

pamsi  
0.4

Generated by Doxygen 1.8.9.1

Sat Mar 19 2016 13:45:04



# Contents

<b>1</b>	<b>Hierarchical Index</b>	<b>1</b>
1.1	Class Hierarchy . . . . .	1
<b>2</b>	<b>Class Index</b>	<b>3</b>
2.1	Class List . . . . .	3
<b>3</b>	<b>File Index</b>	<b>5</b>
3.1	File List . . . . .	5
<b>4</b>	<b>Class Documentation</b>	<b>7</b>
4.1	IKolejka< T > Class Template Reference . . . . .	7
4.1.1	Detailed Description . . . . .	7
4.2	ILista< T > Class Template Reference . . . . .	7
4.2.1	Detailed Description . . . . .	7
4.2.2	Constructor & Destructor Documentation . . . . .	7
4.2.2.1	~ILista . . . . .	8
4.2.3	Member Function Documentation . . . . .	8
4.2.3.1	add . . . . .	8
4.2.3.2	get . . . . .	8
4.2.3.3	isEmpty . . . . .	8
4.2.3.4	remove . . . . .	8
4.2.3.5	size . . . . .	8
4.3	IStoper Class Reference . . . . .	8
4.3.1	Detailed Description . . . . .	8
4.3.2	Constructor & Destructor Documentation . . . . .	9
4.3.2.1	~IStoper . . . . .	9
4.3.3	Member Function Documentation . . . . .	9
4.3.3.1	getElapsedTimeMs . . . . .	9
4.3.3.2	start . . . . .	9
4.3.3.3	stop . . . . .	9
4.4	IStos< T > Class Template Reference . . . . .	10
4.4.1	Detailed Description . . . . .	10

4.5	Itabn< T > Class Template Reference . . . . .	10
4.5.1	Detailed Description . . . . .	11
4.5.2	Constructor & Destructor Documentation . . . . .	11
4.5.2.1	~Itabn . . . . .	11
4.5.3	Member Function Documentation . . . . .	11
4.5.3.1	add . . . . .	11
4.5.3.2	aSize . . . . .	12
4.5.3.3	nOE . . . . .	12
4.5.3.4	operator[] . . . . .	12
4.5.3.5	operator[] . . . . .	12
4.5.3.6	showElems . . . . .	12
4.6	Kolejka Class Reference . . . . .	12
4.6.1	Detailed Description . . . . .	13
4.7	Lista Class Reference . . . . .	13
4.7.1	Detailed Description . . . . .	13
4.8	Runnable Class Reference . . . . .	13
4.8.1	Detailed Description . . . . .	14
4.8.2	Constructor & Destructor Documentation . . . . .	14
4.8.2.1	~Runnable . . . . .	14
4.8.3	Member Function Documentation . . . . .	14
4.8.3.1	generateRandomDgt . . . . .	14
4.8.3.2	prepare . . . . .	14
4.8.3.3	run . . . . .	14
4.8.3.4	seedSrand . . . . .	15
4.9	Starter Class Reference . . . . .	15
4.9.1	Detailed Description . . . . .	15
4.9.2	Constructor & Destructor Documentation . . . . .	16
4.9.2.1	Starter . . . . .	16
4.9.2.2	~Starter . . . . .	16
4.9.3	Member Function Documentation . . . . .	16
4.9.3.1	dumpToFile . . . . .	16
4.9.3.2	printResults . . . . .	16
4.9.3.3	setTestSize . . . . .	17
4.9.3.4	test . . . . .	17
4.10	Stoper Class Reference . . . . .	18
4.10.1	Detailed Description . . . . .	19
4.10.2	Member Function Documentation . . . . .	19
4.10.2.1	getElapsedTimeMs . . . . .	20
4.10.2.2	start . . . . .	21
4.10.2.3	stop . . . . .	21

4.11	Stos Class Reference	21
4.11.1	Detailed Description	21
4.12	tabn< T > Class Template Reference	21
4.12.1	Detailed Description	23
4.12.2	Constructor & Destructor Documentation	23
4.12.2.1	tabn	23
4.12.2.2	~tabn	23
4.12.3	Member Function Documentation	23
4.12.3.1	add	23
4.12.3.2	aSize	23
4.12.3.3	nOE	23
4.12.3.4	operator[]	24
4.12.3.5	operator[]	24
4.12.3.6	showElems	24
4.13	tabn_test Class Reference	24
4.13.1	Detailed Description	25
4.13.2	Constructor & Destructor Documentation	25
4.13.2.1	tabn_test	26
4.13.2.2	~tabn_test	26
4.13.3	Member Function Documentation	26
4.13.3.1	prepare	26
4.13.3.2	run	26
<b>5</b>	<b>File Documentation</b>	<b>29</b>
5.1	kolejka.cpp File Reference	29
5.2	kolejka.hh File Reference	29
5.3	lista.cpp File Reference	30
5.4	lista.hh File Reference	30
5.5	main.cpp File Reference	30
5.5.1	Detailed Description	31
5.5.2	Function Documentation	31
5.5.2.1	main	31
5.6	main.hh File Reference	32
5.6.1	Detailed Description	32
5.7	run.cpp File Reference	32
5.8	run.hh File Reference	33
5.8.1	Detailed Description	33
5.9	starter.cpp File Reference	34
5.10	starter.hh File Reference	34
5.10.1	Detailed Description	35

---

5.11 stoper.cpp File Reference . . . . .	36
5.12 stoper.hh File Reference . . . . .	36
5.13 stos.cpp File Reference . . . . .	37
5.14 stos.hh File Reference . . . . .	38
5.15 tabl.cpp File Reference . . . . .	38
5.16 tabl.hh File Reference . . . . .	39
5.16.1 Detailed Description . . . . .	40
5.16.2 Macro Definition Documentation . . . . .	40
5.16.2.1 SIZE . . . . .	40
<b>Index</b>	<b>41</b>

# Chapter 1

## Hierarchical Index

### 1.1 Class Hierarchy

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

IKolejka< T > . . . . .	7
ILista< T > . . . . .	7
IStoper . . . . .	8
Stoper . . . . .	18
IStos< T > . . . . .	10
Itabn< T > . . . . .	10
tabn< T > . . . . .	21
Itabn< int > . . . . .	10
Kolejka . . . . .	12
Lista . . . . .	13
Runnable . . . . .	13
tabn_test . . . . .	24
Starter . . . . .	15
Stos . . . . .	21





