

PAMSI LAB VII

Generated by Doxygen 1.8.6

Sun May 17 2015 19:32:34

Contents

1	Hierarchical Index	1
1.1	Class Hierarchy	1
2	Class Index	3
2.1	Class List	3
3	File Index	5
3.1	File List	5
4	Class Documentation	7
4.1	BenchmarkInterfejs Class Reference	7
4.1.1	Detailed Description	7
4.1.2	Member Function Documentation	7
4.1.2.1	_Generator	7
4.1.2.2	_Test	8
4.1.2.3	_Wczytaj	8
4.1.2.4	_Zaladuj	8
4.1.2.5	_Zwolnij	8
4.2	Czasomierz Class Reference	8
4.2.1	Detailed Description	9
4.2.2	Constructor & Destructor Documentation	9
4.2.2.1	Czasomierz	9
4.2.3	Member Function Documentation	9
4.2.3.1	_AktualizujCzas	9
4.2.3.2	_CzasTrwania	9
4.2.3.3	_PojedynczyPomiar	10
4.2.3.4	_Reset	10
4.2.3.5	_RozpocznijPomiar	10
4.2.3.6	_StatusPracy	10
4.2.3.7	_ZakonczPomiar	10
4.2.4	Member Data Documentation	10
4.2.4.1	_Aktualny	10

4.2.4.2	_Koniec	10
4.2.4.3	_Start	10
4.2.4.4	_Status	11
4.3	HSort< Typ > Class Template Reference	11
4.3.1	Detailed Description	11
4.3.2	Member Function Documentation	11
4.3.2.1	_Sort	11
4.3.2.2	BudujKopiec	12
4.3.2.3	Kopiec	12
4.3.2.4	SortowanieKopiec	12
4.4	HybSort< Typ > Class Template Reference	12
4.4.1	Detailed Description	13
4.4.2	Member Function Documentation	13
4.4.2.1	_Sort	13
4.4.2.2	Mediana	13
4.4.2.3	Partycjowanie	13
4.4.2.4	Sortowanie_Hybrydowe	14
4.4.2.5	Wstaw_Sort	14
4.5	IObserwator Class Reference	14
4.5.1	Detailed Description	15
4.5.2	Member Function Documentation	15
4.5.2.1	_Aktualizuj	15
4.6	IObserwowany Class Reference	15
4.6.1	Detailed Description	15
4.6.2	Member Function Documentation	15
4.6.2.1	_DodajObserwator	15
4.6.2.2	_PowiadomObserwatorow	16
4.6.2.3	_UsunObserwator	16
4.7	ISortable< Typ > Class Template Reference	16
4.7.1	Detailed Description	16
4.7.2	Member Function Documentation	17
4.7.2.1	_Sort	17
4.8	Iterable< Typ > Class Template Reference	17
4.8.1	Detailed Description	17
4.8.2	Member Function Documentation	18
4.8.2.1	_Rozmiar	18
4.8.2.2	_Zamien	18
4.8.2.3	Adres	18
4.8.2.4	Wartosc	18
4.9	ListArr2x< Typ > Class Template Reference	18

4.9.1	Detailed Description	19
4.9.2	Constructor & Destructor Documentation	20
4.9.2.1	ListArr2x	20
4.9.3	Member Function Documentation	20
4.9.3.1	_Pokaz	20
4.9.3.2	_Pop	20
4.9.3.3	_Push	20
4.9.3.4	_Rozmiar	20
4.9.3.5	_Zamien	21
4.9.3.6	_Zwolnij	22
4.9.3.7	Adres	22
4.9.3.8	Wartosc	22
4.9.3.9	WczytajDane	22
4.9.4	Member Data Documentation	22
4.9.4.1	RozmiarL	22
4.9.4.2	RozmiarT	23
4.9.4.3	tab	23
4.10	MSort< Typ > Class Template Reference	23
4.10.1	Detailed Description	23
4.10.2	Member Function Documentation	23
4.10.2.1	_Sort	24
4.10.2.2	Merge	25
4.10.2.3	Msort	25
4.11	QSort< Typ > Class Template Reference	25
4.11.1	Detailed Description	26
4.11.2	Member Function Documentation	26
4.11.2.1	_Sort	26
4.11.2.2	Partycjowanie	26
4.11.2.3	Qsort	26
4.12	QSortOpt< Typ > Class Template Reference	27
4.12.1	Detailed Description	27
4.12.2	Member Function Documentation	27
4.12.2.1	_Sort	27
4.12.2.2	Mediana	28
4.12.2.3	Partycjowanie	28
4.12.2.4	QsortOpt	28
4.13	StosTab< Typ > Class Template Reference	28
4.13.1	Constructor & Destructor Documentation	29
4.13.1.1	StosTab	29
4.13.1.2	StosTab	29

4.13.1.3	~StosTab	29
4.13.2	Member Function Documentation	29
4.13.2.1	_Pokaz	29
4.13.2.2	_Pop	30
4.13.2.3	_Push	30
4.13.2.4	_Rozmiar	30
4.13.2.5	_Zamien	30
4.13.2.6	_Zwolnij	30
4.13.2.7	Adres	31
4.13.2.8	Wartosc	31
4.13.3	Member Data Documentation	31
4.13.3.1	_L	31
4.13.3.2	_RozmiarL	31
4.13.3.3	_RozmiarT	31
4.14	Struktury< Typ > Class Template Reference	31
4.14.1	Detailed Description	32
4.14.2	Member Function Documentation	32
4.14.2.1	_Pokaz	32
4.14.2.2	_Pop	32
4.14.2.3	_Push	33
4.14.2.4	_Rozmiar	33
4.14.2.5	_Zwolnij	33
4.15	StrukturyBenchmark< Typ > Class Template Reference	33
4.15.1	Detailed Description	35
4.15.2	Constructor & Destructor Documentation	35
4.15.2.1	StrukturyBenchmark	35
4.15.3	Member Function Documentation	35
4.15.3.1	_DodajObserwator	35
4.15.3.2	_Generator	35
4.15.3.3	_PowiadomObserwatorow	35
4.15.3.4	_Test	35
4.15.3.5	_Ustaw	36
4.15.3.6	_UsunObserwator	36
4.15.3.7	_Wczytaj	36
4.15.3.8	_WykonajTest	36
4.15.3.9	_Zaladuj	36
4.15.3.10	_Zwolnij	37
4.15.4	Member Data Documentation	37
4.15.4.1	_IloscDanych	37
4.15.4.2	_IloscPowt	37

4.15.4.3	_IloscProb	37
4.15.4.4	_TablicaRozmiarow	37
4.15.4.5	_Wartosci	37
4.15.4.6	I	37
4.15.4.7	Obserwatorzy	37
4.15.4.8	S	37
4.15.4.9	T	38
4.16	Wyniki Class Reference	38
4.16.1	Constructor & Destructor Documentation	38
4.16.1.1	Wyniki	38
4.16.1.2	~Wyniki	38
4.16.2	Member Function Documentation	38
4.16.2.1	_Aktualizuj	38
4.16.2.2	_ZapiszWyniki	39
4.16.3	Member Data Documentation	39
4.16.3.1	_IloscPowtorzen	39
4.16.3.2	_IloscProb	39
4.16.3.3	_TablicaRozmiarow	39
4.16.3.4	_TablicaWynikow	39
4.16.3.5	Stoper	39
5	File Documentation	41
5.1	/home/bartolomeo/209296/prj/inc/BenchmarkInterfejs.hh File Reference	41
5.1.1	Macro Definition Documentation	41
5.1.1.1	DLUGOSC_SLOWA	41
5.2	/home/bartolomeo/209296/prj/inc/Czasomierz.hh File Reference	41
5.3	/home/bartolomeo/209296/prj/inc/HSort.hh File Reference	42
5.4	/home/bartolomeo/209296/prj/inc/HybSort.hh File Reference	42
5.4.1	Macro Definition Documentation	42
5.4.1.1	ILE	42
5.4.1.2	PROG	42
5.5	/home/bartolomeo/209296/prj/inc/IObserwator.hh File Reference	42
5.6	/home/bartolomeo/209296/prj/inc/IObserwowany.hh File Reference	42
5.7	/home/bartolomeo/209296/prj/inc/ISortable.hh File Reference	43
5.8	/home/bartolomeo/209296/prj/inc/IStruktury.hh File Reference	43
5.9	/home/bartolomeo/209296/prj/inc/IIterable.hh File Reference	43
5.10	/home/bartolomeo/209296/prj/inc/ListArr2x.hh File Reference	43
5.10.1	Detailed Description	44
5.11	/home/bartolomeo/209296/prj/inc/MSort.hh File Reference	44
5.12	/home/bartolomeo/209296/prj/inc/QSort.hh File Reference	44

5.13	/home/bartolomeo/209296/prj/inc/QSortOpt.hh File Reference	44
5.13.1	Macro Definition Documentation	45
5.13.1.1	ILE	45
5.14	/home/bartolomeo/209296/prj/inc/StosTab.hh File Reference	45
5.15	/home/bartolomeo/209296/prj/inc/StrukturyBenchmark.hh File Reference	45
5.16	/home/bartolomeo/209296/prj/inc/Wyniki.hh File Reference	45
5.17	/home/bartolomeo/209296/prj/src/Czasomierz.cpp File Reference	45
5.18	/home/bartolomeo/209296/prj/src/Main.cpp File Reference	46
5.18.1	Detailed Description	46
5.18.2	Macro Definition Documentation	46
5.18.2.1	ILOSC_POW	46
5.18.2.2	ILOSC_PROB	46
5.18.3	Function Documentation	46
5.18.3.1	main	46
5.18.4	Variable Documentation	46
5.18.4.1	Tablica_Rozmiarow	46
5.19	/home/bartolomeo/209296/prj/src/Wyniki.cpp File Reference	46
Index		47

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

BenchmarkInterfejs	7
StrukturyBenchmark< Typ >	33
Czasomierz	8
IObserwator	14
Wyniki	38
IObserwowany	15
StrukturyBenchmark< Typ >	33
ISortable< Typ >	16
HSort< Typ >	11
HybSort< Typ >	12
MSort< Typ >	23
QSort< Typ >	25
QSortOpt< Typ >	27
Iterable< Typ >	17
ListArr2x< Typ >	18
StosTab< Typ >	28
Struktury< Typ >	31
ListArr2x< Typ >	18
StosTab< Typ >	28

