

Pomiar czasu dynamicznej alokacji pamięci

Generated by Doxygen 1.8.6

Sun Mar 13 2016 20:32:17

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	IRunnable Class Reference	7
4.1.1	Member Function Documentation	7
4.1.1.1	prepare	7
4.1.1.2	run	7
4.2	IStoper Class Reference	8
4.2.1	Member Function Documentation	8
4.2.1.1	dumpToFile	8
4.2.1.2	getElapsedTime	8
4.2.1.3	start	8
4.2.1.4	stop	8
4.3	Sedzia Class Reference	8
4.3.1	Member Function Documentation	9
4.3.1.1	setOff	9
4.4	Stoper Class Reference	9
4.4.1	Constructor & Destructor Documentation	10
4.4.1.1	Stoper	10
4.4.1.2	~Stoper	11
4.4.2	Member Function Documentation	11
4.4.2.1	dumpToFile	11
4.4.2.2	getElapsedTime	11
4.4.2.3	start	11
4.4.2.4	stop	12

4.4.3	Member Data Documentation	12
4.4.3.1	_start	12
4.4.3.2	_stop	12
4.4.3.3	my_file	12
4.5	Tablica Class Reference	12
4.5.1	Constructor & Destructor Documentation	14
4.5.1.1	Tablica	14
4.5.1.2	~Tablica	14
4.5.2	Member Function Documentation	14
4.5.2.1	getSize	14
4.5.2.2	increaseSize	14
4.5.2.3	isFull	14
4.5.2.4	prepare	14
4.5.2.5	printAllElements	15
4.5.2.6	run	15
4.5.3	Member Data Documentation	15
4.5.3.1	current_size	15
4.5.3.2	desired_size	15
4.5.3.3	elements	15
4.5.3.4	index	16
5	File Documentation	17
5.1	/home/kkuczaj/PAMSI/lab_07.03/prj/inc/IRunnable.h File Reference	17
5.2	/home/kkuczaj/PAMSI/lab_07.03/prj/inc/IStoper.h File Reference	18
5.3	/home/kkuczaj/PAMSI/lab_07.03/prj/inc/Sedzia.h File Reference	18
5.4	/home/kkuczaj/PAMSI/lab_07.03/prj/inc/Stoper.h File Reference	19
5.5	/home/kkuczaj/PAMSI/lab_07.03/prj/inc/Tablica.h File Reference	20
5.6	/home/kkuczaj/PAMSI/lab_07.03/prj/src/main.cpp File Reference	21
5.6.1	Function Documentation	22
5.6.1.1	main	22
5.6.1.2	zapiszDoPojemnika	22
5.7	/home/kkuczaj/PAMSI/lab_07.03/prj/src/Sedzia.cpp File Reference	22
5.8	/home/kkuczaj/PAMSI/lab_07.03/prj/src/Stoper.cpp File Reference	23
5.9	/home/kkuczaj/PAMSI/lab_07.03/prj/src/Tablica.cpp File Reference	23

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

IRunnable	7
Tablica	12
IStoper	8
Stoper	9
Sedzia	8

Chapter 2

Class Index

2.1 Class List

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

IRunnable	Interfejs dla biegacza	7
IStoper	Interfejs dla stopera	8
Sedzia	Implementacja klasy Sedzia	8
Stoper	Implementacja klasy Stoper	9
Tablica	Klasa tablica, w której odbywa się zapis dynamiczny elementów typu int	12

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

/home/kkuczaj/PAMSI/lab_07.03/prj/inc/ IRunnable.h	
Naglowek zawierajacy interfejs dla biegacza	17
/home/kkuczaj/PAMSI/lab_07.03/prj/inc/ IStoper.h	
Naglowek zawierajacy interfejs dla stopera	18
/home/kkuczaj/PAMSI/lab_07.03/prj/inc/ Sedzia.h	
Naglowek opisujacy implementacje Sedziego	18
/home/kkuczaj/PAMSI/lab_07.03/prj/inc/ Stoper.h	
Implementacja interfejsu IStoper w klasie Stoper	19
/home/kkuczaj/PAMSI/lab_07.03/prj/inc/ Tablica.h	
Implementacja interfejsu IRunnable	20
/home/kkuczaj/PAMSI/lab_07.03/prj/src/ main.cpp	21
/home/kkuczaj/PAMSI/lab_07.03/prj/src/ Sedzia.cpp	22
/home/kkuczaj/PAMSI/lab_07.03/prj/src/ Stoper.cpp	23
/home/kkuczaj/PAMSI/lab_07.03/prj/src/ Tablica.cpp	23

Chapter 4

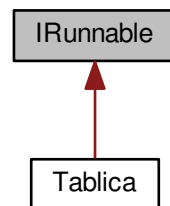
Class Documentation

4.1 IRunnable Class Reference

Interfejs dla biegacza.

```
#include <IRunnable.h>
```

Inheritance diagram for IRunnable:



Private Member Functions

- virtual void [prepare](#) (unsigned int size)=0
- virtual void [run](#) ()=0

4.1.1 Detailed Description

Interfejs dla biegacza.