## Chapter 2

# Class Index

### 2.1 Class List

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

<a href="#">IKolejka&lt; T &gt;</a>	7
<a href="#">ILista&lt; T &gt;</a>	7
<a href="#">IStoper</a>	
Interfejs <a href="#">IStoper</a>	8
<a href="#">IStos&lt; T &gt;</a>	10
<a href="#">Itabn&lt; T &gt;</a>	
Typ wyliczeniowy decydujący o sposobie rozszerzania tablicy dynamicznej	10
<a href="#">Kolejka</a>	12
<a href="#">Lista</a>	13
<a href="#">Runnable</a>	
Klasa ujednolica sposób uruchamiania klasy badającej algorytm	13
<a href="#">Starter</a>	
Klasa pozwala na przeprowadzenie testów	15
<a href="#">Stoper</a>	
Klasa stoper implementująca interfejs <a href="#">IStoper</a>	18
<a href="#">Stos</a>	21
<a href="#">tabn&lt; T &gt;</a>	
Modeluje tablicę dynamicznie rozszerzalną	21
<a href="#">tabn_test</a>	
Definiuje sposób testowania tablicy tabn	24



## Chapter 3

# File Index

### 3.1 File List

Here is a list of all files with brief descriptions:

<a href="#">kolejka.cpp</a>	29
<a href="#">kolejka.hh</a>	29
<a href="#">lista.cpp</a>	30
<a href="#">lista.hh</a>	30
<a href="#">main.cpp</a>	
Główny plik programu	30
<a href="#">main.hh</a>	
Plik posiada wspólne definicje	32
<a href="#">run.cpp</a>	32
<a href="#">run.hh</a>	
Plik definiuje klasę <a href="#">Runnable</a> , ujednolicającą klasy umożliwiające badanie algorytmów	33
<a href="#">starter.cpp</a>	34
<a href="#">starter.hh</a>	
Plik definiuje klasę <a href="#">Starter</a>	34
<a href="#">stoper.cpp</a>	36
<a href="#">stoper.hh</a>	36
<a href="#">stos.cpp</a>	37
<a href="#">stos.hh</a>	38
<a href="#">tabl.cpp</a>	38
<a href="#">tabl.hh</a>	
Definicja interfejsu klasy <code>tabn</code> , klasy <code>tabn</code> oraz klasy <a href="#">tabn_test</a>	39



## Chapter 4

# Class Documentation

### 4.1 IKolejka< T > Class Template Reference

```
#include <kolejka.hh>
```

#### 4.1.1 Detailed Description

```
template<class T>class IKolejka< T >
```

Definition at line 10 of file kolejka.hh.

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

- [kolejka.hh](#)

### 4.2 ILista< T > Class Template Reference

```
#include <lista.hh>
```

#### Public Member Functions

- virtual void [add](#) (T, unsigned int)=0
- virtual void [remove](#) (unsigned int)=0
- virtual bool [isEmpty](#) (void)=0
- virtual T [get](#) (unsigned int)=0
- virtual unsigned int [size](#) (void)=0
- virtual [~ILista](#) ()

#### 4.2.1 Detailed Description

```
template<class T>class ILista< T >
```

Definition at line 9 of file lista.hh.

#### 4.2.2 Constructor & Destructor Documentation

4.2.2.1 `template<class T> virtual ILista<T>::~~ILista ( ) [inline],[virtual]`

Definition at line 16 of file lista.hh.

### 4.2.3 Member Function Documentation

4.2.3.1 `template<class T> virtual void ILista<T>::add ( T, unsigned int ) [pure virtual]`

4.2.3.2 `template<class T> virtual T ILista<T>::get ( unsigned int ) [pure virtual]`

4.2.3.3 `template<class T> virtual bool ILista<T>::isEmpty ( void ) [pure virtual]`

4.2.3.4 `template<class T> virtual void ILista<T>::remove ( unsigned int ) [pure virtual]`

4.2.3.5 `template<class T> virtual unsigned int ILista<T>::size ( void ) [pure virtual]`

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

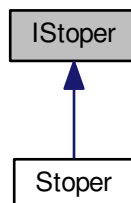
- [lista.hh](#)

## 4.3 IStoper Class Reference

Interfejs [IStoper](#).

```
#include <stoper.hh>
```

Inheritance diagram for IStoper:



### Public Member Functions

- virtual void [start](#) (void)=0
- virtual void [stop](#) (void)=0
- virtual long double [getElapsedTimeMs](#) (void)=0
- virtual [~IStoper](#) ()

#### 4.3.1 Detailed Description

Interfejs [IStoper](#).

Definition at line 21 of file stoper.hh.

### 4.3.2 Constructor & Destructor Documentation

#### 4.3.2.1 virtual IStoper::~IStoper ( ) [inline],[virtual]

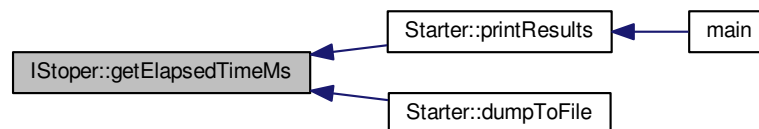
Definition at line 27 of file stoper.hh.

### 4.3.3 Member Function Documentation

#### 4.3.3.1 virtual long double IStoper::getElapsedTimeMs ( void ) [pure virtual]

Implemented in [Stoper](#).

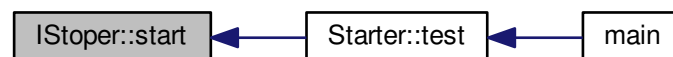
Here is the caller graph for this function:



#### 4.3.3.2 virtual void IStoper::start ( void ) [pure virtual]

Implemented in [Stoper](#).

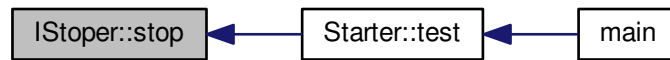
Here is the caller graph for this function:



#### 4.3.3.3 virtual void IStoper::stop ( void ) [pure virtual]

Implemented in [Stoper](#).

