodajWartoscWierzcholka (const unsigned int indeks, const Typ Wartosc)`

Metoda dodajaca wartosc do danego wierzcholka.

Metoda ma za zadanie dodac wartosc do danego wierzcholka

Parameters

<i>in</i>	<i>indeks</i>	- Numer wierzcholka do ktorego ma zostac dodana wartosc
<i>in</i>	<i>Wartosc</i>	- Wartosc,ktora ma zostac dodana do wierzcholka

5.2.3.4 `template<class Typ> unsigned int Graf< Typ>::_IloscKrawedzi () const`

Metoda zwracaja ilsoc krawedzi.

Metoda ma za zadanie zwrocic ilosc krawedzi jaka zostala stworzona w zamodelowanym grafie

Returns

Ilosc krawedzi grafu

5.2.3.5 `template<class Typ> void Graf< Typ>::_UsunKrawedz (const unsigned int i, const unsigned int j)`

Metoda usuwajaca polaczenie.

Metoda ma zadanie usunac polaczenie miedzy wierzchołkami

Parameters

<i>in</i>	<i>i</i>	- numer wierzchołka
<i>in</i>	<i>j</i>	- numer wierzchołka

5.2.3.6 `template<class Typ> void Graf< Typ>::_Wykonaj (unsigned int n) const [virtual]`

Metoda wywołująca testowany algorytm.

Metoda ma za zadanie wywołać zaimplementowany dla tej struktury danych algorytm

Parameters

<i>in</i>	<i>n</i>	- Wielkość problemu
-----------	----------	---------------------

Implements [ltest](#).

5.2.3.7 `template<class Typ> void Graf< Typ>::_Zaladuj (unsigned int n) [virtual]`

Metoda ładująca dane.

Metoda ma za zadanie wypełnić macierz sąsiedztw informacjami o typie połączeń między wierzchołkami

Implements [ltest](#).

5.2.3.8 `template<class Typ> void Graf< Typ>::_Zwolnij () [virtual]`

Metoda zwalniania pamięci.

Metoda ma za zadanie zwolnić pamięć, wykorzystaną przez graf

Parameters

<i>in</i>	<i>n</i>	- Wielkość problemu
-----------	----------	---------------------

Implements [ltest](#).

5.2.3.9 `template<class Typ> Typ Graf< Typ>::_ZwrocWartoscWierzcholka (const unsigned int indeks) const`

Metoda zwracająca wartość z wierzchołka.

Metoda ma za zadanie zwrócić wartość z danego wierzchołka

Parameters

<i>in</i>	<i>indeks</i>	- Indeks wierzchołka z którego ma zostać odczytana wartość
-----------	---------------	--

5.2.3.10 `template<class Typ> void Graf< Typ>::BFS (const int i, const int Szukany) const`

Algorytm BFS.

Implementacja algorytmu przechodzenia grafu wszerz, polegającego na przechodzeniu grafu od zadanego wierzchołka i odwiedzeniu wszystkich dostępnych z niego wierzchołków. Rezultatem jest drzewo przeszukiwania wszerz o korzeniu równemu wierzchołkowi od którego rozpoczęły się poszukiwania

Parameters

in	<i>i</i>	- Początkowy wierzchołek
in	<i>Szukany</i>	- Wierzchołek końcowy

5.2.3.11 `template<class Typ> void Graf< Typ >::DFS (const int x, const int Wymagany) const`

Algorytm BFS.

Implementacja przechodzenia grafu w głąb, polegająca na badaniu wszystkich krawędzi wychodzących z podanego wierzchołka. Po zbadaniu wszystkich krawędzi wychodzących z danego wierzchołka algorytm powraca do wierzchołka z którego dany wierzchołek został odwiedzony

Parameters

in	<i>i</i>	- Wierzchołek od którego zaczyna się przechodzenie
in	<i>Szukany</i>	- Wierzchołek na którym zakończone zostaje przechodzenie grafu

5.2.4 Member Data Documentation

5.2.4.1 `template<class Typ> unsigned int Graf< Typ >::_E [private]`

Pole klasy [Graf](#).

Pole zawiera informacje o ilości krawędzi

5.2.4.2 `template<class Typ> bool** Graf< Typ >::_EMacierz [private]`

Pole klasy [Graf](#).

Pole zawiera informacje o połączeniach między wierzchołkami w grafie

5.2.4.3 `template<class Typ> unsigned int Graf< Typ >::_V [private]`

Pole Klasy [Graf](#).

Pole zawiera informacje o ilości wierzchołków

5.2.4.4 `template<class Typ> Typ* Graf< Typ >::_VMacierz [private]`

Pole klasy [Graf](#).

Pole przechowuje informacje zapisane w danym wierzchołku

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

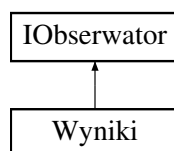
- [/home/bartolomeo/209255/prj/inc/Graf.hh](#)
- [/home/bartolomeo/209255/prj/src/Graf.cpp](#)

5.3 IObserwator Class Reference

Modeluje pojęcie interfejsu dla obserwatora.

```
#include <IObserwator.hh>
```

Inheritance diagram for IObserwator:



Public Member Functions

- virtual void [_Aktualizuj](#) ()=0
Metoda Aktualizujaca stan.

5.3.1 Detailed Description

Modeluje pojecie interfejsu dla obserwatora.

Klasa ta modeluje interfejs dla obiektu ktory bedzie obserwatorem

5.3.2 Member Function Documentation

5.3.2.1 virtual void IObserwator::_Aktualizuj () [pure virtual]

Metoda Aktualizujaca stan.

Metoda ma za zadanie poinformowac o zmianach w obiekcie ktory jest obserwowany

Implemented in [Wyniki](#).

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

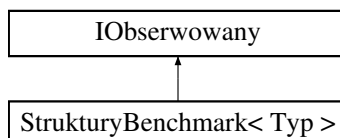
- [/home/bartolomeo/209255/prj/inc/IObserwator.hh](#)

5.4 IObservowany Class Reference

Interfejs dla Obserwatora.

```
#include <IObservowany.hh>
```

Inheritance diagram for IObservowany:



Protected Member Functions

- virtual void [_DodajObserwator](#) (IObserwator *O)=0
Metoda dodajaca obserwator.
- virtual void [_UsunObserwator](#) (IObserwator *O)=0
Metoda usuwajaca obserwator.
- virtual void [_PowiadomObserwatorow](#) ()=0
Metoda informujaca obserwatorow.

5.4.1 Detailed Description

Interfejs dla Obserwatora.