Chapter 2

Class Index

2.1 Class List

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

BenchmarkInterfejs	
Modeluje pojecie Interfejsu Benchmark'u	7
Czasomierz	
Modeluje pojecie Czasomierza	8
HSort< Typ >	
Modeluje sortowanie przez kopcowanie	11
HybSort< Typ >	
Modeluje sortowania hybrydowego	12
IObservator	
Modeluje pojecie interfejsu dla obserwatora	14
IObservowany	
Interfejs dla Obserwatora	15
ISortable< Typ >	
Definicja klasy ISortable	16
Iterable< Typ >	
Interfejs Iterable	17
ListArr2x< Typ >	
Modeluje pojecie Listy (array)	18
MSort< Typ >	
Modeluje sortowanie przez scalanie	23
QSort< Typ >	
Modeluje sortowanie szybkie	25
QSortOpt< Typ >	
Modeluje sortowanie szybkie z optymalizacja	27
StosTab< Typ >	28
Struktury< Typ >	
Modeluje pojecie Struktury danych, klasa bazowa dla Stosu, Kolejki i Listy, zarowno w implemencie- tacji wskaznikowej jak i tablicowej	31
StrukturyBenchmark< Typ >	33
Wyniki	38

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

/home/bartolomeo/209296/prj/inc/BenchmarkInterfejs.hh	41
/home/bartolomeo/209296/prj/inc/Czasomierz.hh	41
/home/bartolomeo/209296/prj/inc/HSort.hh	42
/home/bartolomeo/209296/prj/inc/HybSort.hh	42
/home/bartolomeo/209296/prj/inc/IObserwator.hh	42
/home/bartolomeo/209296/prj/inc/IObserwowany.hh	42
/home/bartolomeo/209296/prj/inc/ISortable.hh	43
/home/bartolomeo/209296/prj/inc/IStruktury.hh	43
/home/bartolomeo/209296/prj/inc/Iterable.hh	43
/home/bartolomeo/209296/prj/inc/ListArr2x.hh	
Definicja klasy ListArr1	43
/home/bartolomeo/209296/prj/inc/MSort.hh	44
/home/bartolomeo/209296/prj/inc/QSort.hh	44
/home/bartolomeo/209296/prj/inc/QSortOpt.hh	44
/home/bartolomeo/209296/prj/inc/StosTab.hh	45
/home/bartolomeo/209296/prj/inc/StrukturyBenchmark.hh	45
/home/bartolomeo/209296/prj/inc/Wyniki.hh	45
/home/bartolomeo/209296/prj/src/Czasomierz.cpp	45
/home/bartolomeo/209296/prj/src/Main.cpp	
Funkcja glowna programu	46
/home/bartolomeo/209296/prj/src/Wyniki.cpp	46

Chapter 4

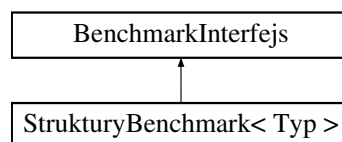
Class Documentation

4.1 BenchmarkInterfejs Class Reference

Modeluje pojecie Interfejsu Benchmark'u.

```
#include <BenchmarkInterfejs.hh>
```

Inheritance diagram for BenchmarkInterfejs:



Protected Member Functions

- virtual void `_Test ()` const =0
Metoda Wykonujaca pojedyncza operacje.
- virtual void `_Wczytaj (string PlikWart)=0`
Metoda wczytujaca dane z pliku Metoda ma za zadanie wczytac dane z pliku wejscowego.
- virtual void `_Zaladuj (const unsigned int n)` const =0
Metoda wypelniajaca Metoda ma za zadanie wypelnic dany kontener danymi.
- virtual void `_Zwolnij ()=0`
Metoda zwalnijaca Pamiec.
- virtual void `_Generator ()` const =0
Metoda generujaca dane.

4.1.1 Detailed Description

Modeluje pojecie Interfejsu Benchmark'u.

Klasa bazowa dla implementowania benchmarku dla kolejnych struktur danych

4.1.2 Member Function Documentation

4.1.2.1 `virtual void BenchmarkInterfejs::_Generator () const` `[protected]`, `[pure virtual]`

Metoda generujaca dane.

Metoda ma za zadanie wygenerowac pseudolosowe dane i zapisac je do pliku

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

4.1.2.2 `virtual void BenchmarkInterfejs::_Test () const` `[protected]`, `[pure virtual]`

Metoda Wykonujaca pojedyncza operacje.

Metoda ma za zadanie wykonan pojedyncza operacja, ktorej czas jest rejestrowany

Parameters

<code>in</code>	<code>llosc</code>	- Liczba danych poddana testowi
-----------------	--------------------	---------------------------------

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

4.1.2.3 `virtual void BenchmarkInterfejs::_Wczytaj (string PlikWart)` `[protected]`, `[pure virtual]`

Metoda wczytujaca dane z pliku Metoda ma za zadanie wczytac dane z pliku wejscowego.

Parameters

<code>in</code>	<code>PlikIn</code>	- Nazwa pliku wejscowego
<code>in</code>	<code>n</code>	- liczba wczytywanych danych

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

4.1.2.4 `virtual void BenchmarkInterfejs::_Zaladuj (const unsigned int n) const` `[protected]`, `[pure virtual]`

Metoda wypelniajaca Metoda ma za zadanie wypelnic dany kontener danymi.

Parameters

<code>in</code>	<code>n</code>	- ilosc danych
-----------------	----------------	----------------

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

4.1.2.5 `virtual void BenchmarkInterfejs::_Zwolnij ()` `[protected]`, `[pure virtual]`

Metoda zwalnijaca Pamiec.

Metoda ma za zadanie zwolnic pamiec przeznaczona na dane przechowywane w kontenerze

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

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

- `/home/bartolomeo/209296/prj/inc/BenchmarkInterfejs.hh`

4.2 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](#).

4.2.1 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

4.2.2 Constructor & Destructor Documentation

4.2.2.1 [Czasomierz::Czasomierz](#) ()

Konstruktor obiektu.

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

4.2.3 Member Function Documentation

4.2.3.1 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

4.2.3.2 long double [Czasomierz::_CzasTrwania](#) () const

Metoda zwracajaca calkowity czas Proby.

Metoda ma za zadanie zwrocic calkowity czas przypadajacy na dana probe

4.2.3.3 double Czasomierz::_PojedynczyPomiar () const

Metoda zwracająca czas pojedynczego pomiaru.

Metoda ma za zadanie zwrócić czas trwania pojedynczego testu

4.2.3.4 void Czasomierz::_Reset ()

Metoda resetująca [Czasomierz](#).

Metoda ma za zadanie wyzerować pola przechowujące informacje o czasie systemowym zapisanym na początku i na końcu pomiaru. Zmienia status czasomierza na wolny

4.2.3.5 void Czasomierz::_RozpocznijPomiar ()

Metoda zaczynająca pomiar.

Metoda ma za zadanie odczytać czas systemowy w momencie jej wywołania i zapisania tej informacji w odpowiednim polu klasy. Zmienia również status pracy czasomierza na zajęty.

4.2.3.6 bool Czasomierz::_StatusPracy () const

informująca o stanie czasomierza

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

4.2.3.7 void Czasomierz::_ZakoncZPomiar ()

Metoda kończąca pomiar.

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

4.2.4 Member Data Documentation

4.2.4.1 long double Czasomierz::_Aktualny [private]

Pole klasy [Czasomierz](#).

Pole przechowuje czas w ms, odpowiadający czasowi aktualnego pomiaru dla danej ilości danych

4.2.4.2 clock_t Czasomierz::_Koniec [private]

Pole klasy [Czasomierz](#).

Pole przechowuje informacje, która jest czas systemu w chwili zakończenia pomiaru

4.2.4.3 clock_t Czasomierz::_Start [private]

Pole klasy [Czasomierz](#).

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

4.2.4.4 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:

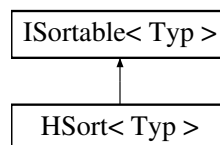
- </home/bartolomeo/209296/prj/inc/Czasomierz.hh>
- </home/bartolomeo/209296/prj/src/Czasomierz.cpp>

4.3 HSort< Typ > Class Template Reference

Modeluje sortowanie przez kopcowanie.

```
#include <HSort.hh>
```

Inheritance diagram for HSort< Typ >:



Public Member Functions

- void [_Sort](#) ([Iterable](#)< Typ > *Kontener)
Metoda inicjalizująca sortowanie przez kopcowanie.

Private Member Functions

- void [SortowanieKopiec](#) (int Rozmiar, [Iterable](#)< Typ > *K)
Metoda sortowania przez Kopcowanie.
- void [Kopiec](#) (int Rozmiar, const int i, [Iterable](#)< Typ > *K)
Metoda składająca kopiec.
- void [BudujKopiec](#) (int Rozmiar, [Iterable](#)< Typ > *K)
Metoda tworząca kopiec.

4.3.1 Detailed Description

```
template<class Typ>class HSort< Typ >
```

Modeluje sortowanie przez kopcowanie.

Klasa zawiera implementacje algorytmu sortowania przez kopcowanie

4.3.2 Member Function Documentation

4.3.2.1 template<class Typ> void HSort< Typ >::_Sort ([Iterable](#)< Typ > * *Kontener*) [inline], [virtual]

Metoda inicjalizująca sortowanie przez kopcowanie.

Metoda ma za zadanie zainicjalizować algorytm sortowania przez kopcowanie dla wybranej struktury danych

Parameters

in	<i>Kontener</i>	- rodzaj kontenera, który zostanie posortowany
----	-----------------	--

Implements [ISortable< Typ >](#).

4.3.2.2 `template<class Typ> void HSort< Typ >::BudujKopiec (int Rozmiar, Iterable< Typ > * K)` [inline],
[private]

Metoda tworząca kopiec.

Metoda ma za zadanie utworzyć abstrakcyjny kopiec z tablicy o podanym poprzez argument rozmiarze

Parameters

in	<i>Rozmiar</i>	- Rozmiar kopca
----	----------------	-----------------

4.3.2.3 `template<class Typ> void HSort< Typ >::Kopiec (int Rozmiar, const int i, Iterable< Typ > * K)` [inline],
[private]

Metoda składająca kopiec.