Here is the caller graph for this function:



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

- [stoper.hh](#)

## 4.4 Istos< T > Class Template Reference

```
#include <stos.hh>
```

### 4.4.1 Detailed Description

```
template<class T>class Istos< T >
```

Definition at line 10 of file stos.hh.

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

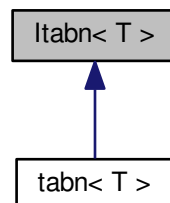
- [stos.hh](#)

## 4.5 Itabn< T > Class Template Reference

Typ wyliczeniowy decydujący o sposobie rozszerzania tablicy dynamicznej.

```
#include <tabl.hh>
```

Inheritance diagram for Itabn< T >:





## Public Member Functions

- virtual void [add](#) (T)=0
- virtual void [showElems](#) (void)=0
- virtual int [nOE](#) (void)=0
- virtual int [aSize](#) (void)=0
- virtual T & [operator\[\]](#) (int)=0
- virtual T [operator\[\]](#) (int) const =0
- virtual [~Itabn](#) ()

### 4.5.1 Detailed Description

`template<class T>class Itabn< T >`

Typ wyliczeniowy decydujący o sposobie rozszerzania tablicy dynamicznej.

Decyduje o typie rozszerzania tablicy

Zdefiniuj zmienną `expandType` w `main`, aby zmienić tryb rozszerzania tablicy w funkcji `tabn`.

Nie jest to idealne rozwiązanie, ale każda klasa implementująca [Runnable](#) może być inna. Musi także istnieć możliwość wyboru trybu rozszerzania tablicy. Stworzenie oddzielnych klas `tabn` dla każdego trybu oddzielnie znacznie zwiększa rozmiar kodu i komplikuje go. Obecne obejście:

- Nie powoduje konieczności zmiany [Runnable](#) zależnie od badanej funkcji
- Nie wymusza tworzenia trzech oddzielnych klas dla każdego trybu, następnie stworzenia trzech klas [tabn\\_↵test](#) itd.
- Wymaga jedynie od użytkownika użycia dodatkowej definicji.

Bardzo proszę o kontakt, jeśli można to rozwiązać lepiej.

Definition at line 51 of file `tabl.hh`.

### 4.5.2 Constructor & Destructor Documentation

4.5.2.1 `template<class T> virtual Itabn< T >::~~Itabn ( ) [inline],[virtual]`

Definition at line 61 of file `tabl.hh`.

### 4.5.3 Member Function Documentation

4.5.3.1 `template<class T> virtual void Itabn< T >::add ( T ) [pure virtual]`

Implemented in [tabn< T >](#).

Here is the caller graph for this function:



4.5.3.2 `template<class T> virtual int Itabn< T >::aSize ( void ) [pure virtual]`

Implemented in [tabn< T >](#).

Here is the caller graph for this function:



4.5.3.3 `template<class T> virtual int Itabn< T >::nOE ( void ) [pure virtual]`

Implemented in [tabn< T >](#).

Here is the caller graph for this function:



4.5.3.4 `template<class T> virtual T& Itabn< T >::operator[] ( int ) [pure virtual]`

Implemented in [tabn< T >](#).

4.5.3.5 `template<class T> virtual T Itabn< T >::operator[] ( int ) const [pure virtual]`

Implemented in [tabn< T >](#).

4.5.3.6 `template<class T> virtual void Itabn< T >::showElems ( void ) [pure virtual]`

Implemented in [tabn< T >](#).

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

- [tabl.hh](#)

## 4.6 Kolejka Class Reference

```
#include <kolejka.hh>
```

### 4.6.1 Detailed Description

Definition at line 18 of file kolejka.hh.

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

- [kolejka.hh](#)

## 4.7 Lista Class Reference

```
#include <lista.hh>
```

### 4.7.1 Detailed Description

Definition at line 19 of file lista.hh.

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

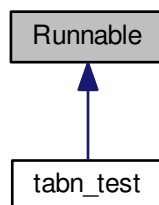
- [lista.hh](#)

## 4.8 Runnable Class Reference

Klasa ujednolica sposób uruchamiania klasy badającej algorytm.

```
#include <run.hh>
```

Inheritance diagram for Runnable:



### Public Member Functions

- virtual bool [prepare](#) (unsigned int)=0  
*Przygotowuje badania.*
- virtual bool [run](#) ()=0  
*Przeprowadza badania.*
- virtual [~Runnable](#) ()  
*Destruktor wirtualny klasy [Runnable](#).*
- void [seedSrand](#) (void)  
*Metoda ustawia punkt startowy generatora pseudolosowego.*
- int [generateRandomDgt](#) (void)  
*Metoda generuje liczbę pseudolosową z zakresu 0..9.*

### 4.8.1 Detailed Description

Klasa ujednolica sposób uruchamiania klasy badającej algorytm.

Definition at line 18 of file run.hh.

### 4.8.2 Constructor & Destructor Documentation

#### 4.8.2.1 virtual Runnable::~~Runnable ( ) [inline],[virtual]

Destruktor wirtualny klasy [Runnable](#).

Definition at line 37 of file run.hh.

### 4.8.3 Member Function Documentation

#### 4.8.3.1 int Runnable::generateRandomDgt ( void )

Metoda generuje liczbę pseudolosową z zakresu 0..9.

Return values

<i>Liczba</i>	pseudolosowa z zakresu 0..9
---------------	-----------------------------

Definition at line 7 of file run.cpp.

#### 4.8.3.2 virtual bool Runnable::prepare ( unsigned int ) [pure virtual]

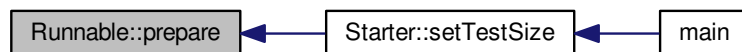
Przygotowuje badania.

Return values

<i>Zawsze</i>	true
---------------	------

Implemented in [tabn\\_test](#).

Here is the caller graph for this function:



#### 4.8.3.3 virtual bool Runnable::run ( ) [pure virtual]

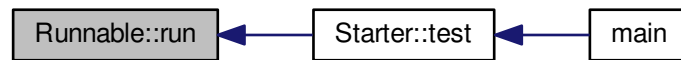
Przeprowadza badania.