Klasa abstrakcyjna z metodami czysto wirtualnymi.

4.1.2 Member Function Documentation

4.1.2.1 virtual void IRunnable::prepare (unsigned int *size*) [private],[pure virtual]

Implemented in [Tablica](#).

4.1.2.2 `virtual void IRunnable::run () [private],[pure virtual]`

Implemented in [Tablica](#).

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

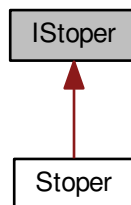
- [/home/kkuczaj/PAMSI/lab_07.03/prj/inc/IRunnable.h](#)

4.2 IStoper Class Reference

Interfejs dla stopera.

```
#include <IStoper.h>
```

Inheritance diagram for IStoper:



Public Member Functions

- `virtual void start ()=0`
- `virtual void stop ()=0`
- `virtual double getElapsedTime ()=0`
- `virtual void dumpToFile ()=0`

4.2.1 Detailed Description

Interfejs dla stopera.

Klasa abstrakcyjna z metodami czysto wirtualnymi.

4.2.2 Member Function Documentation

4.2.2.1 `virtual void IStoper::dumpToFile () [pure virtual]`

Implemented in [Stoper](#).

4.2.2.2 `virtual double IStoper::getElapsedTime () [pure virtual]`

Implemented in [Stoper](#).

4.2.2.3 `virtual void IStoper::start () [pure virtual]`

Implemented in [Stoper](#).

4.2.2.4 `virtual void IStoper::stop () [pure virtual]`

Implemented in [Stoper](#).

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

- `/home/kkuczaj/PAMSI/lab_07.03/prj/inc/IStoper.h`

4.3 Sedzia Class Reference

Implementacja klasy [Sedzia](#).

```
#include <Sedzia.h>
```

Public Member Functions

- `bool setOff (unsigned int how_many)`

Funkcja, w ktorej odbywa sie bieg.

4.3.1 Detailed Description

Implementacja klasy [Sedzia](#).

[Sedzia](#) wykorzystuje elementy klasy [Stoper](#) oraz klasy [Tablica](#). Mierzy czas wypelniania elemntow Tablicy.

4.3.2 Member Function Documentation

4.3.2.1 `bool Sedzia::setOff (unsigned int how_many)`

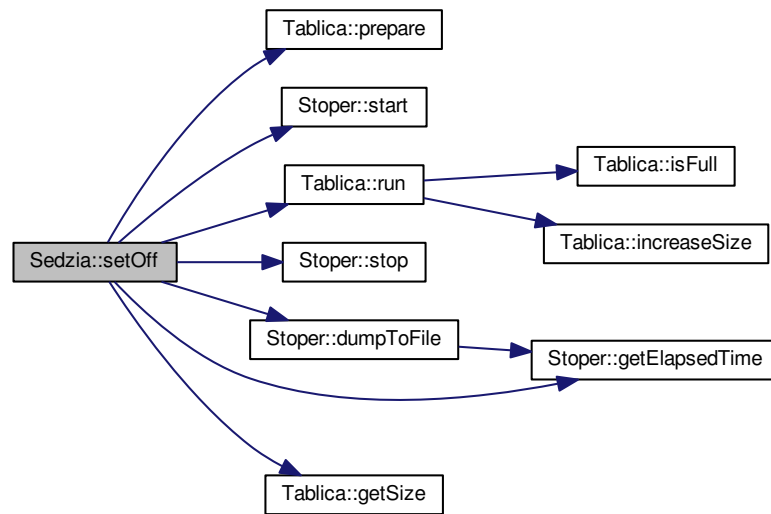
Funkcja, w ktorej odbywa sie bieg.

Podczas wykonywania tej funkcji uruchamiany jest [Stoper](#) oraz wypelniany jest element klasy Tablica po uprzednim jej przygotowaniu.