Metoda ma za zadanie poprzez porównywanie i ustawianie elementów odtworzenie porządku kopcowego.

Parameters

in	<i>Rozmiar</i>	- Rozmiar kopca
in	<i>i</i>	- indeks ostatniego elementu podzbioru

4.3.2.4 `template<class Typ> void HSort< Typ >::SortowanieKopiec (int Rozmiar, Iterable< Typ > * K)` [inline],
[private]

Metoda sortowania przez Kopcowanie.

Metoda realizująca sortowanie rosnące, wykorzystując przy tym kopiec.

Parameters

in	<i>Rozmiar</i>	- Rozmiar kopca do zbudowania, ilość danych do posortowania.
----	----------------	--

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

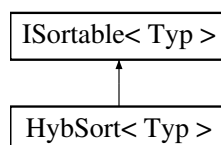
- </home/bartolomeo/209296/prj/inc/HSort.hh>

4.4 HybSort< Typ > Class Template Reference

Modeluje sortowania hybrydowego.

```
#include <HybSort.hh>
```

Inheritance diagram for HybSort< Typ >:



Public Member Functions

- void [_Sort](#) ([Iterable](#)< Typ > *Kontener)
Metoda inicjalizująca sortowanie hybrydowe.

Private Member Functions

- int [Mediana](#) (Typ *W) const
Mediana Metoda wyznaczająca medianę dla tablicy 3 elementowej. Jest to metoda pomocnicza, wykorzystywana przy optymalizacji doboru pivotu w sortowaniu szybkim.
- int [Partycjonowanie](#) (int p, int k, [Iterable](#)< Typ > *K)
Zotymalizowane Sortowanie Szybkie.
- void [Wstaw_Sort](#) (int l, int p, [Iterable](#)< Typ > *K)
Sortowanie przez Wstawianie Metoda ma za zadanie posortować tablice przyjmowaną jako argument.
- void [Sortowanie_Hybrydowe](#) (int l, int h, [Iterable](#)< Typ > *K)
Metoda sortowania hybrydowego.

4.4.1 Detailed Description

template<class Typ>class HybSort< Typ >

Modeluje sortowania hybrydowego.

Klasa zawiera implementację algorytmu sortowania hybrydowego

4.4.2 Member Function Documentation

4.4.2.1 template<class Typ > void HybSort< Typ >::_Sort ([Iterable](#)< Typ > * *Kontener*) [inline],[virtual]

Metoda inicjalizująca sortowanie hybrydowe.

Metoda ma za zadanie zainicjalizować algorytm sortowania hybrydowego dla wybranej struktury danych

Parameters

in	<i>Kontener</i>	- rodzaj kontenera, który zostanie posortowany
----	-----------------	--

Implements [ISortable](#)< Typ >.

4.4.2.2 template<class Typ > int HybSort< Typ >::Mediana (Typ * W) const [inline],[private]

Mediana Metoda wyznaczająca medianę dla tablicy 3 elementowej. Jest to metoda pomocnicza, wykorzystywana przy optymalizacji doboru pivotu w sortowaniu szybkim.

Returns

Zwraca indeks na którym znajduje się mediana w tablicy wejściowej

4.4.2.3 template<class Typ > int HybSort< Typ >::Partycjonowanie (int p, int k, [Iterable](#)< Typ > * K) [inline],[private]

Zotymalizowane Sortowanie Szybkie.

Metoda modeluje algorytm sortowania szybkiego z zaimplementowanym algorytmem doboru pivotu, tak aby nie został wybrany najmniejszy element w danym podzbiorze.

[in] lewy - początkowy indeks podzbioru

Parameters

in	<i>prawy</i>	- koncowy indeks podzbioru
----	--------------	----------------------------

4.4.2.4 `template<class Typ > void HybSort< Typ >::Sortowanie_Hybrydowe (int l, int h, Iterable< Typ > * K)`
`[inline], [private]`

Metoda sortowania hybrydowego.

Metoda ta jest implementacja algorytmu sortowania hybrydowego, bedacego polaczeniem sortowania szybkiego i sortowania przez wstawianie Po zakonczeniu rekurencyjnych wywolan Partycjowania, tablica jest podzielona na szereg malych podzbiorow o o rozmiarze nie przekraczajacemu ustalonego progu. Zbioru sa porozdzielana elementami ktore wykorzystywane byly jako elementy osiowe. Dla czesciowo posortowanej tablicy wywolywane jest sortowanie przez wstawianie, ktore jest wydajne dla tablic o malych rozmiarach

Parameters

in	<i>l</i>	- indeks poczatkowego elementu pozbioru
in	<i>h</i>	- indeks koncowego elementu podzbioru

4.4.2.5 `template<class Typ > void HybSort< Typ >::Wstaw_Sort (int l, int p, Iterable< Typ > * K)` `[inline],`
`[private]`

Sortowanie przez Wstawianie Metoda ma za zadanie posortowac tablice przyjmowana jako argument.

Parameters

in	<i>T</i>	- Wskaznik na tablice z danymi wejsciowymi
in	<i>l</i>	- Poczatkowy indeks elementu sortowanego podzbioru
in	<i>p</i>	- koncowy indeks elementu sortowanego podzbioru

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

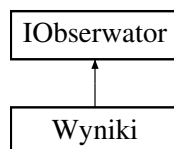
- </home/bartolomeo/209296/prj/inc/HybSort.hh>

4.5 IObservator Class Reference

Modeluje pojecie interfejsu dla obserwatora.

```
#include <IObservator.hh>
```

Inheritance diagram for IObservator:



Public Member Functions

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

4.5.1 Detailed Description

Modeluje pojecie interfejsu dla obserwatora.

Klasa ta modeluje interfejs dla obiektu który bedzie obserwatorem

4.5.2 Member Function Documentation

4.5.2.1 `virtual void IObservator::_Aktualizuj () [pure virtual]`

Metoda Aktualizujaca stan.

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

Implemented in [Wyniki](#).

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

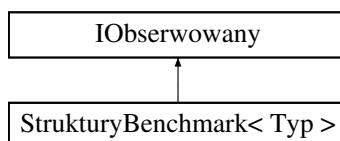
- [/home/bartolomeo/209296/prj/inc/IObservator.hh](#)

4.6 IObservowany Class Reference

Interfejs dla Obserwatora.

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

Inheritance diagram for IObservowany:



Protected Member Functions

- `virtual void _DodajObserwator (IObservator *O)=0`
Metoda dodajaca obserwator.
- `virtual void _UsunObserwator (IObservator *O)=0`
Metoda usuwajaca obserwator.
- `virtual void _PowiadomObserwatorow ()=0`
Metoda informujaca obserwatorow.

4.6.1 Detailed Description

Interfejs dla Obserwatora.

Klasa modeluje pojecie abstrakcyjnego interfejsu dla klasy bedacej obiektem obserwowanym

4.6.2 Member Function Documentation

4.6.2.1 `virtual void IObservowany::_DodajObserwator (IObservator * O) [protected],[pure virtual]`

Metoda dodajaca obserwator.

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

Parameters

in	O	- wskaźnik na dodawany obserwator
----	---	-----------------------------------

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

4.6.2.2 `virtual void IObservowany::_PowiadomObserwatorow () [protected],[pure virtual]`

Metoda informująca obserwatorów.

Metoda ma za zadanie poinformować wszystkich obserwatorów o zmianach, które są istotne dla nich, jakie zostały wykonane na obiekcie obserwowanym.

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

4.6.2.3 `virtual void IObservowany::_UsunObserwator (IObservator * O) [protected],[pure virtual]`

Metoda usuwająca obserwator.

Metoda ma za zadanie usunąć zadanego poprzez argument obserwatora z listy obserwatorów danego obiektu.

Parameters

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

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

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

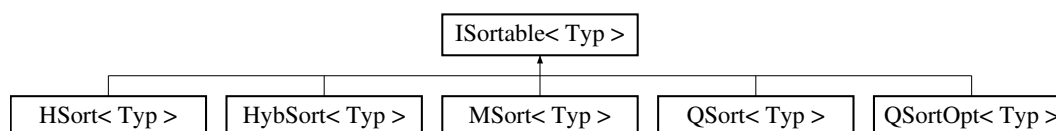
- [/home/bartolomeo/209296/prj/inc/IObservowany.hh](#)

4.7 ISortable< Typ > Class Template Reference

Definicja klasy [ISortable](#).

```
#include <ISortable.hh>
```

Inheritance diagram for `ISortable< Typ >`:



Public Member Functions

- `virtual void _Sort (Iterable< Typ > *Kontener)=0`
Metoda inicjalizująca sortowanie.

4.7.1 Detailed Description

```
template<class Typ>class ISortable< Typ >
```

Definicja klasy [ISortable](#).

Klasa modeluje pojęcie abstrakcyjnego interfejsu [ISortable](#), dzięki któremu użytkownik może wykorzystać dowolny algorytm sortowania dla wybranego konteneru.

4.7.2 Member Function Documentation

4.7.2.1 `template<class Typ> virtual void ISortable< Typ >::_Sort (Iterable< Typ > * Kontener) [pure virtual]`

Metoda inicjalizująca sortowanie.

Metoda ma za zadanie uruchomić dowolny zaimplementowany algorytm sortowania dla struktury danych określonej poprzez argument jaki przyjmuje ta metoda

Parameters

<code>in</code>	<code>Kontener</code>	- wskaźnik na obiekt który ma ulec posortowaniu
-----------------	-----------------------	---

Implemented in [HybSort< Typ >](#), [QSortOpt< Typ >](#), [HSort< Typ >](#), [MSort< Typ >](#), and [QSort< Typ >](#).

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

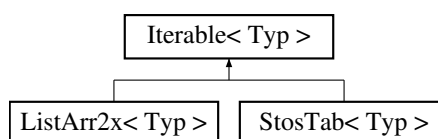
- `/home/bartolomeo/209296/prj/inc/ISortable.hh`

4.8 Iterable< Typ > Class Template Reference

Interfejs [Iterable](#).

```
#include <Iterable.hh>
```

Inheritance diagram for Iterable< Typ >:



Public Member Functions

- `virtual const Typ Wartosc (unsigned int i) const =0`
Metoda zwracająca wartość.
- `virtual Typ & Adres (unsigned int i)=0`
Metoda zwracająca referencje.
- `virtual void _Zamien (unsigned int i, unsigned int j)=0`
Metoda zamieniająca.
- `virtual unsigned int _Rozmiar () const =0`
Metoda informująca o rozmiarze.