Klasa modeluje pojecie abstrakcyjnego interfejsu dla klasy bedacej obiektem obserwowanym

5.4.2 Member Function Documentation

5.4.2.1 `virtual void IObserwowany::_DodajObserwator (IObserwator * O)` `[protected]`, `[pure virtual]`

Metoda dodajaca obserwator.

Metoda ma za zadanie dodac nowego obserwatora do listy obserwatorow danego obiektu

Parameters

<code>in</code>	<code>O</code>	- wskaznik na dodawany obserwator
-----------------	----------------	-----------------------------------

Implemented in [StrukturyBenchmark< Typ >](#).

5.4.2.2 `virtual void IObserwowany::_PowiadomObserwatorow ()` `[protected]`, `[pure virtual]`

Metoda informujaca obserwatorow.

Metoda ma za zadanie poinformowac wszystkich obserwatorow o zmianach, ktore sa istotne dla nich, jakie zostaly wykonane na obiekcie obserwowanym

Implemented in [StrukturyBenchmark< Typ >](#).

5.4.2.3 `virtual void IObserwowany::_UsunObserwator (IObserwator * O)` `[protected]`, `[pure virtual]`

Metoda usuwajaca obserwator.

Metoda ma za zadanei usunac zadanego poprzez argument obserwatora z listy obserwatorow danego obiektu

Parameters

<code>in</code>	<code>O</code>	- wskaznik na obserwator,ktory ma zostac usuniety
-----------------	----------------	---

Implemented in [StrukturyBenchmark< Typ >](#).

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

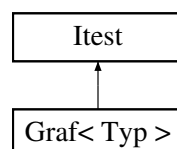
- [/home/bartolomeo/209255/prj/inc/IObserwowany.hh](#)

5.5 Itest Class Reference

Modeluje pojecie Interfejsu Testujacego.

```
#include <Itest.hh>
```

Inheritance diagram for Itest:



Public Member Functions

- virtual void [_Wykonaj](#) (unsigned int n) const =0
Metoda Wywolujaca pojedyncza operacje.
- virtual void [_Zwolnij](#) ()=0
Metoda zwalnijaca pamiec.
- virtual void [_Zaladuj](#) (const unsigned int n)=0

5.5.1 Detailed Description

Modeluje pojecie Interfejsu Testujacego.

Klasa modeluje pojecie abstrakcyjnego interfejsu pozwalajacego wczytac dane do struktury, wykonac pojedyncza operacje testujaca oraz zwolnic zajeta pamiec

5.5.2 Member Function Documentation

5.5.2.1 virtual void ltest::_Wykonaj (unsigned int *n*) const [pure virtual]

Metoda Wywolujaca pojedyncza operacje.

Metoda ma za zadanie wywolac testowana operacje

Parameters

in	<i>n</i>	- wielkosc problemu
----	----------	---------------------

Implemented in [Graf< Typ >](#).

5.5.2.2 virtual void ltest::_Zaladuj (const unsigned int *n*) [pure virtual]

Metoda wczytujaca dane

Metoda ma za zadanie wywolac metode umozliwiajaca wczytanie danych do danej struktury

Parameters

in	<i>n</i>	- Wielkosc problemu
----	----------	---------------------

Implemented in [Graf< Typ >](#).

5.5.2.3 virtual void ltest::_Zwolnij () [pure virtual]

Metoda zwalnijaca pamiec.

Metoda ma za zadanie zwolnic pamiec przeznaczona na dana strukture

Implemented in [Graf< Typ >](#).

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

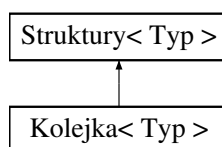
- [/home/bartolomeo/209255/prj/inc/ltest.hh](#)

5.6 Kolejka< Typ > Class Template Reference

Modeluje pojecie Kolejki.

```
#include <Kolejka.hh>
```

Inheritance diagram for Kolejka< Typ >:



Classes

- struct [Wezel](#)

Public Member Functions

- [Kolejka](#) ()
Konstruktor.
- [~Kolejka](#) ()
Destruktor.
- void [_Pokaz](#) () const
Metoda wyswietlajaca elementu Kolejki.
- void [_Push](#) (Typ k, unsigned int Pozycja=0)
Metoda dodajaca nowy [Wezel](#).
- Typ [_Pop](#) (unsigned int Pozycja=0)
Metda usuwajaca wezel.
- unsigned int [_Rozmiar](#) () const
Metoda informujaca o obecnej ilosci Wezlow.
- bool [CzyPusta](#) ()

Private Member Functions

- void [_Zwolnij](#) ()
Metoda zwalnijaca pamiec zajeta przez struktre.

Private Attributes

- [Wezel](#) * [_Pierwszy](#)
Pole klasy [Kolejka](#).
- [Wezel](#) * [_Ostatni](#)
Pole klasy [Kolejka](#).
- unsigned int [_Ilosc](#)
Pole klasy [Kolejka](#).

5.6.1 Detailed Description

```
template<class Typ>class Kolejka< Typ >
```

Modeluje pojecie Kolejki.

Klasa modeluje pojecie kolejki, dodajac nowy element na jej koniec i sciagajac pierwszy dodany element

5.6.2 Constructor & Destructor Documentation

5.6.2.1 `template<class Typ> Kolejka< Typ>::Kolejka () [inline]`

Konstruktor.

Ustawia wskazniki na NULL oraz zeruje ilosc elementow przy tworzeniu obiektow danej klasy

5.6.2.2 `template<class Typ> Kolejka< Typ>::~~Kolejka () [inline]`

Destruktor.

5.6.3 Member Function Documentation

5.6.3.1 `template<class Typ> void Kolejka< Typ>::_Pokaz () const [inline],[virtual]`

Metoda wyswietlajaca elementu Kolejki.

Metoda ma za zadanie wyswietlic wszystkie wartosci znajdujace sie w kolejce

Implements [Struktury< Typ >](#).

5.6.3.2 `template<class Typ> Typ Kolejka< Typ>::_Pop (unsigned int Pozycja = 0) [inline],[virtual]`

Metda usuwajaca wezel.

Metoda ma za zadanie usunac pierwszy dodany element z kolejki oraz ustawic wskaznik przed ostatniego elementu na NULL

Implements [Struktury< Typ >](#).

5.6.3.3 `template<class Typ> void Kolejka< Typ>::_Push (Typ k, unsigned int Pozycja = 0) [inline],[virtual]`

Metoda dodajaca nowy [Wezel](#).

Metoda ma za zadanie dodac nowy wezel na koniec kolejki, umiescic w nim warosc zadana jako argument oraz pokazac wskaznikiem na ostatni [Wezel](#) przed dodaniem nowego Wezla

Parameters

in	k	- Wartosc ktora zostanie umieszczona w odpowiednim polu Wezla
----	---	---

Implements [Struktury< Typ >](#).