Parameters

<i>how_many</i>	Informacja iloma elementami ma zostac wypelniona tablica.
-----------------	---

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:

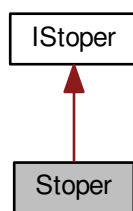
- `/home/kkuczaj/PAMSI/lab_07.03/prj/inc/Sedzia.h`
- `/home/kkuczaj/PAMSI/lab_07.03/prj/src/Sedzia.cpp`

4.4 Stoper Class Reference

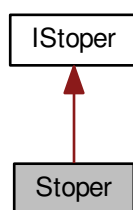
Implementacja klasy [Stoper](#).

```
#include <Stoper.h>
```

Inheritance diagram for Stoper:



Collaboration diagram for Stoper:



Public Member Functions

- virtual void [start](#) ()
Implementacja funkcji [start\(\)](#) z interfejsu [IStoper](#).
- virtual void [stop](#) ()
Implementacja funkcji [stop\(\)](#) z interfejsu [IStoper](#).
- virtual double [getElapsedTime](#) ()
Implementacja funkcji [getElapse\(\)](#) z interfejsu [IStoper](#).
- virtual void [dumpToFile](#) ()
Implementacja funkcji [dumpToFile\(\)](#) z interfejsu [IStoper](#).

Private Attributes

- clock_t [_start](#)
Moment startu stopera.
- clock_t [_stop](#)
Moment zatrzymania stopera.
- std::fstream [my_file](#)
Strumień zapisu do pliku.

Additional Inherited Members

4.4.1 Detailed Description

Implementacja klasy [Stoper](#).

W klasie [Stoper](#) zostały zaimplementowane metody pozwalające na pomiar czasu. Pomiar czasu odbywa się dzięki bibliotece `<ctime>` a zapis do pliku korzysta z biblioteki `<fstream>`.

4.4.2 Member Function Documentation

4.4.2.1 `void Stoper::dumpToFile () [virtual]`

Implementacja funkcji [dumpToFile\(\)](#) z interfejsu [IStoper](#).

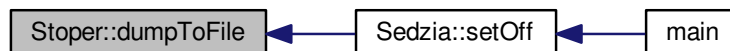
Zapisuje zmierzony czas do pliku o nazwie "tmp_czasy.txt".

Implements [IStoper](#).

Here is the call graph for this function:



Here is the caller graph for this function:



4.4.2.2 `double Stoper::getElapsedTime () [virtual]`

Implementacja funkcji `getElapse()` z interfejsu [IStoper](#).

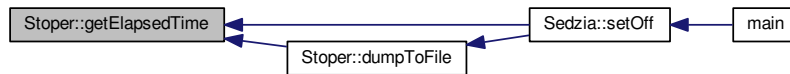
Oblicza czas pomiędzy czasem zapisanym w zmiennych `_start` i `_stop`.

Returns

Zwraca zmierzony czas - roznica pomiedzy polem `_start` a polem `_stop`.

Implements [IStoper](#).

Here is the caller graph for this function:

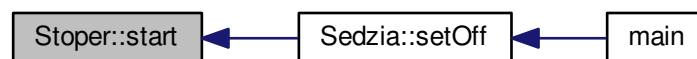
**4.4.2.3 void Stoper::start () [virtual]**

Implementacja funkcji [start\(\)](#) z interfejsu [IStoper](#).

Zapisuje moment uruchomienia stopera.

Implements [IStoper](#).

Here is the caller graph for this function:

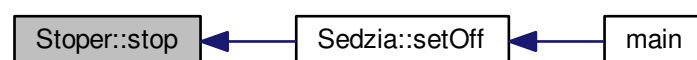
**4.4.2.4 void Stoper::stop () [virtual]**

Implementacja funkcji [stop\(\)](#) z interfejsu [IStoper](#).

Zapisuje moment zatrzymania stopera.

Implements [IStoper](#).

Here is the caller graph for this function:



4.4.3 Member Data Documentation

4.4.3.1 `clock_t Stoper::_start` [private]

Moment startu stopera.

Element przechowujący informacje o czasie systemowym w momencie uruchomienia stopera. Element typu `clock_t`.

4.4.3.2 `clock_t Stoper::_stop` [private]

Moment zatrzymania stopera.

Element przechowujący informacje o czasie systemowym w momencie zatrzymania stopera. Element typu `clock_t`.

4.4.3.3 `std::fstream Stoper::my_file` [private]

Strumień zapisu do pliku.

