pamsi

0.4

Generated by Doxygen 1.8.9.1

Sat Mar 19 2016 13:45:04

Contents

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

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:

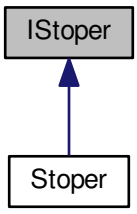
- lista.hh

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]

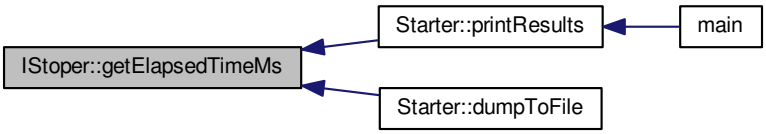
Definition at line 27 of file stoper.hh.

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:

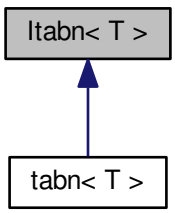
- stos.hh

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 rozmiary 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]

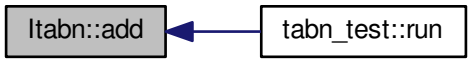
Definition at line 61 of file tabl.hh.

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:

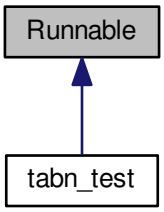
- lista.hh

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

Liczba	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

Zawsze	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

Zawsze	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 ()

  Kontruktor 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]

Kontruktor 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.

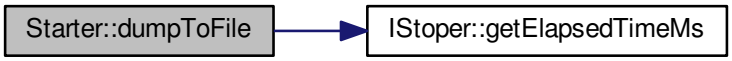
Format zapisu: wielkość_testu czas_trwania_ms

Parameters

in	-	nameOfFile - nazwa pliku wyjściowego

Definition at line 20 of file starter.cpp.

Here is the call graph for this function:

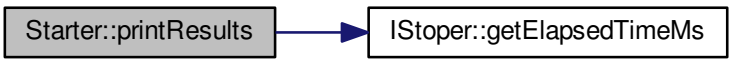


4.9.3.2 void Starter::printResults (void)

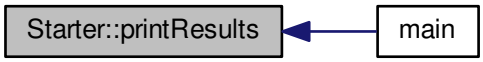
Metoda wyświetla czas trwania testu na standardowym wyjściu.

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::setTestSize (unsigned int *testsize*)

Metoda ustawia wielkość testu.

Parameters

in	*testsize*	- wielkość testu

Definition at line 3 of file starter.cpp.

Here is the call graph for this function:

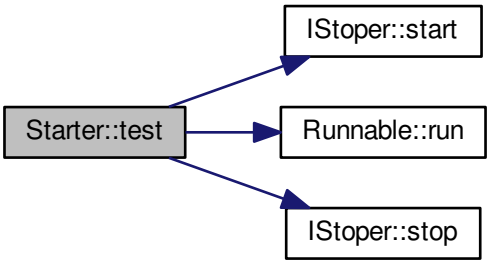
Here is the caller graph for this function:

4.9.3.4 void Starter::test (void)

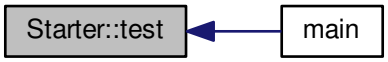
Metoda przeprowadza test.

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:

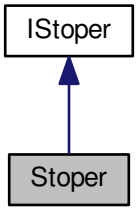
- starter.hh
- starter.cpp

4.10 Stoper Class Reference

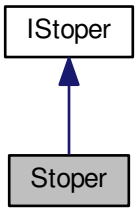
Klasa stoper implementująca interfejs IStoper.

`#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

Czas	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:

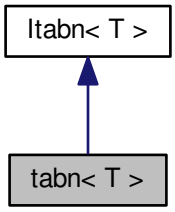
- stos.hh

4.12 tabn< T > Class Template Reference

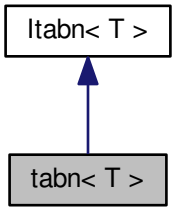
Modeluje tablicę dynamicznie rozszerzalną

`#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

element	- element do dodania

Implements Itabn< 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

int	Ilość zaalokowanych pól

Implements Itabn< 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

int	Liczba elementów w tablicy

Implements Itabn< 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

index	- numer elementu tablicy

Return values

*T**	Wskaźnik na wybrany element tablicy

Implements Itabn< 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

index	- numer elementu tablicy

Return values

T	Element tablicy

Implements Itabn< 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 Itabn< T >.

Definition at line 266 of file tabl.hh.