5.6.3.4 `template<class Typ> unsigned int Kolejka< Typ>::_Rozmiar () const [inline],[virtual]`

Metoda informujaca o obecnej ilosci Wezlow.

Metoda zwraca informacje o ilosci obecnie znajdujacych sie elementow w kolejce

Returns

- Zwraca ilosc elementow w kolejce

Implements [Struktury< Typ >](#).

5.6.3.5 `template<class Typ> void Kolejka< Typ>::_Zwolnij () [inline], [private], [virtual]`

Metoda zwalnijająca pamięć zajęta przez struktury.

Metoda ma za zadanie zwolnić pamięć zajęta przez załadowane do struktury dane, elementy są usuwane dopóki wskaźnik pokazujący na początek kolejki nie będzie wskazywał na NULL

Implements [Struktury< Typ >](#).

5.6.3.6 `template<class Typ> bool Kolejka< Typ>::_CzyPusta () [inline]`

5.6.4 Member Data Documentation

5.6.4.1 `template<class Typ> unsigned int Kolejka< Typ>::_Ilosc [private]`

Pole klasy [Kolejka](#).

Pole przechowuje informacje o ilości obecnie znajdujących się elementów w kolejce

5.6.4.2 `template<class Typ> Wezel* Kolejka< Typ>::_Ostatni [private]`

Pole klasy [Kolejka](#).

Pole jest wskaźnikiem na ostatni element kolejki

5.6.4.3 `template<class Typ> Wezel* Kolejka< Typ>::_Pierwszy [private]`

Pole klasy [Kolejka](#).

Pole jest wskaźnikiem na pierwszy element kolejki

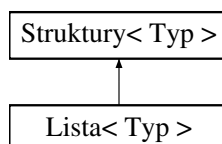
The documentation for this class was generated from the following file:

- </home/bartolomeo/209255/prj/inc/Kolejka.hh>

5.7 Lista< Typ > Class Template Reference

```
#include <Lista.hh>
```

Inheritance diagram for Lista< Typ >:



Classes

- struct [Wezel](#)

Public Member Functions

- void [_Push](#) (Typ Wart, unsigned int Poz)
Metoda dodająca [Wezel](#).

- Typ `_Pop` (unsigned int Pozycja)
Metoda usuwajaca [Wezel](#).
- unsigned int `_Rozmiar` () const
Metoda informujaca o ilosci wezlow.
- `Lista` ()
Konstruktor.
- `~Lista` ()
Destruktor Usuwa wskaznik.
- void `_Pokaz` () const
Konstruktor Kopiujacy.

Private Member Functions

- void `_Zwolnij` ()
Metoda zwalnijaca pamiec zajeta przez struktre.

Private Attributes

- `Wezel` * `Glowa`
Pole klasy [Lista](#) Wskaznik na nowo dodany [Wezel](#).
- unsigned int `_Ilosc`
Pole klasy [Lista](#) Pole przechowuje bierzaca ilosc elementow listy.

5.7.1 Constructor & Destructor Documentation

5.7.1.1 `template<class Typ> Lista< Typ >::Lista () [inline]`

Konstruktor.

Konstruktor ustawia wskaznik na NULL i zeruje ilosc wezlow listy

5.7.1.2 `template<class Typ> Lista< Typ >::~~Lista () [inline]`

Destruktor Usuwa wskaznik.

5.7.2 Member Function Documentation

5.7.2.1 `template<class Typ> void Lista< Typ >::_Pokaz () const [inline],[virtual]`

Konstruktor Kopiujacy.

Metoda wyswietlajaca elementy Listy

Metoda ma za zadanie wyswietlic wszystkie wartosci znajdujace sie na Liscie

Implements [Struktury< Typ >](#).

5.7.2.2 `template<class Typ> Typ Lista< Typ >::_Pop (unsigned int Pozycja) [inline],[virtual]`

Metoda usuwajaca [Wezel](#).

Metoda ma za zadanie usunac wezel zgodny z argumentem metody [Wezel](#) zosatnie usuniety, a sasiednie elementy zostana polaczone wskaznikiem

Parameters

in	Pozycja	- Numer Wezla ktory zostanie usuniety
----	---------	---------------------------------------

Implements [Struktury< Typ >](#).

5.7.2.3 `template<class Typ> void Lista< Typ >::_Push (Typ Wart, unsigned int Poz) [inline],[virtual]`

Metoda dodajaca [Wezel](#).

Metoda ma za zadanie dodac element do listy w zaleznosci od argumentu miejscu i usatwic wartosc zadana przez argument

Parameters

in	wart	- Wartosc jaka zostana dodana do Wezla
in	poz	- Pozycja w ktorej zosatnie dodany Wezel

Implements [Struktury< Typ >](#).

5.7.2.4 `template<class Typ> unsigned int Lista< Typ >::_Rozmiar () const [inline],[virtual]`

Metoda informujaca o ilosci wezlow.

Metoda zwraca informacje o ilosci aktualnych wezlow listy

Returns

- Ilosc elementow listy

Implements [Struktury< Typ >](#).

5.7.2.5 `template<class Typ> void Lista< Typ >::_Zwolnij () [inline],[private],[virtual]`

Metoda zwalnijaca pamiec zajeta przez struktre.

Metoda ma za zadanie zwolnij pamiec zajeta przez zaladowane do struktury dane, elementy sa usuwany dopoki wskaznik pokazujacy na poczatek listy nie bedzie wskazywal na NULL

Implements [Struktury< Typ >](#).

5.7.3 Member Data Documentation

5.7.3.1 `template<class Typ> unsigned int Lista< Typ >::_Ilosc [private]`

Pole klasy [Lista](#) Pole przechowuje bierzaca ilosc elementow listy.

5.7.3.2 `template<class Typ> Wezel* Lista< Typ >::Glowa [private]`

Pole klasy [Lista](#) Wskaznik na nowo dodany [Wezel](#).