Pole ułatwiające zapis do pliku.

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

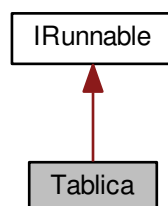
- [/home/kkuczaj/PAMSI/lab_07.03/prj/inc/Stoper.h](#)
- [/home/kkuczaj/PAMSI/lab_07.03/prj/src/Stoper.cpp](#)

4.5 Tablica Class Reference

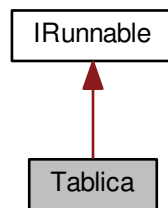
Klasa tablica, w której odbywa się zapis dynamiczny elementów typu `int`.

```
#include <Tablica.h>
```

Inheritance diagram for Tablica:



Collaboration diagram for Tablica:



Public Member Functions

- `Tablica` (int x=10)
Konstruktor parametryczny.
- `~Tablica` ()
Destruktor.
- virtual void `prepare` (unsigned int size)
Implementacja funkcji `prepare()` interfejsu `IRunnable`.
- virtual void `run` ()
Implementacja funkcji `run()` interfejsu `IRunnable`.
- unsigned int `getSize` ()
Zwraca aktualny rozmiar tablicy dynamicznej.

Private Member Functions

- bool `isFull` ()
Pozwala prosto okreslic, czy nalezy przydzielic pamiec.
- void `increaseSize` ()
Zwieksza rozmiar przydzielonej pamieci na stercie.

Private Attributes

- int * `elements`
Wskaźnik do początku tablicy dynamicznej.
- unsigned int `current_size`
Okresla aktualny rozmiar stosu.
- unsigned int `desired_size`
Okresla pozadany rozmiar stosu.
- unsigned int `index`
Okresla aktualny indeks.

4.5.1 Detailed Description

Klasa `tablica`, w której odbywa się zapis dynamiczny elementów typu `int`.

Implementuje metody interfejsu `IRunnable`. Zajmuje się dynamiczną alokacją pamięci.

4.5.2 Constructor & Destructor Documentation

4.5.2.1 `Tablica::Tablica (int x = 10)`

Konstruktor parametryczny.

Umożliwia określenie początkowego rozmiaru tablicy. W przypadku braku określenia tego rozmiaru przyjmuje domyślną wartość równą 10.

Parameters

<code>x</code>	Określa początkową wielkość przydzielonej pamięci. Domyślna wartość w przypadku braku podania to 10.
----------------	--

4.5.2.2 `Tablica::~~Tablica ()`

Destruktor.

Usuwa pamięć przypisaną komórce, na którą wskazuje pole `*elements`.

4.5.3 Member Function Documentation

4.5.3.1 `unsigned int Tablica::getSize ()`

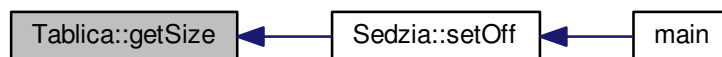
Zwraca aktualny rozmiar tablicy dynamicznej.

Zwraca wartość pola `current_size`.

Returns

Zwraca wartość typu `unsigned int`, gdyż takiego typu jest zmienna `current_size`.

Here is the caller graph for this function:

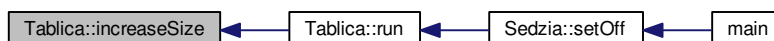


4.5.3.2 `void Tablica::increaseSize ()` `[private]`

Zwiększa rozmiar przydzielonej pamięci na stacku.

Metoda prywatna. Kopiuje elementy starej pamięci do komórki z nowo-przydzieloną pamięcią. Usuwa starą pamięć.

Here is the caller graph for this function:



4.5.3.3 bool Tablica::isFull () [private]

Pozwala prosto okreslic, czy nalezy przydzielic pamiec.

Metoda prywatna. Sluzy do okreslania czy nalezy wywolac metode [increaseSize\(\)](#).

Returns

true Pamiec pelna. Nalezy zwiekszyc rozmiar.
false Jest jeszcze wolne miejsce.

Here is the caller graph for this function:



4.5.3.4 void Tablica::prepare (unsigned int size) [virtual]

Implementacja funkcji [prepare\(\)](#) interfesju [IRunnable](#).

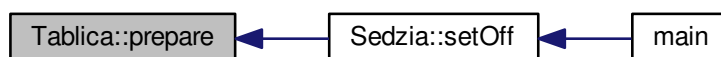
Zapisuje pozadany rozmiar do pola desired_size.