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

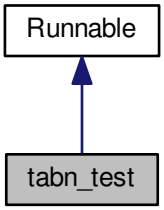
- tabl.hh

4.13 tabn_test Class Reference

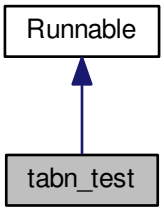
Definiuje sposób testowania tablicy tabn.

`#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

sizeOfTest	- rozmiar testu

Return values

bool	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()

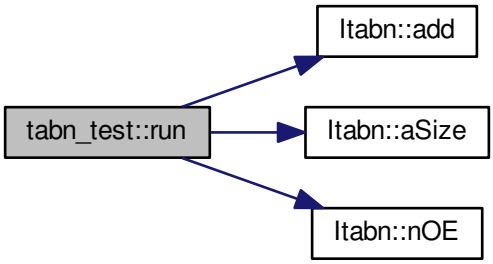
Return values

bool	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

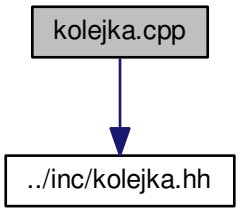
Chapter 5

File Documentation

5.1 kolejka.cpp File Reference

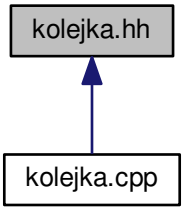
```
#include "../inc/kolejka.hh"
```

Include dependency graph for kolejka.cpp:



5.2 kolejka.hh File Reference

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

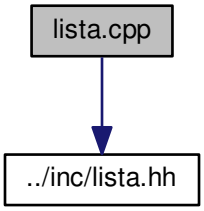


Classes

- class IKolejka< T >
- class Kolejka

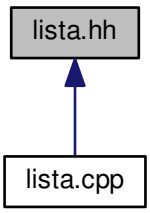
5.3 lista.cpp File Reference

```
#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

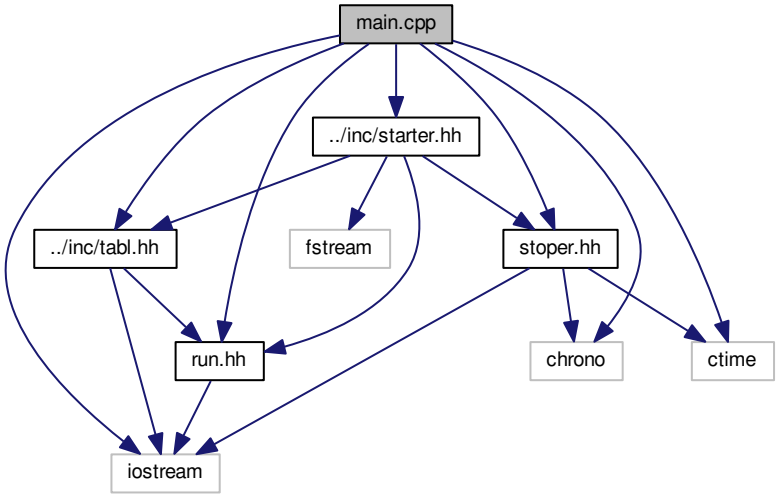
- class ILista< T >
- class Lista

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

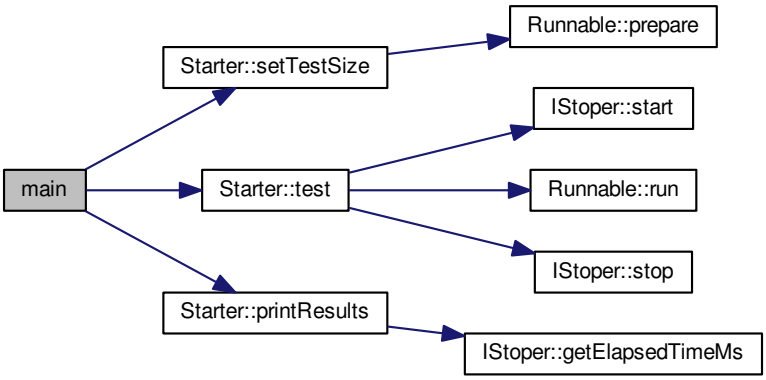
Główny plik programu.

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.

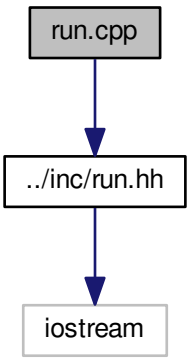
5.6.1 Detailed Description

Plik posiada wspólne definicje.

5.7 run.cpp File Reference

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

Include dependency graph for run.cpp:

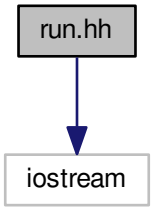


5.8 run.hh File Reference

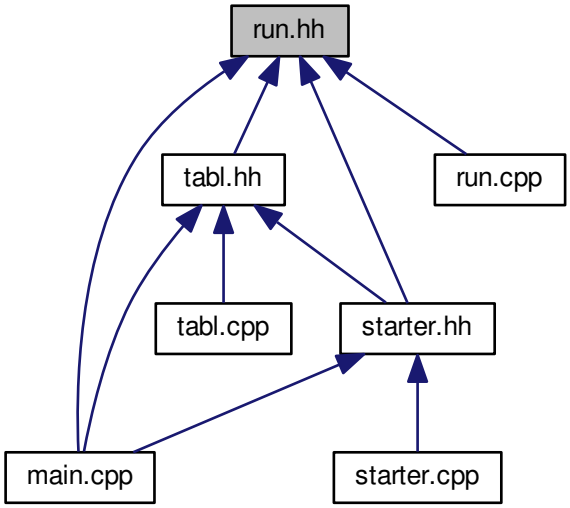
Plik definiuje klasę Runnable, ujednolicającą klasy umożliwiające badanie algorytmów.