Return values

<i>Zawsze</i>	true
---------------	------

Implemented in [tabn\\_test](#).

Here is the caller graph for this function:



#### 4.8.3.4 void Runnable::seedSrand ( void )

Metoda ustawia punkt startowy generatora pseudolosowego.

Definition at line 3 of file run.cpp.

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

- [run.hh](#)
- [run.cpp](#)

## 4.9 Starter Class Reference

Klasa pozwala na przeprowadzenie testów.

```
#include <starter.hh>
```

### Public Member Functions

- [Starter](#) ()  
*Konstruktor klasy tabn.*
- virtual [~Starter](#) ()  
*Destruktor klasy tabn.*
- void [setTestSize](#) (unsigned int)  
*Metoda ustawia wielkość testu.*
- void [printResults](#) (void)  
*Metoda wyświetla czas trwania testu na standardowym wyjściu.*
- void [test](#) (void)  
*Metoda przeprowadza test.*
- void [dumpToFile](#) (string)  
*Metoda dopisuje dane do pliku.*

#### 4.9.1 Detailed Description

Klasa pozwala na przeprowadzenie testów.

Definition at line 17 of file starter.hh.

## 4.9.2 Constructor & Destructor Documentation

### 4.9.2.1 Starter::Starter ( ) [inline]

Konstruktor klasy tabn.

Definition at line 27 of file starter.hh.

### 4.9.2.2 virtual Starter::~Starter ( ) [inline],[virtual]

Destruktor klasy tabn.

Definition at line 33 of file starter.hh.

## 4.9.3 Member Function Documentation

### 4.9.3.1 void Starter::dumpToFile ( string *nameOfFile* )

Metoda dopisuje dane do pliku.

Format zapisu: wielkość\_testu czas\_trwania\_ms

Parameters

in	-	nameOfFile - nazwa pliku wyjściowego
----	---	--------------------------------------

Definition at line 20 of file starter.cpp.

Here is the call graph for this function:



### 4.9.3.2 void Starter::printResults ( void )

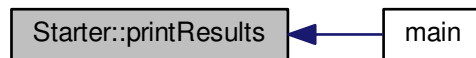
Metoda wyświetla czas trwania testu na standardowym wyjściu.

Definition at line 8 of file starter.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



#### 4.9.3.3 void Starter::setSize ( unsigned int *testsize* )

Metoda ustawia wielkość testu.

##### Parameters

<i>in</i>	<i>testsize</i>	- wielkość testu
-----------	-----------------	------------------

Definition at line 3 of file starter.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

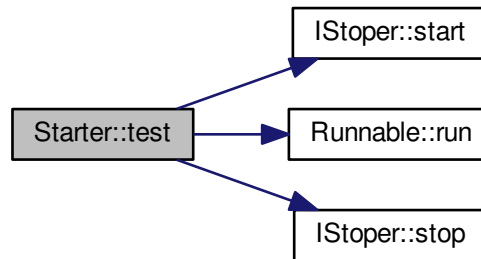


#### 4.9.3.4 void Starter::test ( void )

Metoda przeprowadza test.

Definition at line 14 of file starter.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



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

- [starter.hh](#)
- [starter.cpp](#)

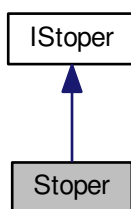
## 4.10 Stoper Class Reference

Klasa `stoper` implementująca interfejs `IStoper`.

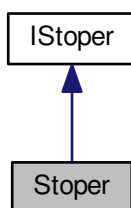
```
#include <stoper.hh>
```



Inheritance diagram for Stoper:



Collaboration diagram for Stoper:



## Public Member Functions

- virtual void `start` (void)  
*Uruchamia zegar.*
- virtual void `stop` (void)  
*Zatrzymuje zegar.*
- virtual long double `getElapsedTimeMs` (void)  
*Oblicza i zwraca czas pomiędzy uruchomieniem zegara a jego zatrzymaniem.*

### 4.10.1 Detailed Description

Klasa stoper implementująca interfejs `IStoper`.

Klasa symuluje działanie stopera - zapisuje początkowy i końcowy moment działania (użycie `start` i `stop`), oraz odejmuje obie te wartości od siebie, by uzyskać czas działania.

Definition at line 37 of file `stoper.hh`.

### 4.10.2 Member Function Documentation

4.10.2.1 `long double Stoper::getElapsedTimeMs ( void ) [virtual]`

Oblicza i zwraca czas pomiędzy uruchomieniem zegara a jego zatrzymaniem.

Return values

<i>Czas</i>	pomiędzy startem a zatrzymaniem zegara
-------------	--

Implements [IStoper](#).

Definition at line 12 of file stoper.cpp.

4.10.2.2 `void Stoper::start ( void ) [virtual]`

Uruchamia zegar.

Implements [IStoper](#).

Definition at line 4 of file stoper.cpp.

4.10.2.3 `void Stoper::stop ( void ) [virtual]`

Zatrzymuje zegar.

Implements [IStoper](#).

Definition at line 8 of file stoper.cpp.

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

- [stoper.hh](#)
- [stoper.cpp](#)

## 4.11 Stos Class Reference

```
#include <stos.hh>
```

### 4.11.1 Detailed Description

Definition at line 18 of file stos.hh.

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

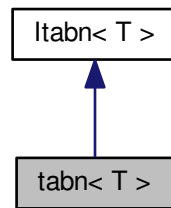
- [stos.hh](#)

## 4.12 `tabn< T >` Class Template Reference

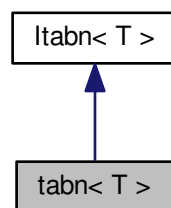
Modeluje tablicę dynamicznie rozszerzalną

```
#include <tabl.hh>
```

Inheritance diagram for `tabn< T >`:



Collaboration diagram for `tabn< T >`:



## Public Member Functions

- `tabn ()`  
*Konstruktor klasy tabn.*
- `virtual ~tabn ()`  
*Destruktor klasy tabn.*
- `virtual void add (T)`  
*Dodaje element w zadanym trybie Dodaje element do tablicy dynamicznej, odpowiednio ją rozszerzając.*
- `virtual void showElems (void)`  
*Wyświetla listę elementów.*
- `virtual int nOE (void)`  
*zwraca liczbę elementów w tablicy*
- `virtual int aSize (void)`  
*zwraca wielkość zaalokowanej przestrzeni dla tablicy*
- `virtual T & operator[] (int)`  
*Umożliwia dostęp do dowolnego elementu tablicy bez sprawdzania zakresu.*
- `virtual T operator[] (int) const`  
*Umożliwia odczyt dowolnego elementu tablicy bez sprawdzania zakresu.*

### 4.12.1 Detailed Description

`template<class T>class tabn< T >`

Modeluje tablicę dynamicznie rozszerzalną

Przechowuje elementy w rozszerzalnej tablicy o rozmiarze początkowym `SIZE`

Definition at line 71 of file `tabl.hh`.