Parameters

<i>size</i>	Parametr typu unsigned int, gdyz rozmiar nie powinien nigdy byc ujemny. Jego wartosc zapisywana jest do pola desired_size.
-------------	--

Implements [IRunnable](#).

Here is the caller graph for this function:



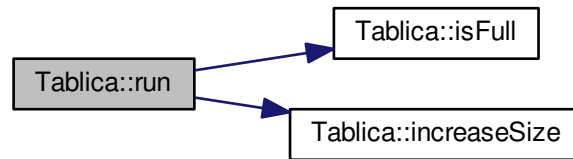
4.5.3.5 void Tablica::run () [virtual]

Implementacja funkcji [run\(\)](#) interfesju [IRunnable](#).

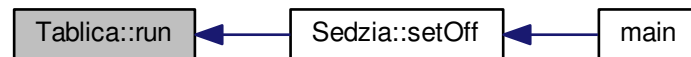
Uruchamia "bieg", w ktorym nastepuje zapis elementow do poszczegolnych elementow tablicy dynamicznej. Tam odbywa sie alokacja pamieci oraz instrukcje warunkowe.

Implements [IRunnable](#).

Here is the call graph for this function:



Here is the caller graph for this function:



4.5.4 Member Data Documentation

4.5.4.1 `unsigned int Tablica::current_size` [private]

Okresla aktualny rozmiar stosu.

Pole prywatne typu `unsigned int`, gdyz rozmiar nigdy nie powinien byc ujemny.

4.5.4.2 `unsigned int Tablica::desired_size` [private]

Okresla pozadany rozmiar stosu.

Pole prywatne typu `unsigned int`, gdyz rozmiar nigdy nie powinien byc ujemny. Zadawane w funkcji [prepare\(\)](#).

4.5.4.3 `int* Tablica::elements` [private]

Wskaźnik do początku tablicy dynamicznej.

Wskazuje na adres w pamięci serty. Pole prywatne.

4.5.4.4 `unsigned int Tablica::index` [private]

Okresla aktualny indeks.

Pole prywatne typu `unsigned int`, gdyz indeks nigdy nie powinien byc ujemny. Przechowuje indeks, pierwszej wolnej komórki pamięci, do ktorego mozliwy bedzie zapis.

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

- [/home/kkuczaj/PAMSI/lab_07.03/prj/inc/Tablica.h](#)
- [/home/kkuczaj/PAMSI/lab_07.03/prj/src/Tablica.cpp](#)

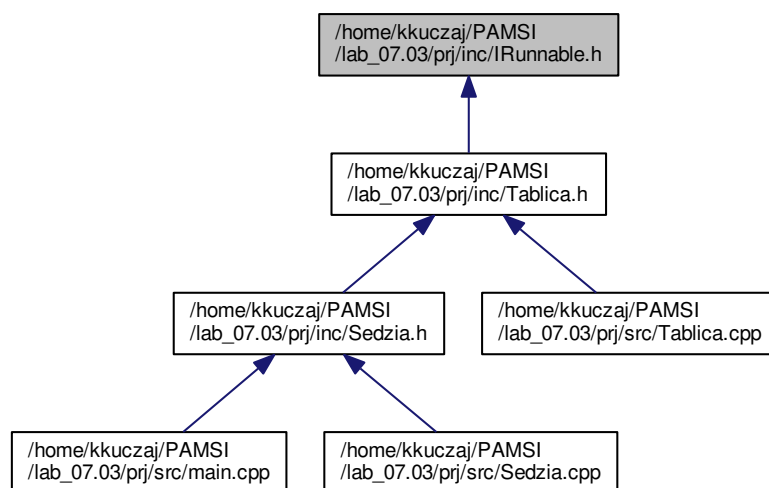
Chapter 5

File Documentation

5.1 /home/kkuczaj/PAMSI/lab_07.03/prj/inc/IRunnable.h File Reference

Naglowek zawierajacy interfejs dla biegacza.

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



Classes

- class [IRunnable](#)
Interfejs dla biegacza.

5.1.1 Detailed Description

Naglowek zawierajacy interfejs dla biegacza.