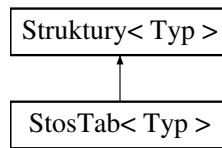
The documentation for this class was generated from the following file:

- /home/bartolomeo/209255/prj/inc/[Lista.hh](#)

5.8 StosTab< Typ > Class Template Reference

```
#include <StosTab.hh>
```

Inheritance diagram for StosTab< Typ >:



Public Member Functions

- void [_Zwolnij](#) ()
- [StosTab](#) ()
Konstruktor Podczas tworzenia obiektu tej klasy automatycznie alokowana jest tablica o rozmiarze 1 oraz ustawienie obecnej liczby elementow listy na 0.
- [StosTab](#) (const [StosTab](#) &K)
Konstruktor Kopiujacy.
- virtual [~StosTab](#) ()
Destruktor obiektu.
- void [_Pokaz](#) () const
Metoda wypisujaca elemeny Stosu.
- Typ [_Pop](#) (unsigned int Pozycja=0)
Metoda sciagajaca element ze stosu.
- void [_Push](#) (Typ k, unsigned int Pozycja=0)
Metoda dodajaca elemet do tablicy.
- unsigned int [_Rozmiar](#) () const
Metoda zwracajaca rozmiar listy.
- bool [CzyPusty](#) ()

Private Attributes

- Typ * [_L](#)
Pole klasy [StosTab](#).
- unsigned int [_RozmiarL](#)
Pole Klasy [StosTab](#).
- unsigned int [_RozmiarT](#)
Pole Klasy [StosTab](#).

5.8.1 Constructor & Destructor Documentation

5.8.1.1 `template<class Typ> StosTab< Typ>::StosTab () [inline]`

Konstruktor Podczas tworzenia obiektu tej klasy automatycznie alokowana jest tablica o rozmiarze 1 oraz ustawienie obecnej liczby elementow listy na 0.

5.8.1.2 `template<class Typ> StosTab< Typ>::StosTab (const StosTab< Typ> & K) [inline]`

Konstruktor Kopiujacy.

5.8.1.3 `template<class Typ> virtual StosTab< Typ>::~~StosTab () [inline]`

Destruktor obiektu.

5.8.2 Member Function Documentation

5.8.2.1 `template<class Typ > void StosTab< Typ >::_Pokaz () const [inline],[virtual]`

Metoda wypisujaca elementy Stosu.

Metoda ma za zadanie wypisac wszystkie elementy znajdujace sie obecnie na liscie danych

Implements [Struktury< Typ >](#).

5.8.2.2 `template<class Typ > Typ StosTab< Typ >::_Pop (unsigned int Pozycja = 0) [inline],[virtual]`

Metoda sciagajaca element ze stosu.

Metoda ma za zadanie sciagnac ostatni element stosu, w przypadku gdy tablica jest do polowy pusta nastepuje utworzenie nowej tablicy o dwa razy mniejszym rozmiarze

Parameters

in	<i>Pozycja</i>	- numer elementy kotry zostanie usuniety z listy i zostanie zwrcona jego wartosc
----	----------------	--

Returns

Zwraca wybrany przez uzytkownika element

Implements [Struktury< Typ >](#).

5.8.2.3 `template<class Typ > void StosTab< Typ >::_Push (Typ k, unsigned int Pozycja = 0) [inline],[virtual]`

Metoda dodajaca elemet do tablicy.

Metoda ma za zadanie dodac nowy element na koncu stosu, w przypadku zapelnienia tablicy nastepuje utworzenie nowej tablicy i przepisanie elementow

Parameters

in	<i>k</i>	- wartosc jaka chcemy dodac do listy
in	<i>Pozycja</i>	- Pozycja na ktorej chcemy dodac wartosc

Implements [Struktury< Typ >](#).

5.8.2.4 `template<class Typ > unsigned int StosTab< Typ >::_Rozmiar () const [inline],[virtual]`

Metoda zwracajaca rozmiar listy.

Metoda zwraca informacje o obecnej ilosci danych w strukturze

Returns

Zwraca ilosc elementow listy

Implements [Struktury< Typ >](#).

5.8.2.5 `template<class Typ > void StosTab< Typ >::_Zwolnij () [inline],[virtual]`

Metoda zwalnijajaca pamiec

Metoda ma za zadanie zwolnij pamiec zajmowana przez dane, dopoki ilosc elementow listy nie wynosi 0 wykonywana jest metoda `_Pop`, aby oproznic stos i zwolnic pamiec

Implements [Struktury< Typ >](#).

5.8.2.6 `template<class Typ> bool StosTab< Typ >::CzyPusty () [inline]`

5.8.3 Member Data Documentation

5.8.3.1 `template<class Typ> Typ* StosTab< Typ >::_L [private]`

Pole klasy [StosTab](#).

Pole zawiera wskaźnik na typ całkowity, służy do alokacji pamięci na dynamiczną tablicę

5.8.3.2 `template<class Typ> unsigned int StosTab< Typ >::_RozmiarL [private]`

Pole Klasy [StosTab](#).

Pole przechowuje informacje o ilości obecnie znajdujących się elementów na liście danych

5.8.3.3 `template<class Typ> unsigned int StosTab< Typ >::_RozmiarT [private]`

Pole Klasy [StosTab](#).

Pole przechowuje informacje o obecnym rozmiarze tablicy danych

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

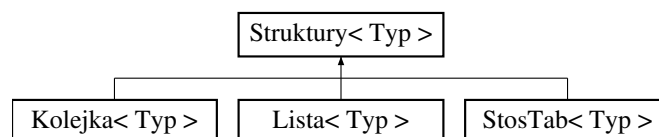
- `/home/bartolomeo/209255/prj/inc/StosTab.hh`

5.9 Struktury< Typ > Class Template Reference

Modeluje pojęcie [Struktury](#) danych, klasa bazowa dla Stosu, Kolejki i Listy, zarówno w implementacji wskaźnikowej jak i tablicowej.

```
#include <Struktury.hh>
```

Inheritance diagram for [Struktury< Typ >](#):



Public Member Functions

- virtual void [_Push](#) (Typ k, unsigned int Pozycja)=0
Metoda dodająca kolejny element struktury.
- virtual Typ [_Pop](#) (unsigned int Pozycja)=0
Metoda usuwająca element.
- virtual unsigned int [_Rozmiar](#) () const =0
Metoda zwracająca rozmiar [Struktury](#).
- virtual void [_Pokaz](#) () const =0
Metoda wyświetlająca dane.
- virtual void [_Zwolnij](#) ()=0
Metoda zwalniania pamięć.

5.9.1 Detailed Description

`template<class Typ>class Struktury< Typ >`

Modeluje pojecie [Struktury](#) danych, klasa bazowa dla Stosu,Kolejki i Listy,zarowno w implemenetacji wskaznikowej jak i tablicowej.

5.9.2 Member Function Documentation

5.9.2.1 `template<class Typ > virtual void Struktury< Typ >::_Pokaz () const` `[pure virtual]`

Metoda wyswietlajaca dane.

Metoda ma za zadanie wyswietlic wszystkie dane nalezace do struktury

Implemented in [Lista< Typ >](#), [Kolejka< Typ >](#), and [StosTab< Typ >](#).