```
#include <iostream>
```

Include dependency graph for run.hh:



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



Classes

- class Runnable

  Klasa ujednolica sposób uruchamiania klasy badającej algorytm.

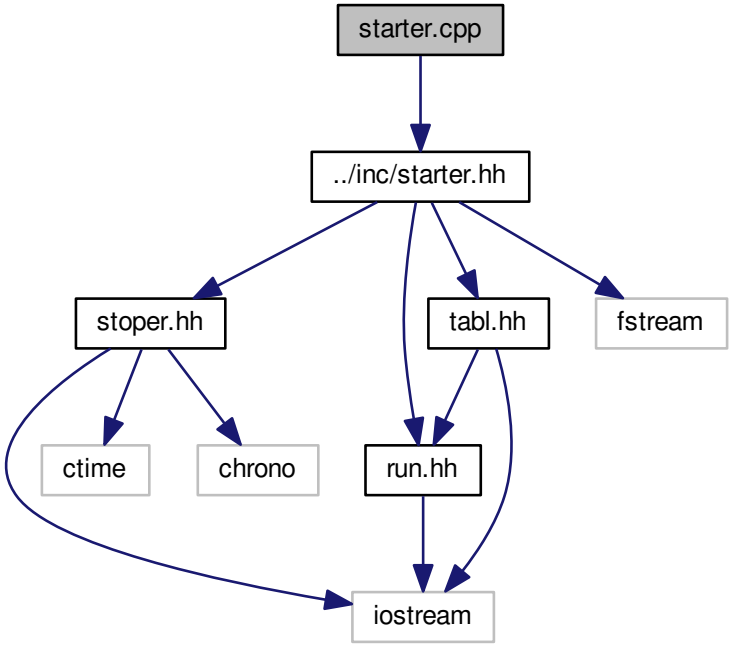
5.8.1 Detailed Description

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:

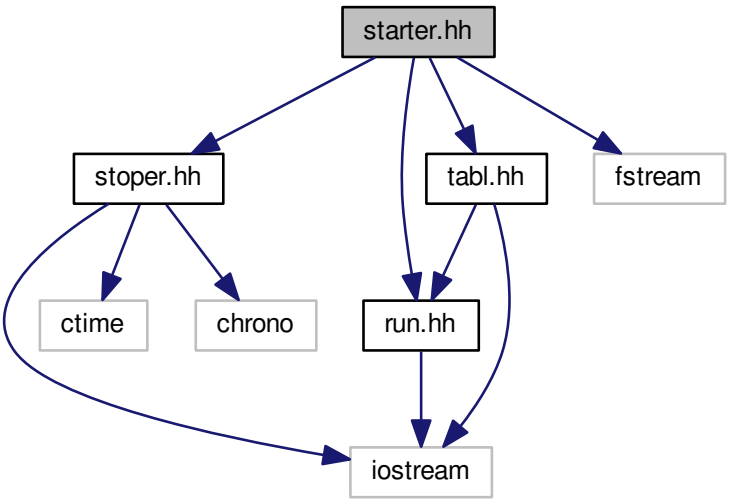


5.10 starter.hh File Reference

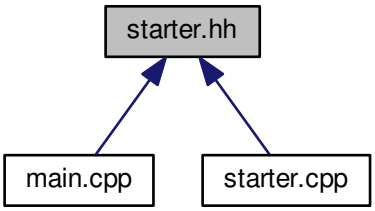
Plik definiuje klasę Starter.