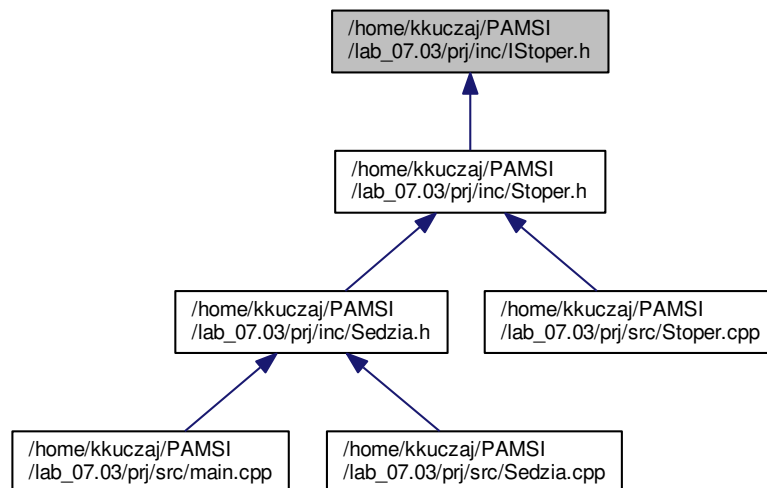
Author

Kamil Kuczaj

5.2 /home/kkuczaj/PAMSI/lab_07.03/prj/inc/IStoper.h File Reference

Naglowek zawierajacy interfejs dla stopera.

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



Classes

- class [IStoper](#)

Interfejs dla stopera.

5.2.1 Detailed Description

Naglowek zawierajacy interfejs dla stopera.

Author

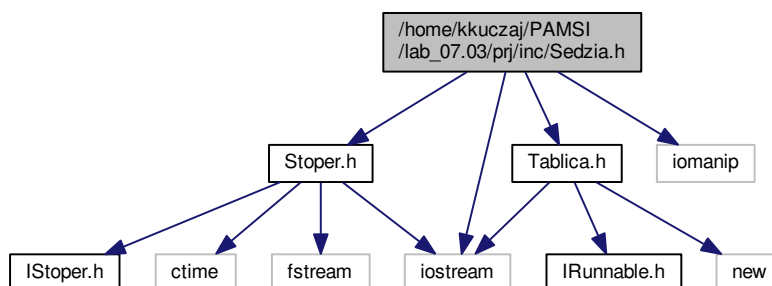
Kamil Kuczaj

5.3 /home/kkuczaj/PAMSI/lab_07.03/prj/inc/Sedzia.h File Reference

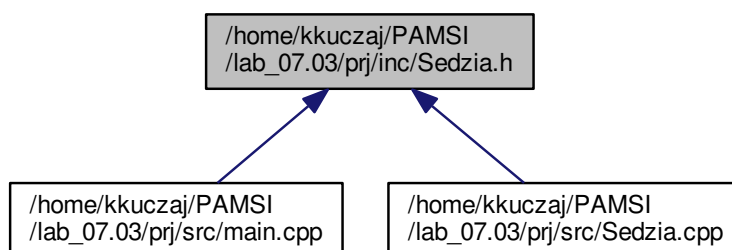
Naglowek opisujacy implementacje Sedziego.

```
#include "Stoper.h"
#include "Tablica.h"
#include <iostream>
#include <iomanip>
```

Include dependency graph for Sedzia.h:



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



Classes

- class [Sedzia](#)

Implementacja klasy [Sedzia](#).

5.3.1 Detailed Description

Nagłówek opisujący implementację Sedziego.

Author

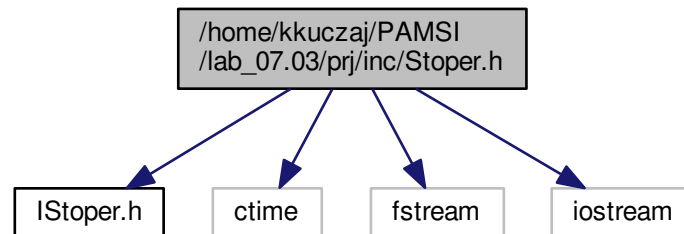
Kamil Kuczaj

5.4 /home/kkuczaj/PAMSI/lab_07.03/prj/inc/Stoper.h File Reference

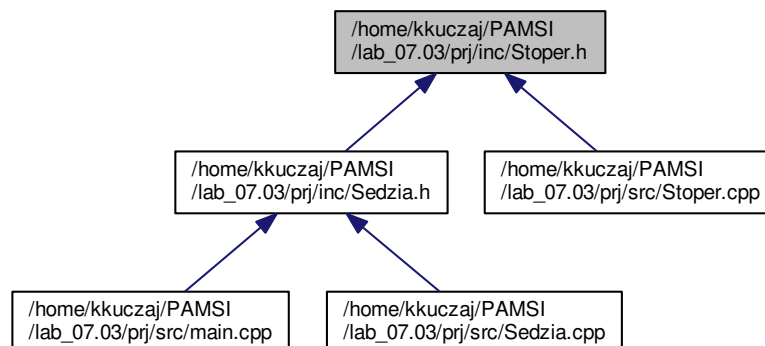
Implementacja interfejsu [IStoper](#) w klasie [Stoper](#).

```
#include "IStoper.h"
#include <ctime>
#include <fstream>
#include <iostream>
```