### 4.12.2 Constructor & Destructor Documentation

4.12.2.1 `template<class T > tabn< T >::tabn ( )` `[inline]`

Konstruktor klasy `tabn`.

Definition at line 83 of file `tabl.hh`.

4.12.2.2 `template<class T > virtual tabn< T >::~~tabn ( )` `[inline]`, `[virtual]`

Destruktor klasy `tabn`.

Definition at line 93 of file `tabl.hh`.

### 4.12.3 Member Function Documentation

4.12.3.1 `template<class T > void tabn< T >::add ( T element )` `[virtual]`

Dodaje element w zadanym trybie Dodaje element do tablicy dynamicznej, odpowiednio ją rozszerzając.

Parameters

<i>element</i>	- element do dodania
----------------	----------------------

Implements [ltabn< T >](#).

Definition at line 188 of file `tabl.hh`.

4.12.3.2 `template<class T > int tabn< T >::aSize ( void )` `[virtual]`

zwraca wielkość zaalokowanej przestrzeni dla tablicy

Return values

<i>int</i>	Ilość zaalokowanych pól
------------	-------------------------

Implements [ltabn< T >](#).

Definition at line 279 of file `tabl.hh`.

4.12.3.3 `template<class T > int tabn< T >::nOE ( void )` `[virtual]`

zwraca liczbę elementów w tablicy

Return values

<i>int</i>	Liczba elementów w tablicy
------------	----------------------------

Implements [ltabn< T >](#).

Definition at line 274 of file tabl.hh.

#### 4.12.3.4 `template<class T> T & tabn< T >::operator[] ( int index ) [virtual]`

Umożliwia dostęp do dowolnego elementu tablicy bez sprawdzania zakresu.

##### Parameters

<i>index</i>	- numer elementu tablicy
--------------	--------------------------

##### Return values

<i>T*</i>	Wskaźnik na wybrany element tablicy
-----------	-------------------------------------

Implements [ltabn< T >](#).

Definition at line 256 of file tabl.hh.

#### 4.12.3.5 `template<class T> T tabn< T >::operator[] ( int index ) const [virtual]`

Umożliwia odczyt dowolnego elementu tablicy bez sprawdzania zakresu.

##### Parameters

<i>index</i>	- numer elementu tablicy
--------------	--------------------------

##### Return values

<i>T</i>	Element tablicy
----------	-----------------

Implements [ltabn< T >](#).

Definition at line 261 of file tabl.hh.

#### 4.12.3.6 `template<class T> void tabn< T >::showElems ( void ) [virtual]`

Wyświetla listę elementów.

Implements [ltabn< T >](#).

Definition at line 266 of file tabl.hh.

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

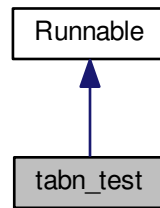
- [tabl.hh](#)

## 4.13 tabn\_test Class Reference

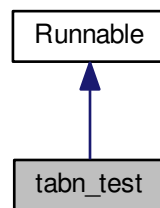
Definiuje sposób testowania tablicy tabn.

```
#include <tabl.hh>
```

Inheritance diagram for tabn\_test:



Collaboration diagram for tabn\_test:



## Public Member Functions

- [tabn\\_test](#) ()  
*Konstruktor klasy [tabn\\_test](#).*
- virtual [~tabn\\_test](#) ()  
*Destruktor klasy [tabn\\_test](#).*
- virtual bool [prepare](#) (unsigned int sizeOfTest)  
*Przygotowuje rozmiar testu.*
- virtual bool [run](#) ()  
*Wykonuje test.*

### 4.13.1 Detailed Description

Definiuje sposób testowania tablicy tabn.

Definition at line 288 of file tabl.hh.

### 4.13.2 Constructor & Destructor Documentation

#### 4.13.2.1 `tabn_test::tabn_test ( )` `[inline]`

Konstruktor klasy [tabn\\_test](#).

Definition at line 296 of file `tabl.hh`.

#### 4.13.2.2 `virtual tabn_test::~~tabn_test ( )` `[inline]`, `[virtual]`

Destruktor klasy [tabn\\_test](#).

Definition at line 302 of file `tabl.hh`.

### 4.13.3 Member Function Documentation

#### 4.13.3.1 `virtual bool tabn_test::prepare ( unsigned int sizeOfTest )` `[inline]`, `[virtual]`

Przygotowuje rozmiar testu.

Parameters

<i>sizeOfTest</i>	- rozmiar testu
-------------------	-----------------

Return values

<i>bool</i>	zawsze true
-------------	-------------

Implements [Runnable](#).

Definition at line 321 of file `tabl.hh`.

#### 4.13.3.2 `virtual bool tabn_test::run ( )` `[inline]`, `[virtual]`

Wykonuje test.

Pozwala na wykonanie testu w pętli `for` iterującej `counter` razy. Zasila funkcję dodawania generując losowe cyfry w funkcji [generateRandomDgt\(\)](#)

Return values

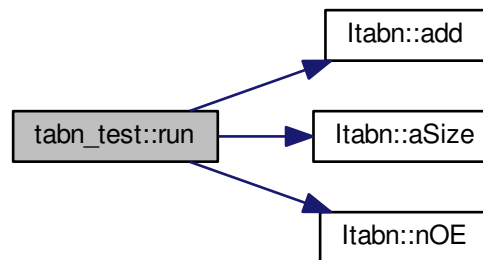
<i>bool</i>	zawsze true
-------------	-------------

Implements [Runnable](#).