5.9.2.2 `template<class Typ > virtual Typ Struktury< Typ >::_Pop (unsigned int Pozycja)` `[pure virtual]`

Metoda usuwajaca element.

Metoda ma za zadanie usunac element i w zaleznosci od implementowanej struktury bedzie to usuwany element usuwany z poczatku,końca lub w przypadku listy z dowolnego jej miejsca

Parameters

<code>in</code>	<i>Pozycja</i>	- Numer elementu ,ktory zostanie dodany. Argument ma znaczenie tylko w przypadku listy i domyślnie jest ustawiony, tak aby element był dodawany zawsze na początku listy
-----------------	----------------	--

Returns

Zwraca wartosc elementu z odpowiedniego dla wybranej struktury miejsca

Implemented in [Kolejka< Typ >](#), [Lista< Typ >](#), and [StosTab< Typ >](#).

5.9.2.3 `template<class Typ > virtual void Struktury< Typ >::_Push (Typ k, unsigned int Pozycja)` `[pure virtual]`

Metoda dodajaca kolejny element struktury.

Metoda ma za zadanie dodac kolejny element do naszej struktury oraz zapisac w nim odpowiednia wartosc.W zaleznosci od implementowanej struktury element bedzie dodawany na poczatku lub na koncu struktury danych.

Parameters

<code>in</code>	<i>k</i>	- wartosc typu calkowitnego, ktora bedzie umieszczona w strukturze
-----------------	----------	--

Implemented in [StosTab< Typ >](#), [Kolejka< Typ >](#), and [Lista< Typ >](#).

5.9.2.4 `template<class Typ > virtual unsigned int Struktury< Typ >::_Rozmiar () const` `[pure virtual]`

Metoda zwracajaca rozmiar [Struktury](#).

Metoda ma zadanie zwrocic bierzaca liczbe elementow nalezacych do danej struktury

Returns

- Bierzaca liczba elementow [Struktury](#) danych

Implemented in [Kolejka< Typ >](#), [StosTab< Typ >](#), and [Lista< Typ >](#).

5.9.2.5 `template<class Typ > virtual void Struktury< Typ >::_Zwolnij () [pure virtual]`

Metoda zwalnijająca pamięć.

Metoda ma za zadanie zwolnić pamięć używaną przy zapelnianiu danej struktury danymi

Implemented in [Kolejka< Typ >](#), [Lista< Typ >](#), and [StosTab< Typ >](#).

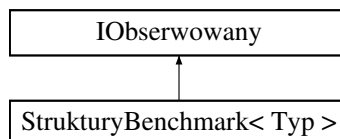
The documentation for this class was generated from the following file:

- `/home/bartolomeo/209255/prj/inc/Struktury.hh`

5.10 StrukturyBenchmark< Typ > Class Template Reference

```
#include <StrukturyBenchmark.hh>
```

Inheritance diagram for StrukturyBenchmark< Typ >:



Public Member Functions

- [StrukturyBenchmark](#) ()
- virtual [~StrukturyBenchmark](#) ()
- [StrukturyBenchmark](#) (const unsigned int Proby, const unsigned int Powt, const unsigned int *Rozmiary)
Konstruktor obiektu.
- void [_WykonajTest](#) (Itest &W, unsigned int n)
Metoda inicjalizująca test.
- void [_DodajObserwator](#) (IObservator *O)
Metoda dodająca obserwator.
- void [_UsunObserwator](#) (IObservator *O)
Metoda usuwająca obserwator.

Private Member Functions

- void [_PowiadomObserwatorow](#) ()
Metoda wykonująca test dla odpowiedniej struktury.

Private Attributes

- std::list< [IObservator](#) * > [Obserwatorzy](#)
Pole StrukturyBenchmark.
- unsigned int [_IloscProb](#)
Pole StrukturyBenchmark.
- unsigned int [_IloscPowt](#)
Pole StrukturyBenchmark.
- unsigned int * [_TablicaRozmiarow](#)
Pole StrukturyBenchmark.

Additional Inherited Members

5.10.1 Detailed Description

```
template<class Typ>class StrukturyBenchmark< Typ >
```

Klasa modeluje pojecie Benchmarku przeznaczonego dla struktur danych przechowujace dane

5.10.2 Constructor & Destructor Documentation

5.10.2.1 `template<class Typ > StrukturyBenchmark< Typ >::StrukturyBenchmark () [inline]`

5.10.2.2 `template<class Typ > virtual StrukturyBenchmark< Typ >::~~StrukturyBenchmark () [inline], [virtual]`

5.10.2.3 `template<class Typ > StrukturyBenchmark< Typ >::StrukturyBenchmark (const unsigned int Proby, const unsigned int Powt, const unsigned int * Rozmiary) [inline]`

Konstruktor obiektu.

5.10.3 Member Function Documentation

5.10.3.1 `template<class Typ > void StrukturyBenchmark< Typ >::_DodajObserwator (IObservator * O) [inline], [virtual]`

Metoda dodajaca obserwator.

Metoda ma za zadanie dodac nowego obserwatora do listy obserwatorow danego obiektu

Parameters

<i>in</i>	<i>O</i>	- wskaznik na dodawany obserwator
-----------	----------	-----------------------------------

Implements [IObservowany](#).

5.10.3.2 `template<class Typ > void StrukturyBenchmark< Typ >::_PowiadomObserwatorow () [inline], [private], [virtual]`

Metoda wykonujaca test dla odpowiedniej struktury.

Metoda ma za zadanie wykonac zapelnienie struktury danymi o zadanej w argumencie ilosci

Parameters

<i>in</i>	<i>n</i>	- ilosc danych ktora zapelnona struktura
-----------	----------	--

Metoda informujaca obserwatorow

Metoda ma za zadanie poinformowac wszystkich obserwatorow o zmianach, ktore sa istotne dla nich, jakie zostaly wykonane na obiekcie obserwowanym

Implements [IObservowany](#).

5.10.3.3 `template<class Typ > void StrukturyBenchmark< Typ >::_UsunObserwator (IObservator * O) [inline], [virtual]`

Metoda usuwajaca obserwator.

Metoda ma za zadanei usunac zadanego poprzez argument obserwatora z listy obserwatorow danego obiektu

Parameters

in	O	- wskaźnik na obserwator, który ma zostać usunięty
----	---	--

Implements [IObserwowany](#).

5.10.3.4 `template<class Typ > void StrukturyBenchmark< Typ >::_WykonajTest (Itest & W, unsigned int n)`
[inline]

Metoda inicjalizująca test.

Metoda ma za zadanie uruchomić określoną ilość razy testowaną metodę, czas jej wykonania jest zbierany przez klasę zewnętrzną

5.10.4 Member Data Documentation

5.10.4.1 `template<class Typ > unsigned int StrukturyBenchmark< Typ >::_IloscPowt` [private]

Pole [StrukturyBenchmark](#).

Pole zawiera informacje o ilości powtórzeń jakie mają zostać wykonane przy teście

5.10.4.2 `template<class Typ > unsigned int StrukturyBenchmark< Typ >::_IloscProb` [private]

Pole [StrukturyBenchmark](#).

Pole zawiera informacje o ilości prob jakie zostaną wykonane

5.10.4.3 `template<class Typ > unsigned int* StrukturyBenchmark< Typ >::_TablicaRozmiarow` [private]

Pole [StrukturyBenchmark](#).

Pole zawiera wskaźnik przechowujący informacje dla jakiej ilości danych mają zostać wykonane testy

5.10.4.4 `template<class Typ > std::list<IObserwator*> StrukturyBenchmark< Typ >::_Obserwatorzy`
[private]

Pole [StrukturyBenchmark](#).

Pole zawiera listę obserwatorów, które obserwują ten obiekt

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

- </home/bartolomeo/209255/prj/inc/StrukturyBenchmark.hh>

5.11 Kolejka< Typ >::Wezel Struct Reference

Public Attributes

- [Wezel](#) * [_Nast](#)
Pole Wezła Pole będące wskaźnikiem na kolejny element Kolejki.
- Typ [_Wartosc](#)
Pole Wezła Pole przechowuje wartość typu całkowitego.

5.11.1 Detailed Description

template<class Typ>struct Kolejka< Typ >::Wezel

Klasy [Kolejka](#) Pole modelje pojecie wezla,bedacego podstawa dla implementacji struktury danych w rozwinieniu wskaznikowym

5.11.2 Member Data Documentation

5.11.2.1 template<class Typ > Wezel* Kolejka< Typ >::Wezel::_Nast

Pole Wezla Pole bedace wskaznikiem na kolejny element Kolejki.

5.11.2.2 template<class Typ > Typ Kolejka< Typ >::Wezel::_Wartosc

Pole Wezla Pole przechowuje wartosc typu calkowitego.

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

- </home/bartolomeo/209255/prj/inc/Kolejka.hh>

5.12 Lista< Typ >::Wezel Struct Reference

Public Attributes

- [Wezel](#) * [_Nast](#)
Pole Wezla Pole bedace wskaznikiem na kolejny element listy.
- Typ [_Wartosc](#)
Pole Wezla Pole przechowuje wartosc typu calkowitego.

5.12.1 Detailed Description

template<class Typ>struct Lista< Typ >::Wezel

Klasy lista Pole modelje pojecie wezla,bedacego podstawa dla implementacji struktury danych w rozwinieniu wskaznikowym

5.12.2 Member Data Documentation

5.12.2.1 template<class Typ > Wezel* Lista< Typ >::Wezel::_Nast

Pole Wezla Pole bedace wskaznikiem na kolejny element listy.

5.12.2.2 template<class Typ > Typ Lista< Typ >::Wezel::_Wartosc

Pole Wezla Pole przechowuje wartosc typu calkowitego.

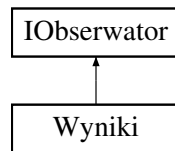
The documentation for this struct was generated from the following file:

- </home/bartolomeo/209255/prj/inc/Lista.hh>

5.13 Wyniki Class Reference

```
#include <Wyniki.hh>
```

Inheritance diagram for Wyniki:



Public Member Functions

- [Wyniki](#) ()
- [Wyniki](#) (const unsigned int Powtorzen, const unsigned int Proby, unsigned int *Rozmiary)
- virtual [~Wyniki](#) ()
- void [_ZapiszWyniki](#) (std::string PlikWy) const
- void [_Aktualizuj](#) ()

Metoda Aktualizujaca stan.

Private Attributes

- unsigned int [_IloscProb](#)
- unsigned int [_IloscPowtorzen](#)
- unsigned int * [_TablicaRozmiarow](#)
- long double * [_TablicaWynikow](#)
- [Czasomierz Stoper](#)

5.13.1 Constructor & Destructor Documentation

5.13.1.1 [Wyniki::Wyniki](#) ()

5.13.1.2 [Wyniki::Wyniki](#) (const unsigned int *Powtorzen*, const unsigned int *Proby*, unsigned int * *Rozmiary*)

5.13.1.3 [Wyniki::~~Wyniki](#) () [virtual]

5.13.2 Member Function Documentation

5.13.2.1 void [Wyniki::_Aktualizuj](#) () [virtual]

Metoda Aktualizujaca stan.

Metoda ma za zadanie poinformowac o zmianach w obiekcie ktory jest obserwowany

Implements [IObserwator](#).

5.13.2.2 void [Wyniki::_ZapiszWyniki](#) (std::string *PlikWy*) const

5.13.3 Member Data Documentation

5.13.3.1 unsigned int [Wyniki::_IloscPowtorzen](#) [private]

5.13.3.2 unsigned int Wyniki::_IloscProb [private]

5.13.3.3 unsigned int* Wyniki::_TablicaRozmiarow [private]

5.13.3.4 long double* Wyniki::_TablicaWynikow [private]

5.13.3.5 Czasomierz Wyniki::Stoper [private]

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

- /home/bartolomeo/209255/prj/inc/[Wyniki.hh](#)
- /home/bartolomeo/209255/prj/src/[Wyniki.cpp](#)

Chapter 6

File Documentation

6.1 /home/bartolomeo/209255/prj/doc/strona-glowna.dox File Reference

6.2 /home/bartolomeo/209255/prj/inc/Czasomierz.hh File Reference

```
#include <iostream>
#include <cstdlib>
#include <fstream>
#include <cstring>
```

Classes

- class [Czasomierz](#)
Modeluje pojecie Czasomierza.

6.3 /home/bartolomeo/209255/prj/inc/Graf.hh File Reference

```
#include <iostream>
#include "Kolejka.hh"
#include "Lista.hh"
#include "StosTab.hh"
#include "Itest.hh"
#include "../src/Graf.cpp"
```

Classes

- class [Graf< Typ >](#)
Modeluje pojecie grafu.

6.4 /home/bartolomeo/209255/prj/inc/IObserwator.hh File Reference

Classes

- class [IObserwator](#)

Modeluje pojecie interfejsu dla obserwatora.

6.5 /home/bartolomeo/209255/prj/inc/IObserwowany.hh File Reference

```
#include <iostream>
#include "IObserwator.hh"
```

Classes

- class [IObserwowany](#)
Interfejs dla Obserwatora.