Include dependency graph for Stoper.h:



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



Classes

- class [Stoper](#)

Implementacja klasy [Stoper](#).

5.4.1 Detailed Description

Implementacja interfejsu [IStoper](#) w klasie [Stoper](#).

Author

Kamil Kuczaj

5.5 /home/kkuczaj/PAMSI/lab_07.03/prj/inc/Tablica.h File Reference

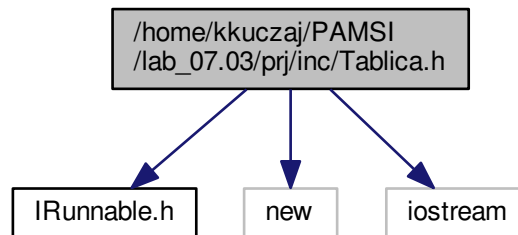
Implementacja interfejsu [IRunnable](#).

```
#include "IRunnable.h"
```

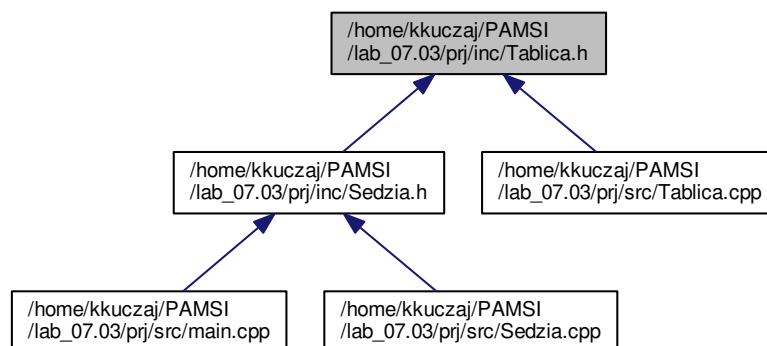
```
#include <new>
```

```
#include <iostream>
```

Include dependency graph for Tablica.h:



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



Classes

- class [Tablica](#)

Klasa tablica, w której odbywa się zapis dynamiczny elementów typu int.

5.5.1 Detailed Description

Implementacja interfejsu [IRunnable](#).

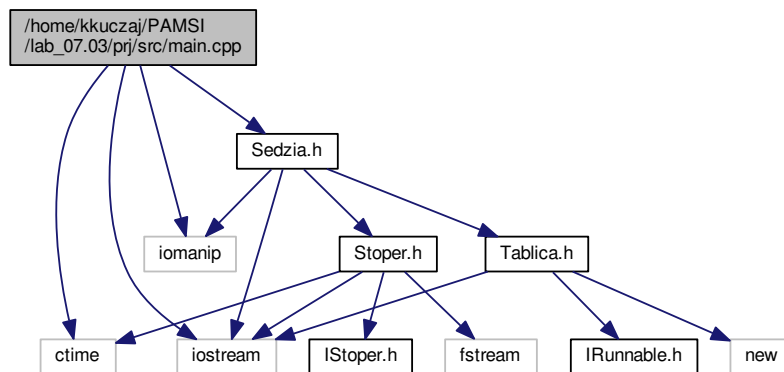
Author

Kamil Kuczaj

5.6 /home/kkuczaj/PAMSI/lab_07.03/prj/src/main.cpp File Reference

```
#include <iostream>
#include <ctime>
#include <iomanip>
#include "Sedzia.h"
```

Include dependency graph for main.cpp:



Functions

- int [main](#) (int argc, char **argv)

5.6.1 Detailed Description

Author

Kamil Kuczaj 218478@student.pwr.edu.pl

Date

5.6.2 LICENCJA

Program udostępniam na licencji GPLv3.