4.8.1 Detailed Description

```
template<class Typ>class Iterable< Typ >
```

Interfejs [Iterable](#).

Klasa modeluje pojęcie abstrakcyjnego interfejsu iterable, dzięki któremu algorytmu sortujące mają dostęp do kontenera z danymi poprzez odczyt danego pola i możliwości zamiany elementów miejscami

4.8.2 Member Function Documentation

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

Metoda informujaca o rozmiarze.

Metoda ma za zadanie zwrocic informajce o rozmiarze kontenera

Implemented in [ListArr2x< Typ >](#), and [StosTab< Typ >](#).

4.8.2.2 `template<class Typ> virtual void Iterable< Typ >::_Zamien (unsigned int i, unsigned int j)` [pure virtual]

Metoda zamieniajaca.

Metoda ma za zadanie zamienic elementy miejscami

Parameters

in	<i>i</i>	- indeks,pod którym wartosc z konteneru zostanie zamieniona miejscem
in	<i>j</i>	- indeks,pod którym wartosc z konteneru zostanie zamieniona miejscem

Implemented in [StosTab< Typ >](#), and [ListArr2x< Typ >](#).

4.8.2.3 `template<class Typ> virtual Typ& Iterable< Typ >::Adres (unsigned int i)` [pure virtual]

Metoda zwracajaca referencje.

Metoda ma za zadanie zwrocic referencje do pola kontenera zadanego poprzez argument metody

Implemented in [ListArr2x< Typ >](#), and [StosTab< Typ >](#).

4.8.2.4 `template<class Typ> virtual const Typ Iterable< Typ >::Wartosc (unsigned int i) const` [pure virtual]

Metoda zwracajaca wartosc.

Metoda ma za zadanei zwrocic wartosc,kryjaca sie pod danym indeksem dla dowolnego kontenera posiadajacego interfejs.

Parameters

in	<i>i</i>	-Indeks z ktorego zostanie odczytana wartosc
----	----------	--

Returns

Zwraca wartosc, kryjaca sie pod danym indeksem konteneru

Implemented in [ListArr2x< Typ >](#), and [StosTab< Typ >](#).

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

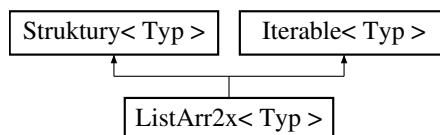
- [/home/bartolomeo/209296/prj/inc/Iterable.hh](#)

4.9 ListArr2x< Typ > Class Template Reference

Modeluje pojęcie Listy (array)

```
#include <ListArr2x.hh>
```

Inheritance diagram for ListArr2x< Typ >:



Public Member Functions

- [ListArr2x](#) ()
Konstruktor bezargumentowy.
- void [_Push](#) (Typ dana, unsigned int pole)
Dodaje element do ListyArr1.
- Typ [_Pop](#) (unsigned int pole)
Pobiera element z ListyArr1.
- unsigned int [_Rozmiar](#) () const
Wielkość listy.
- void [WczytajDane](#) (const std::string PlikIn, unsigned int n)
Wczytuje dane z pliku.
- void [_Pokaz](#) () const
Metoda wypisująca elementy listy.
- const Typ [Wartosc](#) (unsigned int i) const
Metoda zwracająca wartość.
- Typ & [Adres](#) (unsigned int i)
Metoda zwracająca referencje.

Private Member Functions

- void [_Zamien](#) (unsigned int i, unsigned int j)
Metoda zamieniająca Metoda ma za zadanie zamienić miejscami elementy wybrane przez argumenty wywołania.
- void [_Zwolnij](#) ()
Zwalnia pamięć

Private Attributes

- Typ * [tab](#)
Wskaźnik na dynamiczną tablicę
- unsigned int [RozmiarT](#)
Rozmiar tablicy.
- unsigned int [RozmiarL](#)
Rozmiar Listy.

4.9.1 Detailed Description

```
template<class Typ>class ListArr2x< Typ >
```

Modeluje pojęcie Listy (array)

Modeluje pojęcie Listy opartej na dynamicznej tablicy. Dodając elementy zwiększa tablicę dwukrotnie, jeżeli brakuje miejsca.

4.9.2 Constructor & Destructor Documentation

4.9.2.1 `template<class Typ> ListArr2x< Typ >::ListArr2x () [inline]`

Konstruktor bezargumentowy.

Konstruktor alokujący tablicę jednoelementową z której będzie tworzona lista

4.9.3 Member Function Documentation

4.9.3.1 `template<class Typ> void ListArr2x< Typ >::_Pokaz () const [inline],[virtual]`

Metoda wypisująca elementy listy.

Metoda ma za zadanie wypisać wszystkie elementy znajdujące się obecnie na liście danych

Implements [Struktury< Typ >](#).

4.9.3.2 `template<class Typ> Typ ListArr2x< Typ >::_Pop (unsigned int pole) [inline],[virtual]`

Pobiera element z ListyArr1.

Pobiera element z ListyArr2x usuwając go z niej i zmniejszając rozmiar o połowę w przypadku przekroczenia stosunku 1:4 (RozmiarL:RozmiarT)

param[in] - *pole* - nr pola z którego chcemy pobrać element (indeksowane od 0)

retval - zwraca wartość pobranej danej lub '-1' w przypadku błędu

Implements [Struktury< Typ >](#).

4.9.3.3 `template<class Typ> void ListArr2x< Typ >::_Push (Typ dana, unsigned int pole) [inline],[virtual]`

Dodaje element do ListyArr1.

Dodaje nowy element do ListyArr1

Parameters

<i>in</i>	<i>dana</i>	- element który chcemy umieścić na liście
<i>in</i>	<i>pole</i>	- nr pola na którym chcemy umieścić element jeżeli chcesz umieścić na początku listy podaj wartość 0, na końcu wartość size()

Implements [Struktury< Typ >](#).

4.9.3.4 `template<class Typ> unsigned int ListArr2x< Typ >::_Rozmiar () const [inline],[virtual]`

Wielkość listy.

Informuje o ilości elementów znajdujących się na LiścieArr1

Return values

-	zwraca liczbę elementów ListyArr1
---	-----------------------------------

Implements [Struktury< Typ >](#).

```
4.9.3.5  template<class Typ > void ListArr2x< Typ >::_Zamien ( unsigned int i, unsigned int j ) [inline],  
        [private],[virtual]
```

Metoda zamieniająca Metoda ma za zadanie zamienić miejscami elementy wybrane przez argumenty wywołania.

Parameters

in	<i>i</i>	- Adres elementu podlegający zamianie
in	<i>j</i>	- Adres elementu podlegający zamianie

Implements [Iterable< Typ >](#).

4.9.3.6 `template<class Typ > void ListArr2x< Typ >::_Zwolnij ()` `[inline],[private],[virtual]`

Zwalnia pamięć

Zwalnia pamięć zaalokowaną przez ListArr1

Implements [Struktury< Typ >](#).

4.9.3.7 `template<class Typ > Typ& ListArr2x< Typ >::Adres (unsigned int i)` `[inline],[virtual]`

Metoda zwracająca referencje.

Metoda ma za zadanie zwrócić referencje do pola kontenera zadanego poprzez argument metody

Implements [Iterable< Typ >](#).

4.9.3.8 `template<class Typ > const Typ ListArr2x< Typ >::Wartosc (unsigned int i) const` `[inline],[virtual]`

Metoda zwracająca wartość.

Metoda ma za zadanie zwrócić wartość, kryjącą się pod danym indeksem dla dowolnego kontenera posiadającego interfejs.

Parameters

in	<i>i</i>	-Indeks z którego zostanie odczytana wartość
----	----------	--

Returns

Zwraca wartość, kryjącą się pod danym indeksem konteneru

Implements [Iterable< Typ >](#).

4.9.3.9 `template<class Typ > void ListArr2x< Typ >::WczytajDane (const std::string PlikIn, unsigned int n)` `[inline]`

Wczytuje dane z pliku.

Wczytuje dane z pliku do ListArr1

param[in] nazwaPliku - nazwa pliku z danymi param[in] n - ilość danych do wczytania, 0 oznacza wszystkie dane z pliku

4.9.4 Member Data Documentation

4.9.4.1 `template<class Typ > unsigned int ListArr2x< Typ >::RozmiarL` `[private]`

Rozmiar Listy.

Aktualny rozmiar ListyArr2x

4.9.4.2 `template<class Typ> unsigned int ListArr2x< Typ >::RozmiarT` [private]

Rozmiar tablicy.

Aktualny rozmiar tablicy.

4.9.4.3 `template<class Typ> Typ* ListArr2x< Typ >::tab` [private]

Wskaźnik na dynamiczną tablicę

Wskaźnik na dynamiczną tablicę tworzącą ListęArr2x

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

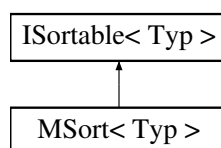
- </home/bartolomeo/209296/prj/inc/ListArr2x.hh>

4.10 MSort< Typ > Class Template Reference

Modeluje sortowanie przez scalanie.

```
#include <MSort.hh>
```

Inheritance diagram for MSort< Typ >:



Public Member Functions

- `void _Sort (Iterable< Typ > *Kontener)`
Metoda inicjalizująca sortowanie przez scalanie.

Private Member Functions

- `void Msort (Typ *T, int p, int k, Iterable< Typ > *K)`
Metoda Dzielaca tablice.
- `void Merge (Typ *Temp, int l, int s, int p, Iterable< Typ > *K)`
Metoda Dzielaca tablice.

4.10.1 Detailed Description

```
template<class Typ>class MSort< Typ >
```

Modeluje sortowanie przez scalanie.

Klasa zawiera implementacje algorytmu sortowania przez scalanie

4.10.2 Member Function Documentation

4.10.2.1 `template<class Typ> void MSort< Typ>::_Sort (Iterable< Typ> * Kontener) [inline],[virtual]`

Metoda inicjalizująca sortowanie przez scalanie.

Metoda ma za zadanie zainicjalizować algorytm sortowania przez scalanie dla wybranej struktury danych

Parameters

<code>in</code>	<i>Kontener</i>	- rodzaj kontenera, który zostanie posortowany
-----------------	-----------------	--

Implements [ISortable< Typ >](#).

4.10.2.2 `template<class Typ> void MSort< Typ >::Merge (Typ * Temp, int l, int s, int p, Iterable< Typ > * K)`
`[inline], [private]`

Metoda Dzielaca tablice.

Metoda ma za zadanie przekopiowac zawartosc zbioru glownego do tablicy tymczasowej. Następnie operując na kopii ustawia wskaźniki na początki kolejnych zbiorów i porównywane są wskazywane wartości. Mniejsze wpisujemy do zbioru glownego i przesuwamy odpowiedni wskaźnik. Czynnosc wykonujemy rekurencyjnie az do momentu gdy jeden ze wskaźników osiągnie koniec zbioru

Parameters

<code>in</code>	<i>Temp</i>	- Wskaźnik na tablice pomocnicza
<code>in</code>	<i>l</i>	- Początkowy indeks tablicy
<code>in</code>	<i>s</i>	- Środkowy indeks tablicy
<code>in</code>	<i>p</i>	- Koncowy indks tablicy

4.10.2.3 `template<class Typ> void MSort< Typ >::Msort (Typ * T, int p, int k, Iterable< Typ > * K)` `[inline], [private]`

Metoda Dzielaca tablice.

Metoda ma za zadanie przekopiowac zawartosc zbioru glownego do tablicy tymczasowej. Następnie operując na kopii ustawia wskaźniki na początki kolejnych zbiorów i porównywane są wskazywane wartości. Mniejsze wpisujemy do zbioru glownego i przesuwamy odpowiedni wskaźnik. Czynnosc wykonujemy rekurencyjnie az do momentu gdy jeden ze wskaźników osiągnie koniec zbioru

Parameters

<code>in</code>	<i>Temp</i>	- Wskaźnik na tablice pomocnicza
<code>in</code>	<i>l</i>	- Początkowy indeks tablicy
<code>in</code>	<i>s</i>	- Środkowy indeks tablicy
<code>in</code>	<i>p</i>	- Koncowy indks tablicy

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

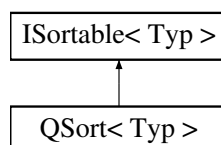
- </home/bartolomeo/209296/prj/inc/MSort.hh>

4.11 QSort< Typ > Class Template Reference

Modeluje sortowanie szybkie.

```
#include <QSort.hh>
```

Inheritance diagram for QSort< Typ >:



Public Member Functions

- void `_Sort` (`Iterable`< `Typ` > *`Kontener`)
Metoda inicjalizująca sortowanie szybkie.

Private Member Functions

- void `Qsort` (int `l`, int `h`, `Iterable`< `Typ` > *`K`)
Metoda będąca implementacją algorytmu sortowania szybkiego.
- int `Partycjowanie` (int `p`, int `k`, `Iterable`< `Typ` > *`K`)
Metoda segregująca.

4.11.1 Detailed Description

`template<class Typ>class QSort< Typ >`

Modeluje sortowanie szybkie.

Klasa zawiera implementacje algorytmu sortowania szybkiego

4.11.2 Member Function Documentation

4.11.2.1 `template<class Typ > void QSort< Typ >::_Sort (Iterable< Typ > * Kontener)` `[inline],[virtual]`

Metoda inicjalizująca sortowanie szybkie.

Metoda ma za zadanie zainicjalizować algorytm sortowania szybkiego dla wybranej struktury danych

Parameters

<code>in</code>	<code>Kontener</code>	- rodzaj kontenera, który zostanie posortowany
-----------------	-----------------------	--

Implements `ISortable< Typ >`.

4.11.2.2 `template<class Typ > int QSort< Typ >::Partycjowanie (int p, int k, Iterable< Typ > * K)` `[inline],[private]`

Metoda segregująca.

Metoda ma za zadanie wybór elementu, który ma być użyty do podziału i przenosi wszystkie elementy mniejsze na lewo od tego elementu, a większe elementy na prawo od wybranego elementu

Parameters

<code>in</code>	<code>p</code>	- początkowy indeks podzbioru
<code>in</code>	<code>k</code>	- końcowy indeks podzbioru

Returns

4.11.2.3 `template<class Typ > void QSort< Typ >::Qsort (int l, int h, Iterable< Typ > * K)` `[inline],[private]`

Metoda będąca implementacją algorytmu sortowania szybkiego.

Parameters

<code>in</code>	<code>l</code>	- początkowy indeks tablicy
<code>in</code>	<code>h</code>	- końcowy indeks tablicy

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

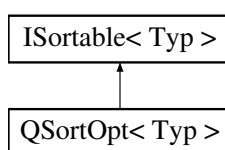
- </home/bartolomeo/209296/prj/inc/QSort.hh>

4.12 QSortOpt< Typ > Class Template Reference

Modeluje sortowanie szybkie z optymalizacją.

```
#include <QSortOpt.hh>
```

Inheritance diagram for QSortOpt< Typ >:



Public Member Functions

- void [_Sort](#) ([Iterable](#)< Typ > *Kontener)
Metoda inicjalizująca sortowanie szybkie z optymalizacją.

Private Member Functions

- int [Partycjonowanie](#) (int p, int k, [Iterable](#)< Typ > *K)
Zoptymalizowane Sortowanie Szybkie.
- void [QsortOpt](#) (int lewy, int prawy1, [Iterable](#)< Typ > *K)
- int [Mediana](#) (Typ *W) const
Mediana Metoda wyznaczająca medianę dla tablicy 3 elementowej. Jest to metoda pomocnicza, wykorzystywana przy optymalizacji doboru pivotu w sortowaniu szybkim.

4.12.1 Detailed Description

```
template<class Typ>class QSortOpt< Typ >
```

Modeluje sortowanie szybkie z optymalizacją.

Klasa zawiera implementację algorytmu sortowania szybkiego z optymalizacją

4.12.2 Member Function Documentation

4.12.2.1 `template<class Typ > void QSortOpt< Typ >::_Sort (Iterable< Typ > * Kontener) [inline], [virtual]`

Metoda inicjalizująca sortowanie szybkie z optymalizacją.

Metoda ma za zadanie zainicjalizować algorytm sortowania szybkiego z optymalizacją dla wybranej struktury danych

Parameters

<i>in</i>	<i>Kontener</i>	- rodzaj kontenera, który zostanie posortowany
-----------	-----------------	--

Implements [ISortable< Typ >](#).

4.12.2.2 `template<class Typ> int QSortOpt< Typ >::Mediana (Typ * W) const` `[inline], [private]`

Mediana Metoda wyznaczająca medianę dla tablicy 3 elementowej. Jest to metoda pomocnicza, wykorzystywana przy optymalizacji doboru pivotu w sortowaniu szybkim.

Returns

Zwraca indeks na którym znajduje się mediana w tablicy wejściowej

4.12.2.3 `template<class Typ> int QSortOpt< Typ >::Partycjowanie (int p, int k, Iterable< Typ > * K)` `[inline], [private]`

Zoptymalizowane Sortowanie Szybkie.

Metoda modeluje algorytm sortowania szybkiego z zaimplementowanym algorytmem doboru pivotu, tak aby nie został wybrany najmniejszy element w danym podzbiorze.

[in] lewy - początkowy indeks podzbioru

Parameters

<i>in</i>	<i>prawy</i>	- końcowy indeks podzbioru
-----------	--------------	----------------------------

4.12.2.4 `template<class Typ> void QSortOpt< Typ >::QsortOpt (int lewy, int prawy1, Iterable< Typ > * K)` `[inline], [private]`

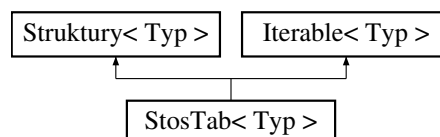
The documentation for this class was generated from the following file:

- </home/bartolomeo/209296/prj/inc/QSortOpt.hh>

4.13 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 elementów listy na 0.

- [StosTab](#) (const [StosTab](#) &K)
Konstruktor Kopiujacy.
- virtual [~StosTab](#) ()
Destruktor obiektu.
- void [_Pokaz](#) () const
Metoda wypisujaca elementy Stosu.
- Typ [_Pop](#) (unsigned int Pozycja=0)
Metoda sciagajaca element ze stosu.
- void [_Push](#) (Typ k, unsigned int Pozycja=0)
Metoda dodajaca element do tablicy.
- unsigned int [_Rozmiar](#) () const
Metoda zwracajaca rozmiar listy.
- const Typ [Wartosc](#) (unsigned int Index) const
Metoda zwracajaca wartosc.
- Typ & [Adres](#) (unsigned int Index)
Metoda zwracajaca referencje.
- void [_Zamien](#) (unsigned int i, unsigned int j)
Metoda zamieniajaca Metoda ma za zadanie zamienic miejscami elementy wybrane przez argumenty wywołania.

Private Attributes

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

4.13.1 Constructor & Destructor Documentation

4.13.1.1 `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.

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

Konstruktor Kopiujacy.

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

Destruktor obiektu.

4.13.2 Member Function Documentation

4.13.2.1 `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 >](#).

4.13.2.2 `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 >](#).

4.13.2.3 `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 >](#).

4.13.2.4 `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 >](#).

4.13.2.5 `template<class Typ> void StosTab< Typ>::_Zamien (unsigned int i, unsigned int j) [inline],[virtual]`

Metoda zamieniajaca Metoda ma za zadanie zamienic miejscami elementy wybrane przez argumenty wywolania.

Parameters

in	<i>i</i>	- Adres elementu podlegajacy zamianie
in	<i>j</i>	- Adres elementu podlegajacy zamianie

Implements [Iterable< Typ >](#).

4.13.2.6 `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 >](#).

4.13.2.7 `template<class Typ> Typ& StosTab< Typ >::Adres (unsigned int i) [inline],[virtual]`

Metoda zwracajaca referencje.

Metoda ma za zadanie zwrocic referencje do pola kontenera zadanego poprzez argument metody

Implements [Iterable< Typ >](#).

4.13.2.8 `template<class Typ> const Typ StosTab< Typ >::Wartosc (unsigned int i) const [inline],[virtual]`

Metoda zwracajaca wartosc.

Metoda ma za zadaniei zwrocic wartosc,kryjaca sie pod danym indeksem dla dowolnego kontenera posiadajacego interfejs.

Parameters

<code>in</code>	<code>i</code>	-Indeks z ktorego zostanie odczytana wartosc
-----------------	----------------	--

Returns

Zwraca wartosc, kryjaca sie pod danym indeksem konteneru

Implements [Iterable< Typ >](#).

4.13.3 Member Data Documentation

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

Pole klasy [StosTab](#).

Pole zawiera wskaznik na typ calkowity, sluzzy do alokacji pamieci na dynamiczna tablice

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

Pole Klasy [StosTab](#).

Pole przechowuje informacje o ilosci obecnie znajdujacych sie elementow na liscie danych

4.13.3.3 `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:

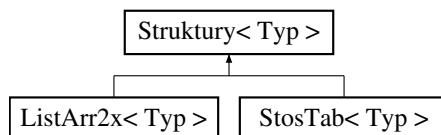
- `/home/bartolomeo/209296/prj/inc/StosTab.hh`

4.14 Struktury< Typ > Class Template Reference

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

```
#include <IStruktury.hh>
```

Inheritance diagram for Struktury< Typ >:



Public Member Functions

- virtual void [_Push](#) (Typ k, unsigned int Pozycja)=0
Metoda dodajaca kolejny element struktury.
- virtual Typ [_Pop](#) (unsigned int Pozycja)=0
Metoda usuwajaca element.
- virtual unsigned int [_Rozmiar](#) () const =0
Metoda zwracajaca rozmiar [Struktury](#).
- virtual void [_Pokaz](#) () const =0
Metoda wyswietlajaca dane.
- virtual void [_Zwolnij](#) ()=0
Metoda zwalnijaca pamiec.

4.14.1 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.

4.14.2 Member Function Documentation

4.14.2.1 `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 [ListArr2x< Typ >](#), and [StosTab< Typ >](#).

4.14.2.2 `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

in	<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 [ListArr2x< Typ >](#), and [StosTab< Typ >](#).

4.14.2.3 `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>	<code>k</code>	- wartosc typu calkowitnego, ktora bedzie umieszczona w strukturze
-----------------	----------------	--

Implemented in [StosTab< Typ >](#), and [ListArr2x< Typ >](#).

4.14.2.4 `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 [ListArr2x< Typ >](#), and [StosTab< Typ >](#).

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

Metoda zwalnijajaca pamiec.

Metoda ma za zadanie zwolnic pamiec uzywana przy zapelnienianiu danej struktury danymi

Implemented in [ListArr2x< Typ >](#), and [StosTab< Typ >](#).

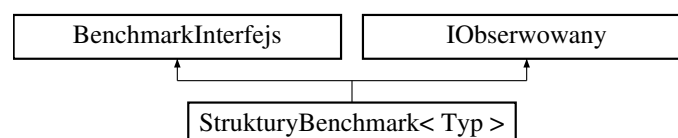
The documentation for this class was generated from the following file:

- </home/bartolomeo/209296/prj/inc/IStruktury.hh>

4.15 StrukturyBenchmark< Typ > Class Template Reference

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

Inheritance diagram for StrukturyBenchmark< Typ >:



Public Member Functions

- [StrukturyBenchmark](#) (const unsigned int Proby, const unsigned int Powt, unsigned int *Rozmiary)
Konstruktor obiektu.
- void [_WykonajTest](#) ()
Metoda inicjalizująca test.
- void [_Ustaw](#) (ISortable< Typ > *A, [Struktury](#)< Typ > *B, [Iterable](#)< Typ > *C)
Metoda Ustawiająca.
- void [_Wczytaj](#) (string PlikWart)
Metoda Wczytująca dane.
- void [_DodajObserwator](#) (IObserwator *O)
Metoda dodająca obserwator.
- void [_UsunObserwator](#) (IObserwator *O)
Metoda usuwająca obserwator.
- void [_Generator](#) () const
Metoda generująca dane.

Private Member Functions

- void [_Test](#) () const
Metoda wykonująca test dla odpowiedniej struktury.
- void [_Zaladuj](#) (const unsigned int n) const
Metoda wypełniająca Metoda ma za zadanie wypełnić dany kontener danymi.
- void [_Zwolnij](#) ()
Metoda zwalniania pamięci.
- void [_PowiadomObserwatorow](#) ()
Metoda informująca obserwatorów.

Private Attributes

- [ISortable](#)< Typ > * [I](#)
Pole StrukturyBenchmark Pole zawiera wskaźnik na interfejs sortujący, za pomocą niego i metod wirtualnych będą wywoływane odpowiednie dla danej struktury metody.
- [Struktury](#)< Typ > * [S](#)
Pole StrukturyBenchmark Pole zawiera wskaźnik na [Struktury](#), za pomocą niego i metod wirtualnych będą wywoływane odpowiednie dla danej struktury metody.
- [Iterable](#)< Typ > * [T](#)
Pole StrukturyBenchmark Pole zawiera wskaźnik na [Iterable](#), za pomocą niego i metod wirtualnych będą wywoływane odpowiednie dla danej struktury metody.
- int * [_Wartosci](#)
Pole StrukturyBenchmark Pole zawiera wskaźnik na typ całkowity, służy on do alokowania pamięci dla wczytanych z pliku danych.
- std::list< [IObserwator](#) * > [Obserwatorzy](#)
Pole [StrukturyBenchmark](#).
- unsigned int [_IloscProb](#)
Pole [StrukturyBenchmark](#).
- unsigned int [_IloscPowt](#)
Pole [StrukturyBenchmark](#).
- unsigned int * [_TablicaRozmiarow](#)
Pole [StrukturyBenchmark](#).
- unsigned int [_IloscDanych](#)
Pole [StrukturyBenchmark](#).

Additional Inherited Members

4.15.1 Detailed Description

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

Klasa modeluje pojecie Benchmarku przeznaczonego dla struktur danych przechowujace dane

4.15.2 Constructor & Destructor Documentation

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

Konstruktor obiektu.

4.15.3 Member Function Documentation

4.15.3.1 `template<class Typ > void StrukturyBenchmark< Typ >::_DodajObserwator (IObszerwator * 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 [IObserwowany](#).

4.15.3.2 `template<class Typ > void StrukturyBenchmark< Typ >::_Generator () const [inline],[virtual]`

Metoda generujaca dane.

Metoda ma za zadanie wygenerowac pseudolosowe dane i zapisac je do pliku

Implements [BenchmarkInterfejs](#).

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

Metoda informujaca obserwatorow.

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

Implements [IObserwowany](#).

4.15.3.4 `template<class Typ > void StrukturyBenchmark< Typ >::_Test () const [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 zapelniona struktura
-----------	----------	---

Implements [BenchmarkInterfejs](#).

4.15.3.5 `template<class Typ > void StrukturyBenchmark< Typ >::_Ustaw (ISortable< Typ > * A, Struktury< Typ > * B, Iterable< Typ > * C) [inline]`

Metoda Ustawiajaca.

Metoda ma za zadanie okreslic na jakich obiektach zostanie wykonana praca poprzez przypisanie do wskaznikow abstrakcyjnych interfejsow obiektow, ktore posiadaja dany interfejs

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

Metoda usuwajaca obserwator.

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

Parameters

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

Implements [IObszerwowany](#).

4.15.3.7 `template<class Typ > void StrukturyBenchmark< Typ >::_Wczytaj (string PlikWart) [inline],[virtual]`

Metoda Wczytujaca dane.

Metoda ma za zadanie wczytac dane wejciowe o podanej przez argument nazwie oraz przypisac wskaznik do zaalokowanych w pamieci danych

Parameters

<i>in</i>	<i>PlikIn</i>	- nazwa pliku wejsciowego z danymi
-----------	---------------	------------------------------------

Implements [BenchmarkInterfejs](#).

4.15.3.8 `template<class Typ > void StrukturyBenchmark< Typ >::_WykonajTest () [inline]`

Metoda inicjalizujaca test.

Metoda ma za zadanie uruchomic okreslona ilosc razy testowana metode, czas jej wykonania jest zbierany przez klase zewnetrzna

4.15.3.9 `template<class Typ > void StrukturyBenchmark< Typ >::_Zaladuj (const unsigned int n) const [inline],[private],[virtual]`

Metoda wypelniajaca Metoda ma za zadanie wypelnic dany kontener danymi.

Parameters

<i>in</i>	<i>n</i>	- ilosc danych
-----------	----------	----------------

Implements [BenchmarkInterfejs](#).

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

Metoda zwalnijająca Pamięć.

Metoda ma za zadanie zwolnić pamięć przeznaczoną na dane przechowywane w kontenerze

Implements [BenchmarkInterfejs](#).

4.15.4 Member Data Documentation

4.15.4.1 `template<class Typ > unsigned int StrukturyBenchmark< Typ >::_IloscDanych [private]`

Pole [StrukturyBenchmark](#).

Pole przechowuje informacje o ilości testowanych danych

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

Pole [StrukturyBenchmark](#).

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

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

Pole [StrukturyBenchmark](#).

Pole zawiera informacje o ilości prób jakie zostaną wykonane

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

Pole [StrukturyBenchmark](#).

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

4.15.4.5 `template<class Typ > int* StrukturyBenchmark< Typ >::_Wartosci [private]`

Pole StrukturyBenchmark Pole zawiera wskaźnik na typ całkowity, służy on do alokowania pamięci dla wczytanych z pliku danych.

4.15.4.6 `template<class Typ > ISortable<Typ>* StrukturyBenchmark< Typ >::_I [private]`

Pole StrukturyBenchmark Pole zawiera wskaźnik na interfejs sortujący, za pomocą niego i metod wirtualnych będą wywoływane odpowiednie dla danej struktury metody.

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

Pole [StrukturyBenchmark](#).

Pole zawiera listę obserwatorów, które obserwują ten obiekt

4.15.4.8 `template<class Typ > Struktury<Typ>* StrukturyBenchmark< Typ >::_S [private]`

Pole StrukturyBenchmark Pole zawiera wskaźnik na [Struktury](#), za pomocą niego i metod wirtualnych będą wywoływane odpowiednie dla danej struktury metody.

4.15.4.9 `template<class Typ> Iterable<Typ>* StrukturyBenchmark< Typ >::T [private]`

Pole StrukturyBenchmark Pole zawiera wskaźnik na [Iterable](#), za pomocą niego i metod wirtualnych będą wywołane odpowiednie dla danej struktury metody.

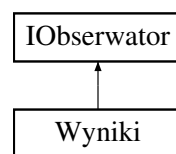
The documentation for this class was generated from the following file:

- </home/bartolomeo/209296/prj/inc/StrukturyBenchmark.hh>

4.16 Wyniki Class Reference

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

Inheritance diagram for Wyniki:



Public Member Functions

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

Metoda Aktualizująca stan.

Private Attributes

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

4.16.1 Constructor & Destructor Documentation

4.16.1.1 `Wyniki::Wyniki (const unsigned int Powtorzen, const unsigned int Proby, unsigned int * Rozmiary)`

4.16.1.2 `Wyniki::~~Wyniki ()`

4.16.2 Member Function Documentation

4.16.2.1 `void Wyniki::_Aktualizuj () [virtual]`

Metoda Aktualizująca stan.

Metoda ma za zadanie poinformować o zmianach w obiekcie który jest obserwowany

Implements [IObserwator](#).

4.16.2.2 void Wyniki::_ZapiszWyniki (std::string *PlikWy*) const

4.16.3 Member Data Documentation

4.16.3.1 unsigned int Wyniki::_IloscPowtorzen [private]

4.16.3.2 unsigned int Wyniki::_IloscProb [private]

4.16.3.3 unsigned int* Wyniki::_TablicaRozmiarow [private]

4.16.3.4 long double* Wyniki::_TablicaWynikow [private]

4.16.3.5 Czasomierz Wyniki::Stoper [private]

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

- /home/bartolomeo/209296/prj/inc/[Wyniki.hh](#)
- /home/bartolomeo/209296/prj/src/[Wyniki.cpp](#)

Chapter 5

File Documentation

5.1 /home/bartolomeo/209296/prj/inc/BenchmarkInterfejs.hh File Reference

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

Classes

- class [BenchmarkInterfejs](#)
Modeluje pojecie Interfejsu Benchmark'u.

Macros

- `#define` [DLUGOSC_SLOWA](#) 5

5.1.1 Macro Definition Documentation

5.1.1.1 `#define` [DLUGOSC_SLOWA](#) 5

5.2 /home/bartolomeo/209296/prj/inc/Czasomierz.hh File Reference

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

Classes

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

5.3 /home/bartolomeo/209296/prj/inc/HSort.hh File Reference

```
#include "ISortable.hh"
#include "Iterable.hh"
```

Classes

- class [HSort< Typ >](#)
Modeluje sortowanie przez kopcowanie.

5.4 /home/bartolomeo/209296/prj/inc/HybSort.hh File Reference

```
#include "QSortOpt.hh"
#include "ISortable.hh"
#include "Iterable.hh"
```

Classes

- class [HybSort< Typ >](#)
Modeluje sortowania hybrydowego.

Macros

- `#define` [PROG](#) 13
- `#define` [ILE](#) 3

5.4.1 Macro Definition Documentation

5.4.1.1 `#define` ILE 3

5.4.1.2 `#define` PROG 13

5.5 /home/bartolomeo/209296/prj/inc/IObserwator.hh File Reference

Classes

- class [IObserwator](#)
Modeluje pojecie interfejsu dla obserwatora.

5.6 /home/bartolomeo/209296/prj/inc/IObserwowany.hh File Reference

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

Classes

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

5.7 /home/bartolomeo/209296/prj/inc/ISortable.hh File Reference

```
#include "Iterable.hh"
```

Classes

- class [ISortable< Typ >](#)
Definicja klasy [ISortable](#).

5.8 /home/bartolomeo/209296/prj/inc/IStruktury.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.

5.9 /home/bartolomeo/209296/prj/inc/Iterable.hh File Reference

Classes

- class [Iterable< Typ >](#)
Interfejs [Iterable](#).

5.10 /home/bartolomeo/209296/prj/inc/ListArr2x.hh File Reference

Definicja klasy ListArr1.

```
#include "IStruktury.hh"
#include "Iterable.hh"
#include <fstream>
#include <cstdlib>
#include <cmath>
```

Classes

- class [ListArr2x< Typ >](#)
Modeluje pojęcie Listy (array)

5.10.1 Detailed Description

Definicja klasy ListArr1. Plik zawiera definicję klasy ListaArr2x ujętej w szablon typu wraz z jej składowymi metodami.

5.11 /home/bartolomeo/209296/prj/inc/MSort.hh File Reference

```
#include "ISortable.hh"
#include "Iterable.hh"
```

Classes

- class [MSort< Typ >](#)
Modeluje sortowanie przez scalanie.

5.12 /home/bartolomeo/209296/prj/inc/QSort.hh File Reference

```
#include "ISortable.hh"
#include "Iterable.hh"
```

Classes

- class [QSort< Typ >](#)
Modeluje sortowanie szybkie.

5.13 /home/bartolomeo/209296/prj/inc/QSortOpt.hh File Reference

```
#include "ISortable.hh"
#include "Iterable.hh"
```

Classes

- class [QSortOpt< Typ >](#)
Modeluje sortowanie szybkie z optymalizacją.

Macros

- `#define` [ILE](#) 3

5.13.1 Macro Definition Documentation

5.13.1.1 #define ILE 3

5.14 /home/bartolomeo/209296/prj/inc/StosTab.hh File Reference

```
#include <cstdlib>
#include <iostream>
#include "IStruktury.hh"
#include "Iterable.hh"
```

Classes

- class [StosTab< Typ >](#)

5.15 /home/bartolomeo/209296/prj/inc/StrukturyBenchmark.hh File Reference

```
#include "BenchmarkInterfejs.hh"
#include "ISortable.hh"
#include "IObserwowany.hh"
#include "IStruktury.hh"
```

Classes

- class [StrukturyBenchmark< Typ >](#)

5.16 /home/bartolomeo/209296/prj/inc/Wyniki.hh File Reference

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

Classes

- class [Wyniki](#)

5.17 /home/bartolomeo/209296/prj/src/Czasomierz.cpp File Reference

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

5.18 /home/bartolomeo/209296/prj/src/Main.cpp File Reference

funkcja glowna programu

```
#include "StrukturyBenchmark.hh"
#include "Wyniki.hh"
#include "HSort.hh"
#include "HybSort.hh"
#include "ListArr2x.hh"
#include "MSort.hh"
#include "QSort.hh"
#include "QSortOpt.hh"
#include "StosTab.hh"
```

Macros

- `#define ILOSC_POW 10`
- `#define ILOSC_PROB 5`

Functions

- `int main ()`

Variables

- `unsigned int Tablica_Rozmiarow [] = {100,1000,10000,100000,1000000}`

5.18.1 Detailed Description

funkcja glowna programu

5.18.2 Macro Definition Documentation

5.18.2.1 `#define ILOSC_POW 10`

5.18.2.2 `#define ILOSC_PROB 5`

5.18.3 Function Documentation

5.18.3.1 `int main ()`

5.18.4 Variable Documentation

5.18.4.1 `unsigned int Tablica_Rozmiarow[] = {100,1000,10000,100000,1000000}`

5.19 /home/bartolomeo/209296/prj/src/Wyniki.cpp File Reference

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

Index

- ~StosTab
 - StosTab, [29](#)
- ~Wyniki
 - Wyniki, [38](#)
- /home/bartolomeo/209296/prj/inc/BenchmarkInterfejs.-hh, [41](#)
- /home/bartolomeo/209296/prj/inc/Czasomierz.hh, [41](#)
- /home/bartolomeo/209296/prj/inc/HSort.hh, [42](#)
- /home/bartolomeo/209296/prj/inc/HybSort.hh, [42](#)
- /home/bartolomeo/209296/prj/inc/IObserwator.hh, [42](#)
- /home/bartolomeo/209296/prj/inc/IObserwowany.hh, [42](#)
- /home/bartolomeo/209296/prj/inc/ISortable.hh, [43](#)
- /home/bartolomeo/209296/prj/inc/IStruktury.hh, [43](#)
- /home/bartolomeo/209296/prj/inc/Iterable.hh, [43](#)
- /home/bartolomeo/209296/prj/inc/ListArr2x.hh, [43](#)
- /home/bartolomeo/209296/prj/inc/MSort.hh, [44](#)
- /home/bartolomeo/209296/prj/inc/QSort.hh, [44](#)
- /home/bartolomeo/209296/prj/inc/QSortOpt.hh, [44](#)
- /home/bartolomeo/209296/prj/inc/StosTab.hh, [45](#)
- /home/bartolomeo/209296/prj/inc/StrukturyBenchmark.-hh, [45](#)
- /home/bartolomeo/209296/prj/inc/Wyniki.hh, [45](#)
- /home/bartolomeo/209296/prj/src/Czasomierz.cpp, [45](#)
- /home/bartolomeo/209296/prj/src/Main.cpp, [46](#)
- /home/bartolomeo/209296/prj/src/Wyniki.cpp, [46](#)
- _Aktualizuj
 - IObserwator, [15](#)
 - Wyniki, [38](#)
- _AktualizujCzas
 - Czasomierz, [9](#)
- _Aktualny
 - Czasomierz, [10](#)
- _CzasTrwania
 - Czasomierz, [9](#)
- _DodajObserwator
 - IObserwowany, [15](#)
 - StrukturyBenchmark, [35](#)
- _Generator
 - BenchmarkInterfejs, [7](#)
 - StrukturyBenchmark, [35](#)
- _IloscDanych
 - StrukturyBenchmark, [37](#)
- _IloscPowt
 - StrukturyBenchmark, [37](#)
- _IloscPowtorzen
 - Wyniki, [39](#)
- _IloscProb
 - StrukturyBenchmark, [37](#)
 - Wyniki, [39](#)
- _Koniec
 - Czasomierz, [10](#)
- _L
 - StosTab, [31](#)
- _PojedynczyPomiar
 - Czasomierz, [9](#)
- _Pokaz
 - ListArr2x, [20](#)
 - StosTab, [29](#)
 - Struktury, [32](#)
- _Pop
 - ListArr2x, [20](#)
 - StosTab, [29](#)
 - Struktury, [32](#)
- _PowiadomObserwatorow
 - IObserwowany, [16](#)
 - StrukturyBenchmark, [35](#)
- _Push
 - ListArr2x, [20](#)
 - StosTab, [30](#)
 - Struktury, [33](#)
- _Reset
 - Czasomierz, [10](#)
- _Rozmiar
 - Iterable, [18](#)
 - ListArr2x, [20](#)
 - StosTab, [30](#)
 - Struktury, [33](#)
- _RozmiarL
 - StosTab, [31](#)
- _RozmiarT
 - StosTab, [31](#)
- _RozpocznijPomiar
 - Czasomierz, [10](#)
- _Sort
 - HSort, [11](#)
 - HybSort, [13](#)
 - ISortable, [17](#)
 - MSort, [23](#)
 - QSort, [26](#)
 - QSortOpt, [27](#)
- _Start
 - Czasomierz, [10](#)
- _Status
 - Czasomierz, [10](#)
- _StatusPracy
 - Czasomierz, [10](#)
- _TablicaRozmiarow
 - StrukturyBenchmark, [37](#)

- Wyniki, [39](#)
- _TablicaWynikow
 - Wyniki, [39](#)
- _Test
 - BenchmarkInterfejs, [8](#)
 - StrukturyBenchmark, [35](#)
- _Ustaw
 - StrukturyBenchmark, [36](#)
- _UsunObserwator
 - IObserwowany, [16](#)
 - StrukturyBenchmark, [36](#)
- _Wartosci
 - StrukturyBenchmark, [37](#)
- _Wczytaj
 - BenchmarkInterfejs, [8](#)
 - StrukturyBenchmark, [36](#)
- _WykonajTest
 - StrukturyBenchmark, [36](#)
- _ZakoncZPomiar
 - Czasomierz, [10](#)
- _Zaladuj
 - BenchmarkInterfejs, [8](#)
 - StrukturyBenchmark, [36](#)
- _Zamien
 - Iterable, [18](#)
 - ListArr2x, [20](#)
 - StosTab, [30](#)
- _ZapiszWyniki
 - Wyniki, [38](#)
- _Zwolnij
 - BenchmarkInterfejs, [8](#)
 - ListArr2x, [22](#)
 - StosTab, [30](#)
 - Struktury, [33](#)
 - StrukturyBenchmark, [36](#)
- Adres
 - Iterable, [18](#)
 - ListArr2x, [22](#)
 - StosTab, [31](#)
- BenchmarkInterfejs, [7](#)
 - _Generator, [7](#)
 - _Test, [8](#)
 - _Wczytaj, [8](#)
 - _Zaladuj, [8](#)
 - _Zwolnij, [8](#)
- BenchmarkInterfejs.hh
 - DLUGOSC_SLOWA, [41](#)
- BudujKopiec
 - HSort, [12](#)
- Czasomierz, [8](#)
 - _AktualizujCzas, [9](#)
 - _Aktualny, [10](#)
 - _CzasTrwania, [9](#)
 - _Koniec, [10](#)
 - _PojedynczyPomiar, [9](#)
 - _Reset, [10](#)
- _RozpocznijPomiar, [10](#)
 - _Start, [10](#)
 - _Status, [10](#)
 - _StatusPracy, [10](#)
 - _ZakoncZPomiar, [10](#)
 - Czasomierz, [9](#)
- DLUGOSC_SLOWA
 - BenchmarkInterfejs.hh, [41](#)
- HSort
 - _Sort, [11](#)
 - BudujKopiec, [12](#)
 - Kopiec, [12](#)
 - SortowanieKopiec, [12](#)
- HSort< Typ >, [11](#)
- HybSort
 - _Sort, [13](#)
 - Mediana, [13](#)
 - Partycjowanie, [13](#)
 - Sortowanie_Hybrydowe, [14](#)
 - Wstaw_Sort, [14](#)
- HybSort< Typ >, [12](#)
- HybSort.hh
 - ILE, [42](#)
 - PROG, [42](#)
- I
 - StrukturyBenchmark, [37](#)
- ILE
 - HybSort.hh, [42](#)
 - QSortOpt.hh, [45](#)
- ILOSC_POW
 - Main.cpp, [46](#)
- ILOSC_PROB
 - Main.cpp, [46](#)
- IObservator, [14](#)
 - _Aktualizuj, [15](#)
- IObserwowany, [15](#)
 - _DodajObserwator, [15](#)
 - _PowiadomObserwatorow, [16](#)
 - _UsunObserwator, [16](#)
- ISortable
 - _Sort, [17](#)
- ISortable< Typ >, [16](#)
- Iterable
 - _Rozmiar, [18](#)
 - _Zamien, [18](#)
 - Adres, [18](#)
 - Wartosc, [18](#)
- Iterable< Typ >, [17](#)
- Kopiec
 - HSort, [12](#)
- ListArr2x
 - _Pokaz, [20](#)
 - _Pop, [20](#)
 - _Push, [20](#)

- [_Rozmiar](#), [20](#)
 - [_Zamien](#), [20](#)
 - [_Zwolnij](#), [22](#)
 - [Adres](#), [22](#)
 - [ListArr2x](#), [20](#)
 - [ListArr2x](#), [20](#)
 - [RozmiarL](#), [22](#)
 - [RozmiarT](#), [22](#)
 - [tab](#), [23](#)
 - [Wartosc](#), [22](#)
 - [WczytajDane](#), [22](#)
- [ListArr2x< Typ >](#), [18](#)
- [MSort](#)
 - [_Sort](#), [23](#)
 - [Merge](#), [25](#)
 - [Msort](#), [25](#)
- [MSort< Typ >](#), [23](#)
- [main](#)
 - [Main.cpp](#), [46](#)
- [Main.cpp](#)
 - [ILOSC_POW](#), [46](#)
 - [ILOSC_PROB](#), [46](#)
 - [main](#), [46](#)
 - [Tablica_Rozmiarow](#), [46](#)
- [Mediana](#)
 - [HybSort](#), [13](#)
 - [QSortOpt](#), [28](#)
- [Merge](#)
 - [MSort](#), [25](#)
- [Msort](#)
 - [MSort](#), [25](#)
- [Obserwatorzy](#)
 - [StrukturyBenchmark](#), [37](#)
- [PROG](#)
 - [HybSort.hh](#), [42](#)
- [Partycjowanie](#)
 - [HybSort](#), [13](#)
 - [QSort](#), [26](#)
 - [QSortOpt](#), [28](#)
- [QSort](#)
 - [_Sort](#), [26](#)
 - [Partycjowanie](#), [26](#)
 - [Qsort](#), [26](#)
- [QSort< Typ >](#), [25](#)
- [QSortOpt](#)
 - [_Sort](#), [27](#)
 - [Mediana](#), [28](#)
 - [Partycjowanie](#), [28](#)
 - [QsortOpt](#), [28](#)
- [QSortOpt< Typ >](#), [27](#)
- [QSortOpt.hh](#)
 - [ILE](#), [45](#)
- [Qsort](#)
 - [QSort](#), [26](#)
- [QsortOpt](#)
 - [QSortOpt](#), [28](#)
- [RozmiarL](#)
 - [ListArr2x](#), [22](#)
- [RozmiarT](#)
 - [ListArr2x](#), [22](#)
- [S](#)
 - [StrukturyBenchmark](#), [37](#)
- [Sortowanie_Hybrydowe](#)
 - [HybSort](#), [14](#)
- [SortowanieKopiec](#)
 - [HSort](#), [12](#)
- [Stoper](#)
 - [Wyniki](#), [39](#)
- [StosTab](#)
 - [~StosTab](#), [29](#)
 - [_L](#), [31](#)
 - [_Pokaz](#), [29](#)
 - [_Pop](#), [29](#)
 - [_Push](#), [30](#)
 - [_Rozmiar](#), [30](#)
 - [_RozmiarL](#), [31](#)
 - [_RozmiarT](#), [31](#)
 - [_Zamien](#), [30](#)
 - [_Zwolnij](#), [30](#)
 - [Adres](#), [31](#)
 - [StosTab](#), [29](#)
 - [StosTab](#), [29](#)
 - [Wartosc](#), [31](#)
- [StosTab< Typ >](#), [28](#)
- [Struktury](#)
 - [_Pokaz](#), [32](#)
 - [_Pop](#), [32](#)
 - [_Push](#), [33](#)
 - [_Rozmiar](#), [33](#)
 - [_Zwolnij](#), [33](#)
- [Struktury< Typ >](#), [31](#)
- [StrukturyBenchmark](#)
 - [_DodajObserwator](#), [35](#)
 - [_Generator](#), [35](#)
 - [_IloscDanych](#), [37](#)
 - [_IloscPowt](#), [37](#)
 - [_IloscProb](#), [37](#)
 - [_PowiadomObserwatorow](#), [35](#)
 - [_TablicaRozmiarow](#), [37](#)
 - [_Test](#), [35](#)
 - [_Ustaw](#), [36](#)
 - [_UsunObserwator](#), [36](#)
 - [_Wartosci](#), [37](#)
 - [_Wczytaj](#), [36](#)
 - [_WykonajTest](#), [36](#)
 - [_Zaladuj](#), [36](#)
 - [_Zwolnij](#), [36](#)
- [I](#), [37](#)
- [Obserwatorzy](#), [37](#)
- [S](#), [37](#)
- [StrukturyBenchmark](#), [35](#)
- [StrukturyBenchmark](#), [35](#)

T, [37](#)
StrukturyBenchmark< Typ >, [33](#)

T

- StrukturyBenchmark, [37](#)

tab

- ListArr2x, [23](#)

Tablica_Rozmiarow

- Main.cpp, [46](#)

Wartosc

- Iterable, [18](#)
- ListArr2x, [22](#)
- StosTab, [31](#)

WczytajDane

- ListArr2x, [22](#)

Wstaw_Sort

- HybSort, [14](#)

Wyniki, [38](#)

- ~Wyniki, [38](#)
- _Aktualizuj, [38](#)
- _IloscPowtorzen, [39](#)
- _IloscProb, [39](#)
- _TablicaRozmiarow, [39](#)
- _TablicaWynikow, [39](#)
- _ZapiszWyniki, [38](#)
- Stoper, [39](#)
- Wyniki, [38](#)