6.6 /home/bartolomeo/209255/prj/inc/ltest.hh File Reference

Classes

- class [ltest](#)
Modeluje pojecie Interfejsu Testujacego.

6.7 /home/bartolomeo/209255/prj/inc/Kolejka.hh File Reference

```
#include <iostream>
#include <cstdlib>
#include "Struktury.hh"
```

Classes

- class [Kolejka< Typ >](#)
Modeluje pojecie Kolejki.
- struct [Kolejka< Typ >::Wezel](#)

6.8 /home/bartolomeo/209255/prj/inc/Lista.hh File Reference

```
#include <iostream>
#include "Struktury.hh"
```

Classes

- class [Lista< Typ >](#)
- struct [Lista< Typ >::Wezel](#)

6.9 /home/bartolomeo/209255/prj/inc/StosTab.hh File Reference

```
#include <cstdlib>
#include <iostream>
#include "Struktury.hh"
```

Classes

- class [StosTab](#)< Typ >

6.10 /home/bartolomeo/209255/prj/inc/Struktury.hh File Reference

```
#include <iostream>
#include <cstring>
#include <sstream>
#include <fstream>
```

Classes

- class [Struktury](#)< Typ >

Modeluje pojecie [Struktury](#) danych, klasa bazowa dla Stosu, Kolejki i Listy, zarowno w implemenetacji wskaznikowej jak i tablicowej.

6.11 /home/bartolomeo/209255/prj/inc/StrukturyBenchmark.hh File Reference

```
#include <iostream>
#include <cstdlib>
#include <fstream>
#include <cstring>
#include <list>
#include "IObservowany.hh"
#include "Itest.hh"
```

Classes

- class [StrukturyBenchmark](#)< Typ >

6.12 /home/bartolomeo/209255/prj/inc/Wyniki.hh File Reference

```
#include <iostream>
#include "IObservator.hh"
#include "Czasomierz.hh"
#include <fstream>
#include <cstdlib>
#include <string>
```

Classes

- class [Wyniki](#)

6.13 /home/bartolomeo/209255/prj/src/Czasomierz.cpp File Reference

```
#include "Czasomierz.hh"
```

6.14 /home/bartolomeo/209255/prj/src/Graf.cpp File Reference

6.15 /home/bartolomeo/209255/prj/src/Main.cpp File Reference

funkcja glowna programu

```
#include "StrukturyBenchmark.hh"
#include "Wyniki.hh"
#include "Graf.hh"
```

Functions

- int [main](#) ()

Variables

- const unsigned int [ILOSC_POW](#) = 10
- const unsigned int [ILOSC_PROB](#) = 7

6.15.1 Detailed Description

funkcja glowna programu

6.15.2 Function Documentation

6.15.2.1 int main ()

6.15.3 Variable Documentation

6.15.3.1 const unsigned int ILOSC_POW = 10

6.15.3.2 const unsigned int ILOSC_PROB = 7

6.16 /home/bartolomeo/209255/prj/src/Wyniki.cpp File Reference

```
#include "Wyniki.hh"
```

Index

- ~Graf
 - Graf, [13](#)
- ~Kolejka
 - Kolejka, [20](#)
- ~Lista
 - Lista, [22](#)
- ~StosTab
 - StosTab, [24](#)
- ~StrukturyBenchmark
 - StrukturyBenchmark, [29](#)
- ~Wyniki
 - Wyniki, [32](#)
- /home/bartolomeo/209255/prj/doc/strona-glowna.dox, [35](#)
- /home/bartolomeo/209255/prj/inc/Czasomierz.hh, [35](#)
- /home/bartolomeo/209255/prj/inc/Graf.hh, [35](#)
- /home/bartolomeo/209255/prj/inc/IObserwator.hh, [35](#)
- /home/bartolomeo/209255/prj/inc/IObserwowany.hh, [36](#)
- /home/bartolomeo/209255/prj/inc/ltest.hh, [36](#)
- /home/bartolomeo/209255/prj/inc/Kolejka.hh, [36](#)
- /home/bartolomeo/209255/prj/inc/Lista.hh, [36](#)
- /home/bartolomeo/209255/prj/inc/StosTab.hh, [37](#)
- /home/bartolomeo/209255/prj/inc/Struktury.hh, [37](#)
- /home/bartolomeo/209255/prj/inc/StrukturyBenchmark.-hh, [37](#)
- /home/bartolomeo/209255/prj/inc/Wyniki.hh, [37](#)
- /home/bartolomeo/209255/prj/src/Czasomierz.cpp, [38](#)
- /home/bartolomeo/209255/prj/src/Graf.cpp, [38](#)
- /home/bartolomeo/209255/prj/src/Main.cpp, [38](#)
- /home/bartolomeo/209255/prj/src/Wyniki.cpp, [38](#)
- _Aktualizuj
 - IObserwator, [16](#)
 - Wyniki, [32](#)
- _AktualizujCzas
 - Czasomierz, [10](#)
- _Aktualny
 - Czasomierz, [11](#)
- _CzasTrwania
 - Czasomierz, [10](#)
- _CzyKrawedz
 - Graf, [13](#)
- _DodajKrawedz
 - Graf, [13](#)
- _DodajObserwator
 - IObserwowany, [17](#)
 - StrukturyBenchmark, [29](#)
- _DodajWartoscWierzcholka
 - Graf, [13](#)
- _E
 - Graf, [15](#)
- _EMacierz
 - Graf, [15](#)
- _Ilosc
 - Kolejka, [21](#)
 - Lista, [23](#)
- _IloscKrawedzi
 - Graf, [13](#)
- _IloscPowt
 - StrukturyBenchmark, [30](#)
- _IloscPowtorzen
 - Wyniki, [32](#)
- _IloscProb
 - StrukturyBenchmark, [30](#)
 - Wyniki, [32](#)
- _Koniec
 - Czasomierz, [11](#)
- _L
 - StosTab, [26](#)
- _Nast
 - Kolejka::Wezel, [31](#)
 - Lista::Wezel, [31](#)
- _Ostatni
 - Kolejka, [21](#)
- _Pierwszy
 - Kolejka, [21](#)
- _PojedynczyPomiar
 - Czasomierz, [10](#)
- _Pokaz
 - Kolejka, [20](#)
 - Lista, [22](#)
 - StosTab, [25](#)
 - Struktury, [27](#)
- _Pop
 - Kolejka, [20](#)
 - Lista, [22](#)
 - StosTab, [25](#)
 - Struktury, [27](#)
- _PowiadomObserwatorow
 - IObserwowany, [17](#)
 - StrukturyBenchmark, [29](#)
- _Push
 - Kolejka, [20](#)
 - Lista, [23](#)
 - StosTab, [25](#)
 - Struktury, [27](#)
- _Reset
 - Czasomierz, [10](#)
- _Rozmiar