Definition at line 337 of file `tabl.hh`.



Here is the call graph for this function:



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

- [tabl.hh](#)



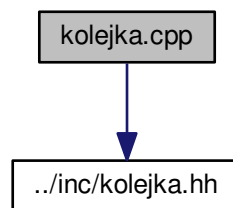
## Chapter 5

# File Documentation

### 5.1 kolejka.cpp File Reference

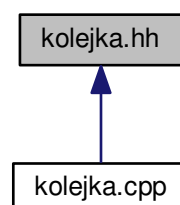
```
#include "../inc/kolejka.hh"
```

Include dependency graph for kolejka.cpp:



### 5.2 kolejka.hh File Reference

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



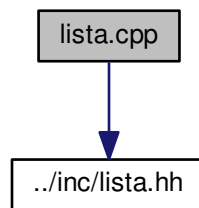
## Classes

- class [IKolejka< T >](#)
- class [Kolejka](#)

## 5.3 lista.cpp File Reference

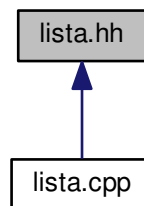
```
#include "../inc/lista.hh"
```

Include dependency graph for lista.cpp:



## 5.4 lista.hh File Reference

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



## Classes

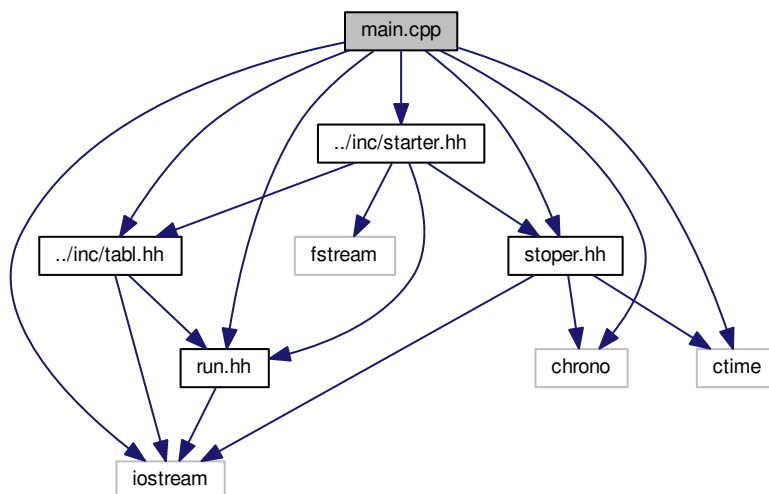
- class [ILista< T >](#)
- class [Lista](#)

## 5.5 main.cpp File Reference

Główny plik programu.

```
#include <iostream>
#include <chrono>
#include <ctime>
#include "../inc/tabl.hh"
#include "../inc/run.hh"
#include "../inc/starter.hh"
#include "../inc/stoper.hh"
```

Include dependency graph for main.cpp:



## Functions

- int [main](#) (void)

### 5.5.1 Detailed Description

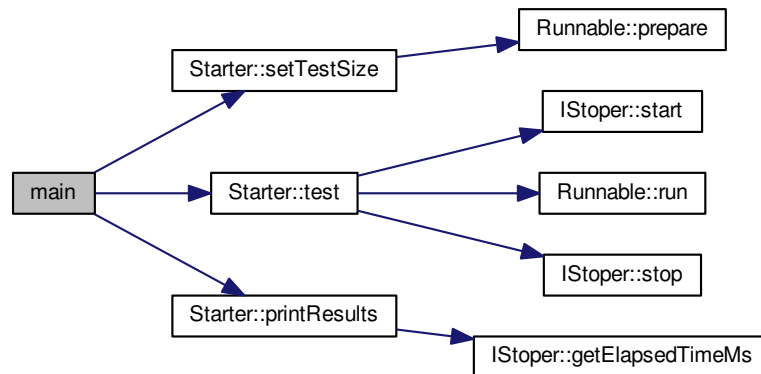
Główny plik programu.

### 5.5.2 Function Documentation

#### 5.5.2.1 int main ( void )

Definition at line 18 of file main.cpp.

Here is the call graph for this function:



## 5.6 main.hh File Reference

Plik posiada wspólne definicje.

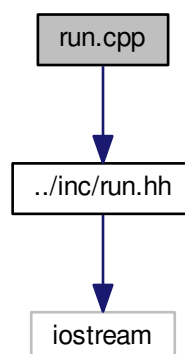
### 5.6.1 Detailed Description

Plik posiada wspólne definicje.

## 5.7 run.cpp File Reference

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

Include dependency graph for run.cpp:

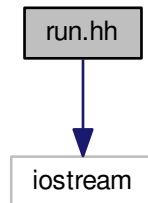


## 5.8 run.hh File Reference

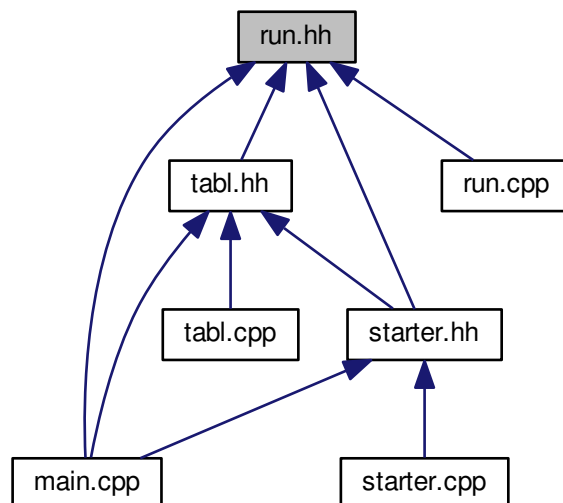
Plik definiuje klasę [Runnable](#), ujednolicającą klasy umożliwiające badanie algorytmów.

```
#include <iostream>
```

Include dependency graph for run.hh:



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



### Classes

- class [Runnable](#)

*Klasa ujednolici sposób uruchamiania klasy badającej algorytm.*

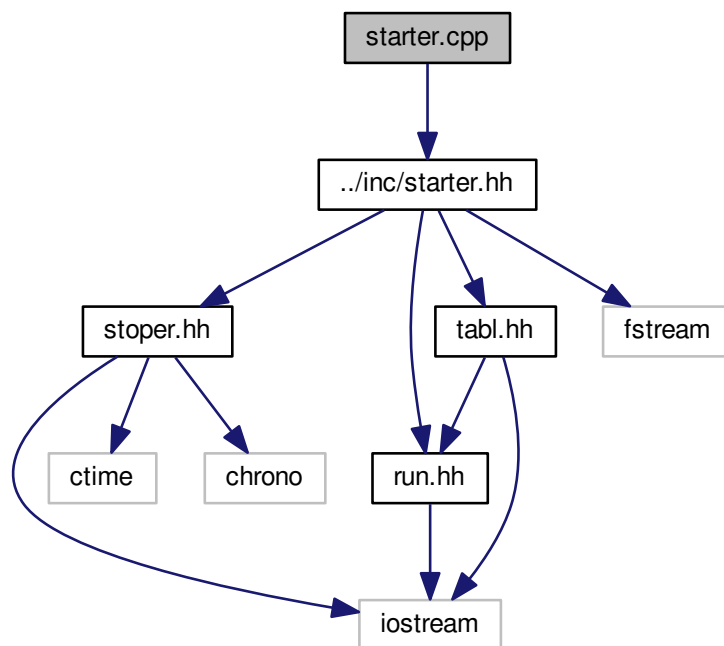
#### 5.8.1 Detailed Description

Plik definiuje klasę [Runnable](#), ujednolicającą klasy umożliwiające badanie algorytmów.

## 5.9 starter.cpp File Reference

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

Include dependency graph for starter.cpp:



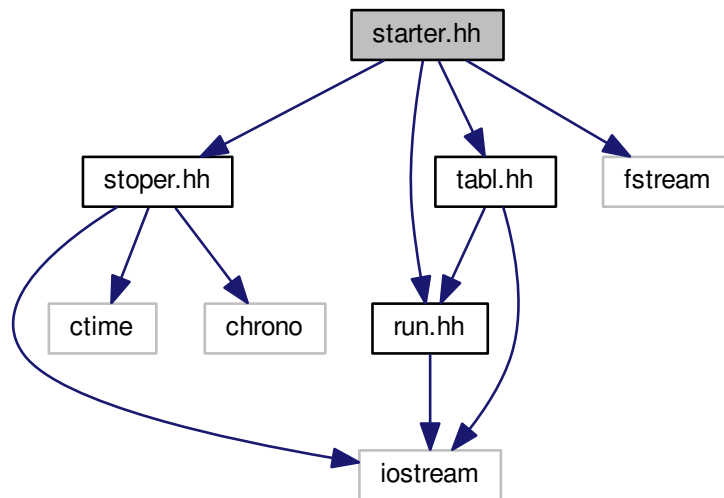
## 5.10 starter.hh File Reference

Plik definiuje klasę [Starter](#).

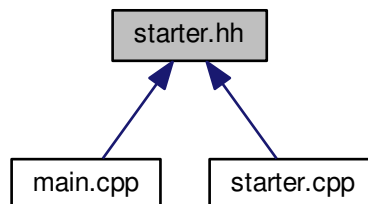
```
#include "stoper.hh"
#include "run.hh"
#include "tabl.hh"
#include <fstream>
```



Include dependency graph for starter.hh:



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



## Classes

- class [Starter](#)

*Klasa pozwala na przeprowadzenie testów.*

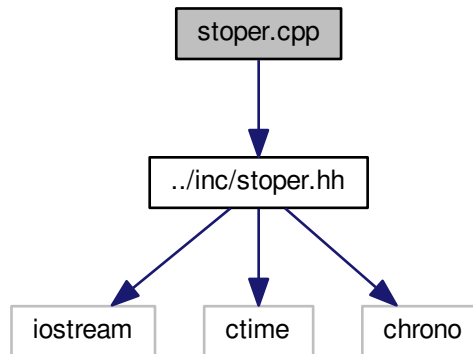
### 5.10.1 Detailed Description

Plik definiuje klasę [Starter](#).

## 5.11 stoper.cpp File Reference

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

Include dependency graph for stoper.cpp:



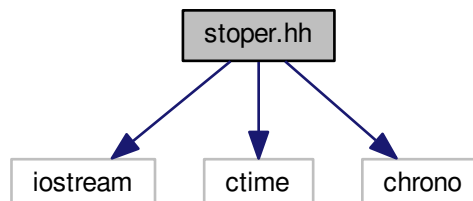
## 5.12 stoper.hh File Reference

```
#include <iostream>
```

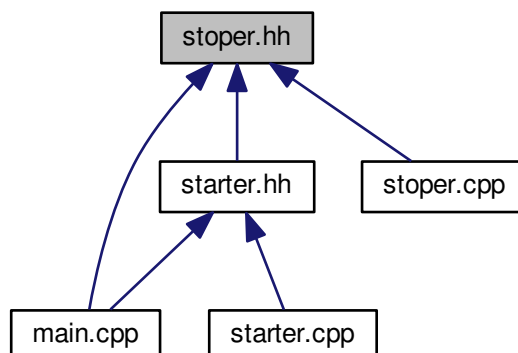
```
#include <ctime>
```

```
#include <chrono>
```

Include dependency graph for stoper.hh:



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



## Classes

- class `IStoper`

*Interfejs `IStoper`.*

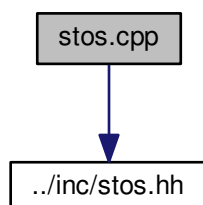
- class `Stoper`

*Klasa `stoper` implementująca interfejs `IStoper`.*

## 5.13 stos.cpp File Reference

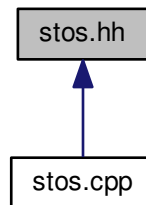
```
#include "../inc/stos.hh"
```

Include dependency graph for `stos.cpp`:



## 5.14 stos.hh File Reference

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



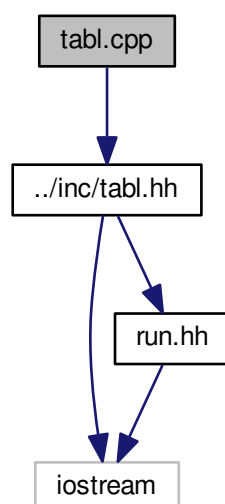
### Classes

- class `IStos< T >`
- class `Stos`

## 5.15 tabl.cpp File Reference

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

Include dependency graph for `tabl.cpp`:



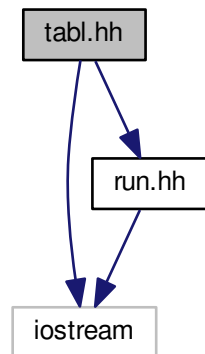
## 5.16 `tabl.hh` File Reference

Definicja interfejsu klasy `tabn`, klasy `tabn` oraz klasy `tabn_test`.