```
#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

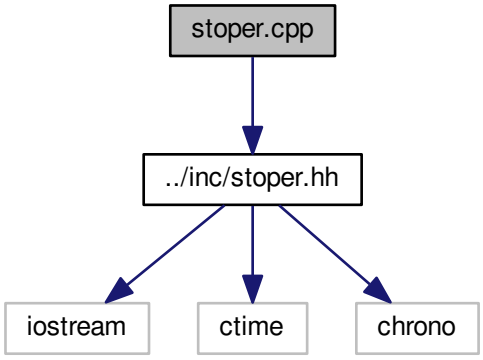
  Klasa pozwala na przeprowadzenie testów.

5.10.1 Detailed Description

Plik definiuje klasę Starter.

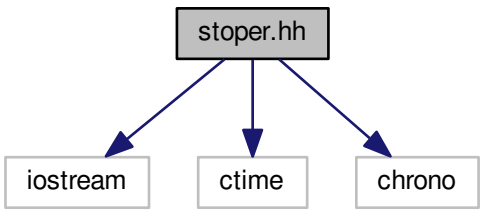
5.11 stoper.cpp File Reference

```
#include "../inc/stoper.hh"
```
Include dependency graph for stoper.cpp:

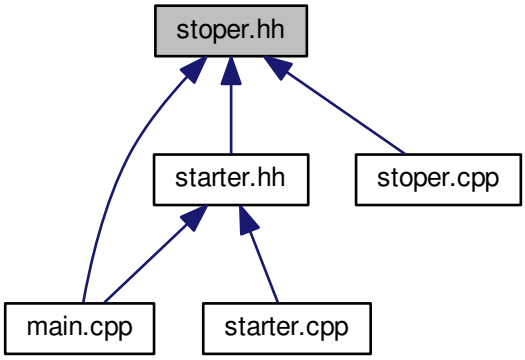


5.12 stoper.hh File Reference

```
#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

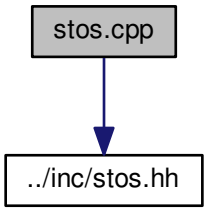
  Interfejs IStoper.

- class Stoper

  Klasa stoper implementująca interfejs IStoper.

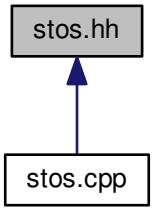
5.13 stos.cpp File Reference

`#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:



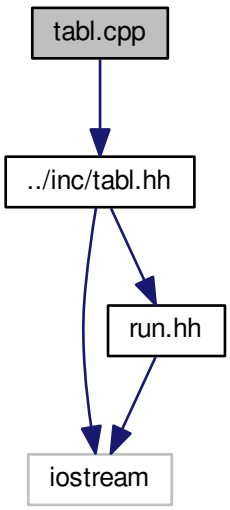
Classes

- class IStos< T >
- class Stos

5.15 tabl.cpp File Reference

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

Include dependency graph for tabl.cpp:

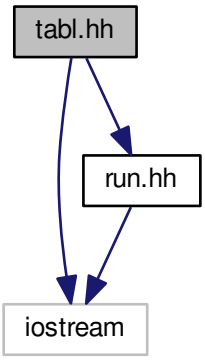


5.16 tabl.hh File Reference

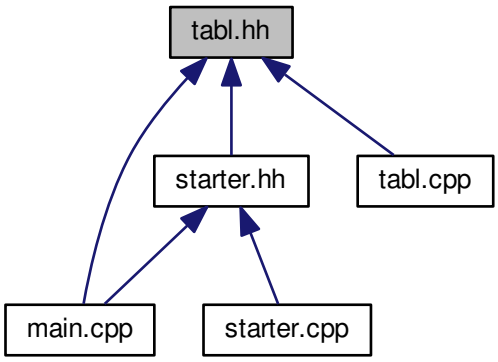
Definicja interfejsu klasy tabn, klasy tabn oraz klasy tabn_test.

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

Include dependency graph for tabl.hh:



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



Classes

- class Itabn< 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