- Kolejka, 20
- Lista, 23
- StosTab, 25
- Struktury, 27
- _RozmiarL
 - StosTab, 26
- _RozmiarT
 - StosTab, 26
- _RozpocznijPomiar
 - Czasomierz, 10
- _Start
 - Czasomierz, 11
- _Status
 - Czasomierz, 11
- _StatusPracy
 - Czasomierz, 10
- _TablicaRozmiarow
 - StrukturyBenchmark, 30
 - Wyniki, 33
- _TablicaWynikow
 - Wyniki, 33
- _UsunKrawedz
 - Graf, 13
- _UsunObserwator
 - IObserwowany, 17
 - StrukturyBenchmark, 29
- _V
 - Graf, 15
- _VMacierz
 - Graf, 15
- _Wartosc
 - Kolejka::Wezel, 31
 - Lista::Wezel, 31
- _Wykonaj
 - Graf, 14
 - Ittest, 18
- _WykonajTest
 - StrukturyBenchmark, 30
- _ZakoncZPomiar
 - Czasomierz, 10
- _Zaladuj
 - Graf, 14
 - Ittest, 18
- _ZapiszWyniki
 - Wyniki, 32
- _Zwolnij
 - Graf, 14
 - Ittest, 18
 - Kolejka, 20
 - Lista, 23
 - StosTab, 25
 - Struktury, 27
- _ZwrocWartoscWierzcholka
 - Graf, 14
- BFS
 - Graf, 14
- Czasomierz, 9
 - _AktualizujCzas, 10
 - _Aktualny, 11
 - _CzasTrwania, 10
 - _Koniec, 11
 - _PojedynczyPomiar, 10
 - _Reset, 10
 - _RozpocznijPomiar, 10
 - _Start, 11
 - _Status, 11
 - _StatusPracy, 10
 - _ZakoncZPomiar, 10
 - Czasomierz, 10
- CzyPusta
 - Kolejka, 21
- CzyPusty
 - StosTab, 26
- DFS
 - Graf, 15
- Glowa
 - Lista, 23
- Graf
 - ~Graf, 13
 - _CzyKrawedz, 13
 - _DodajKrawedz, 13
 - _DodajWartoscWierzcholka, 13
 - _E, 15
 - _EMacierz, 15
 - _IloscKrawedzi, 13
 - _UsunKrawedz, 13
 - _V, 15
 - _VMacierz, 15
 - _Wykonaj, 14
 - _Zaladuj, 14
 - _Zwolnij, 14
 - _ZwrocWartoscWierzcholka, 14
 - BFS, 14
 - DFS, 15
 - Graf, 12
- Graf < Typ >, 11
- ILOSC_POW
 - Main.cpp, 38
- ILOSC_PROB
 - Main.cpp, 38
- IObserwator, 15
 - _Aktualizuj, 16
- IObserwowany, 16
 - _DodajObserwator, 17
 - _PowiadomObserwatorow, 17
 - _UsunObserwator, 17
- Ittest, 17
 - _Wykonaj, 18
 - _Zaladuj, 18
 - _Zwolnij, 18
- Kolejka
 - ~Kolejka, 20

- [_Ilosc, 21](#)
 - [_Ostatni, 21](#)
 - [_Pierwszy, 21](#)
 - [_Pokaz, 20](#)
 - [_Pop, 20](#)
 - [_Push, 20](#)
 - [_Rozmiar, 20](#)
 - [_Zwolnij, 20](#)
 - [CzyPusta, 21](#)
 - [Kolejka, 20](#)
- [Kolejka< Typ >, 18](#)
- [Kolejka< Typ >::Wezel, 30](#)
- [Kolejka::Wezel](#)
 - [_Nast, 31](#)
 - [_Wartosc, 31](#)
- [Lista](#)
 - [~Lista, 22](#)
 - [_Ilosc, 23](#)
 - [_Pokaz, 22](#)
 - [_Pop, 22](#)
 - [_Push, 23](#)
 - [_Rozmiar, 23](#)
 - [_Zwolnij, 23](#)
 - [Glowa, 23](#)
 - [Lista, 22](#)
- [Lista< Typ >, 21](#)
- [Lista< Typ >::Wezel, 31](#)
- [Lista::Wezel](#)
 - [_Nast, 31](#)
 - [_Wartosc, 31](#)
- [main](#)
 - [Main.cpp, 38](#)
- [Main.cpp](#)
 - [ILOS_C POW, 38](#)
 - [ILOS_C PROB, 38](#)
 - [main, 38](#)
- [Obserwatorzy](#)
 - [StrukturyBenchmark, 30](#)
- [Stoper](#)
 - [Wyniki, 33](#)
- [StosTab](#)
 - [~StosTab, 24](#)
 - [_L, 26](#)
 - [_Pokaz, 25](#)
 - [_Pop, 25](#)
 - [_Push, 25](#)
 - [_Rozmiar, 25](#)
 - [_RozmiarL, 26](#)
 - [_RozmiarT, 26](#)
 - [_Zwolnij, 25](#)
 - [CzyPusty, 26](#)
 - [StosTab, 24](#)
 - [StosTab, 24](#)
- [StosTab< Typ >, 23](#)
- [Struktury](#)
 - [_Pokaz, 27](#)
 - [_Pop, 27](#)
 - [_Push, 27](#)
 - [_Rozmiar, 27](#)
 - [_Zwolnij, 27](#)
- [Struktury< Typ >, 26](#)
- [StrukturyBenchmark](#)
 - [~StrukturyBenchmark, 29](#)
 - [_DodajObserwator, 29](#)
 - [_IloscPowt, 30](#)
 - [_IloscProb, 30](#)
 - [_PowiadomObserwatorow, 29](#)
 - [_TablicaRozmiarow, 30](#)
 - [_UsunObserwator, 29](#)
 - [_WykonajTest, 30](#)
 - [Obserwatorzy, 30](#)
 - [StrukturyBenchmark, 29](#)
 - [StrukturyBenchmark, 29](#)
- [StrukturyBenchmark< Typ >, 28](#)
- [Wyniki, 32](#)
 - [~Wyniki, 32](#)
 - [_Aktualizuj, 32](#)
 - [_IloscPowtorzen, 32](#)
 - [_IloscProb, 32](#)
 - [_TablicaRozmiarow, 33](#)
 - [_TablicaWynikow, 33](#)
 - [_ZapiszWyniki, 32](#)
 - [Stoper, 33](#)
 - [Wyniki, 32](#)