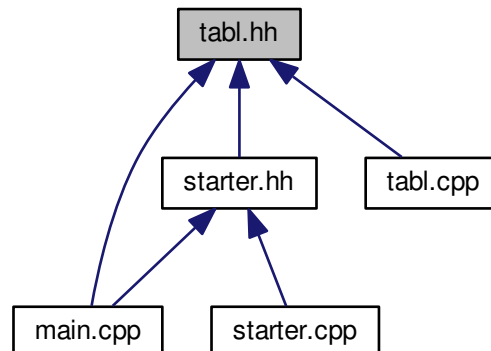
```
#include <iostream>
```

```
#include "run.hh"
```

Include dependency graph for `tabl.hh`:



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



### Classes

- class `ltabn< T >`  
*Typ wyliczeniowy decydujący o sposobie rozszerzania tablicy dynamicznej.*
- class `tabn< T >`  
*Modeluje tablicę dynamicznie rozszerzalną*
- class `tabn_test`  
*Definiuje sposób testowania tablicy `tabn`.*

## Macros

- `#define` [SIZE](#) 10

### 5.16.1 Detailed Description

Definicja interfejsu klasy `tabn`, klasy `tabn` oraz klasy [tabn\\_test](#).

### 5.16.2 Macro Definition Documentation

#### 5.16.2.1 `#define` `SIZE` 10

Definition at line 13 of file `tabl.hh`.

# Index

- ~ILista
  - ILista, [7](#)
- ~IStoper
  - IStoper, [9](#)
- ~Itabn
  - Itabn, [11](#)
- ~Runnable
  - Runnable, [14](#)
- ~Starter
  - Starter, [16](#)
- ~tabn
  - tabn, [23](#)
- ~tabn\_test
  - tabn\_test, [26](#)
- aSize
  - Itabn, [11](#)
  - tabn, [23](#)
- add
  - ILista, [8](#)
  - Itabn, [11](#)
  - tabn, [23](#)
- dumpToFile
  - Starter, [16](#)
- generateRandomDgt
  - Runnable, [14](#)
- get
  - ILista, [8](#)
- getElapsedTimeMs
  - IStoper, [9](#)
  - Stoper, [19](#)
- IKolejka< T >, [7](#)
- ILista
  - ~ILista, [7](#)
  - add, [8](#)
  - get, [8](#)
  - isEmpty, [8](#)
  - remove, [8](#)
  - size, [8](#)
- ILista< T >, [7](#)
- IStoper, [8](#)
  - ~IStoper, [9](#)
  - getElapsedTimeMs, [9](#)
  - start, [9](#)
  - stop, [9](#)
- IStos< T >, [10](#)
- isEmpty
  - ILista, [8](#)
- Itabn
  - ~Itabn, [11](#)
  - aSize, [11](#)
  - add, [11](#)
  - nOE, [12](#)
  - operator[], [12](#)
  - showElems, [12](#)
- Itabn< T >, [10](#)
- Kolejka, [12](#)
- kolejka.cpp, [29](#)
- kolejka.hh, [29](#)
- Lista, [13](#)
- lista.cpp, [30](#)
- lista.hh, [30](#)
- main
  - main.cpp, [31](#)
- main.cpp, [30](#)
  - main, [31](#)
- main.hh, [32](#)
- nOE
  - Itabn, [12](#)
  - tabn, [23](#)
- operator[]
  - Itabn, [12](#)
  - tabn, [24](#)
- prepare
  - Runnable, [14](#)
  - tabn\_test, [26](#)
- printResults
  - Starter, [16](#)
- remove
  - ILista, [8](#)
- run
  - Runnable, [14](#)
  - tabn\_test, [26](#)
- run.cpp, [32](#)
- run.hh, [33](#)
- Runnable, [13](#)
  - ~Runnable, [14](#)
  - generateRandomDgt, [14](#)
  - prepare, [14](#)
  - run, [14](#)
  - seedSrand, [15](#)

- SIZE
  - tabl.hh, [40](#)
- seedSrand
  - Runnable, [15](#)
- setTestSize
  - Starter, [17](#)
- showElems
  - ltabn, [12](#)
  - tabn, [24](#)
- size
  - lLista, [8](#)
- start
  - IStoper, [9](#)
  - Stoper, [21](#)
- Starter, [15](#)
  - ~Starter, [16](#)
  - dumpToFile, [16](#)
  - printResults, [16](#)
  - setTestSize, [17](#)
  - Starter, [16](#)
  - test, [17](#)
- starter.cpp, [34](#)
- starter.hh, [34](#)
- stop
  - IStoper, [9](#)
  - Stoper, [21](#)
- Stoper, [18](#)
  - getElapsedTimeMs, [19](#)
  - start, [21](#)
  - stop, [21](#)
- stoper.cpp, [36](#)
- stoper.hh, [36](#)
- Stos, [21](#)
- stos.cpp, [37](#)
- stos.hh, [38](#)
- tabl.cpp, [38](#)
- tabl.hh, [39](#)
  - SIZE, [40](#)
- tabn
  - ~tabn, [23](#)
  - aSize, [23](#)
  - add, [23](#)
  - nOE, [23](#)
  - operator[], [24](#)
  - showElems, [24](#)
  - tabn, [23](#)
- tabn< T >, [21](#)
- tabn\_test, [24](#)
  - ~tabn\_test, [26](#)
  - prepare, [26](#)
  - run, [26](#)
  - tabn\_test, [25](#)
- test
  - Starter, [17](#)