5.6.3 OPIS

Program został zbudowany modułowo. W folderze inc/ znajdują się pliki nagłówkowe. Folder src/ zawiera pliki źródłowe. W głównym folderze zbudowany został Makefile. Pliki obiektowe są budowane w folderze obj/ a następnie linkowane do głównego folderu (prj/). Testowano przy wykorzystaniu kompilatora g++ w wersji 4.8.4 na systemie Linux Ubuntu 14.04.04 opartego o jądro 4.2.0-30-generic.

Aby zbudowac i jednocześnie odpalic program: `$ make`

Aby pozbyc sie plikow z koncowka `*~` lub zaczynajacych sie na `#*`: `$ make order`

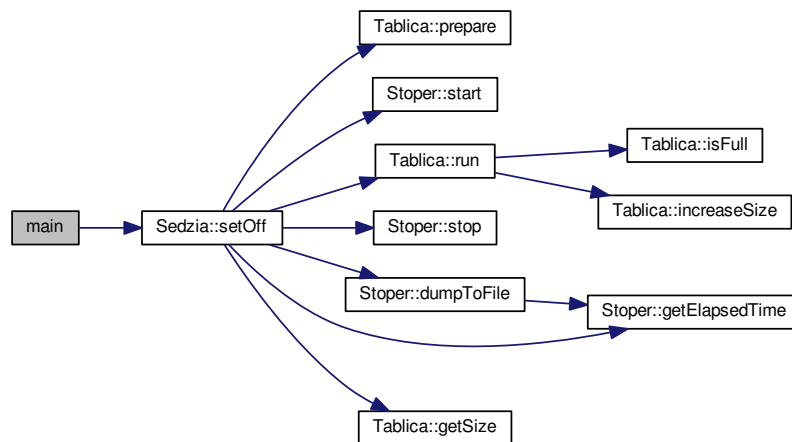
Aby pozbyc sie programu wykonywalnego oraz plikow obiektowych: `$ make clean`

Aby wyswietlic pomoc do pliku Makefile: `$ make help`

5.6.4 Function Documentation

5.6.4.1 `int main (int argc, char ** argv)`

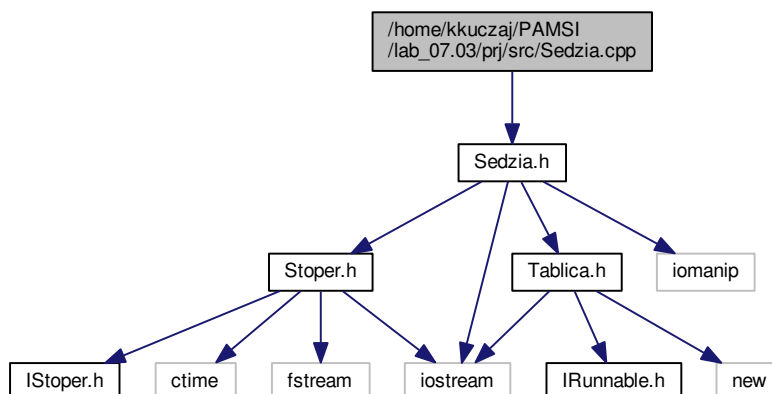
Here is the call graph for this function:



5.7 /home/kkuczaj/PAMSI/lab_07.03/prj/src/Sedzia.cpp File Reference

```
#include "Sedzia.h"
```

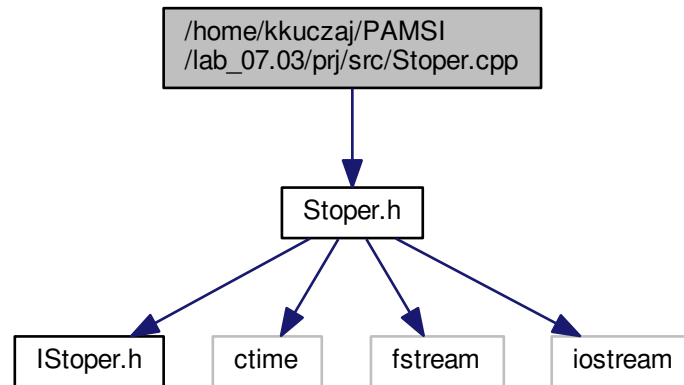
Include dependency graph for Sedzia.cpp:



5.8 /home/kkuczaj/PAMSI/lab_07.03/prj/src/Stoper.cpp File Reference

```
#include "Stoper.h"
```

Include dependency graph for Stoper.cpp:



5.9 /home/kkuczaj/PAMSI/lab_07.03/prj/src/Tablica.cpp File Reference

```
#include "Tablica.h"
```

Include dependency graph for Tablica.cpp:

