

Pomiar czasu dynamicznie alokowanej pamięci w tablicy dynamicznej.

Generated by Doxygen 1.8.6

Tue Mar 29 2016 21:57:02

Contents

Chapter 1

Opis programu

Author

Kamil Kuczaj 218478@student.pwr.edu.pl

1.1 Wstęp

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.

1.2 Licencja

Program udostępniam na licencji GPLv3.

1.3 Instalacja

Aby zbudować i jednocześnie odpalić program: `$ make`

Aby pozbyć się plików z końcówką `*~` lub zaczynających się na `#*`: `$ make order`

Aby pozbyć się programu wykonywalnego oraz plików obiektowych: `$ make clean`

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

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

IRunnable	??
Tablica< Type >	??
IStoper	??
Stoper	??
Sedzia	??

Chapter 3

Class Index

3.1 Class List

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

IRunnable	Interfejs dla biegacza	??
IStoper	Interfejs dla stopera	??
Sedzia	Implementacja klasy Sedzia	??
Stoper	Implementacja klasy Stoper	??
Tablica< Type >	Klasa Tablica , w której odbywa się zapis dynamiczny elementów	??

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

/home/kkuczaj/PAMSI/lab02_07.03/prj/inc/ IRunnable.h	
Naglowek zawierajacy interfejs dla biegacza	??
/home/kkuczaj/PAMSI/lab02_07.03/prj/inc/ IStoper.h	
Naglowek zawierajacy interfejs dla stopera	??
/home/kkuczaj/PAMSI/lab02_07.03/prj/inc/ Sedzia.h	
Naglowek opisujacy implementacje Sedziego	??
/home/kkuczaj/PAMSI/lab02_07.03/prj/inc/ Stoper.h	
Implementacja interfejsu IStoper w klasie Stoper	??
/home/kkuczaj/PAMSI/lab02_07.03/prj/inc/ Tablica.h	
Implementacja interfejsu IRunnable	??
/home/kkuczaj/PAMSI/lab02_07.03/prj/src/ main.cpp	??
/home/kkuczaj/PAMSI/lab02_07.03/prj/src/ Sedzia.cpp	??
/home/kkuczaj/PAMSI/lab02_07.03/prj/src/ Stoper.cpp	??
/home/kkuczaj/PAMSI/lab02_07.03/prj/src/ Tablica.cpp	??

Chapter 5

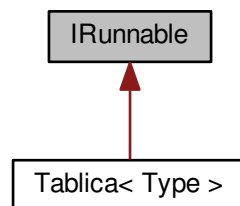
Class Documentation

5.1 IRunnable Class Reference

Interfejs dla biegacza.

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

Inheritance diagram for IRunnable:



Protected Member Functions

- virtual void `prepare` (unsigned int size)=0
Przygotowanie pojemnika przed pomiarem czasu.
- virtual void `run` ()=0
Uruchomienie czynności, która będziemy mierzyć stoperem.

5.1.1 Detailed Description

Interfejs dla biegacza.

Klasa abstrakcyjna z metodami czysto wirtualnymi.

5.1.2 Member Function Documentation

5.1.2.1 `virtual void IRunnable::prepare (unsigned int size) [protected],[pure virtual]`

Przygotowanie pojemnika przed pomiarem czasu.

Metoda czysto wirtualna.

Implemented in [Tablica< Type >](#).

5.1.2.2 `virtual void IRunnable::run () [protected],[pure virtual]`

Uruchomienie czynnosci, ktora bedziemy mierzyc stoperem.

Metoda czysto wirtualna.

Implemented in [Tablica< Type >](#).

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

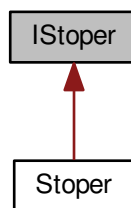
- [/home/kkuczaj/PAMSI/lab02_07.03/prj/inc/IRunnable.h](#)

5.2 IStoper Class Reference

Interfejs dla stopera.

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

Inheritance diagram for IStoper:



Protected Member Functions

- `virtual void start ()=0`
Ma symulowac moment startu stopera.
- `virtual void stop ()=0`
- `virtual double getElapsedTime ()=0`
Ma symulowac rezultat pokazania wyniku pomiaru czasu na stoperze.
- `virtual void dumpToFile (std::string file_name)=0`
Ma symulowac moment zapisu zmierzonego czasu na kartke papieru.

5.2.1 Detailed Description

Interfejs dla stopera.

Klasa abstrakcyjna z metodami czysto wirtualnymi.

5.2.2 Member Function Documentation

5.2.2.1 `virtual void IStoper::dumpToFile (std::string file_name)` `[protected]`, `[pure virtual]`

Ma symulowac moment zapisu zmierzonego czasu na kartke papieru.

Metoda czysto wirtualna.

Parameters

<i>file_name</i>	Nazwa pliku. Obiekt klasy string.
------------------	-----------------------------------

Implemented in [Stoper](#).

5.2.2.2 `virtual double IStoper::getElapsedTime ()` `[protected]`, `[pure virtual]`

Ma symulowac rezultat pokazania wyniku pomiaru czasu na stoperze.

Metoda czysto wirtualna.

Implemented in [Stoper](#).

5.2.2.3 `virtual void IStoper::start ()` `[protected]`, `[pure virtual]`

Ma symulowac moment startu stopera.

Metoda czysto wirtualna.

Implemented in [Stoper](#).

5.2.2.4 `virtual void IStoper::stop ()` `[protected]`, `[pure virtual]`

Implemented in [Stoper](#).

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

- `/home/kkuczaj/PAMSI/lab02_07.03/prj/inc/IStoper.h`

5.3 Sedzia Class Reference

Implementacja klasy [Sedzia](#).

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

Public Member Functions

- `bool setOffsetTable (int how_many)`
Funkcja, w ktorej odbywa sie bieg.

5.3.1 Detailed Description

Implementacja klasy [Sedzia](#).

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

5.3.2 Member Function Documentation

5.3.2.1 bool Sedzia::setOffTable (int *how_many*)

Funkcja, w której odbywa się bieg.

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

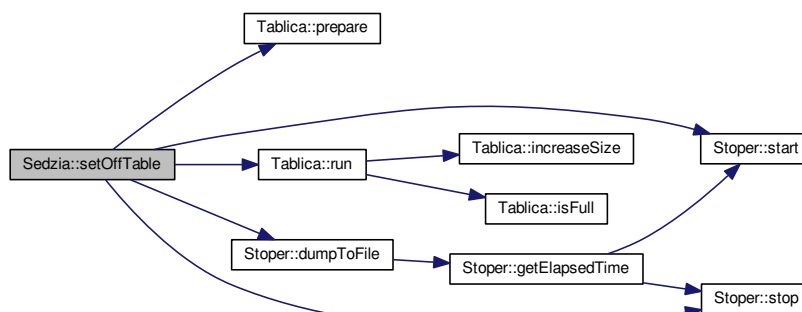
Parameters

<i>how_many</i>	Informacja iloma elementami ma zostać wypełniona tablica.
-----------------	---

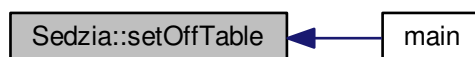
Return values

<i>true</i>	Wycig zakończony.
-------------	-------------------

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:

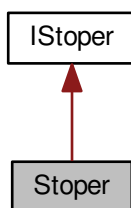
- [/home/kkuczaj/PAMSI/lab02_07.03/prj/inc/Sedzia.h](#)
- [/home/kkuczaj/PAMSI/lab02_07.03/prj/src/Sedzia.cpp](#)

5.4 Stoper Class Reference

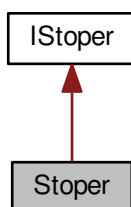
Implementacja klasy [Stoper](#).

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


Inheritance diagram for Stoper:



Collaboration diagram for Stoper:



Public Member Functions

- `Stoper ()`
Konstruktor bezparametryczny.
- `~Stoper ()`
Destruktor.
- `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 (std::string file_name)`
Implementacja funkcji `dumpToFile()` z interfejsu `IStoper`.

Private Attributes

- `timeval * start_time`
Moment startu stopera.
- `timeval * stop_time`
Moment zatrzymania stopera.

Additional Inherited Members

5.4.1 Detailed Description

Implementacja klasy [Stoper](#).

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

5.4.2 Constructor & Destructor Documentation

5.4.2.1 `Stoper::Stoper ()`

Konstruktor bezparametryczny.

Alokuje pamięć dla pól, ponieważ są wskaźnikami i będą przechowywać wartości.

5.4.2.2 `Stoper::~~Stoper ()`

Destruktor.

Zwalnia pamięć po polach.

5.4.3 Member Function Documentation

5.4.3.1 `void Stoper::dumpToFile (std::string file_name) [virtual]`

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

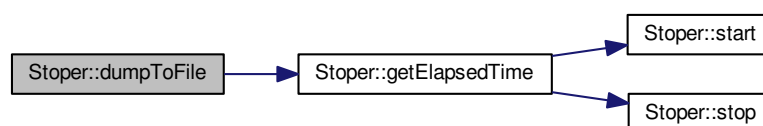
Zapisuje zmierzony czas do pliku o nazwie `"${file_name}.csv"`. Plik otwierany w trybie dopisywania (append) oraz wyjściowym (out). Plik .csv to tzw. Comma-Separated Values - łatwo je potem zaimportować do arkusza kalkulacyjnego oraz są zgodne z ogólnie przyjętym standardem.

Parameters

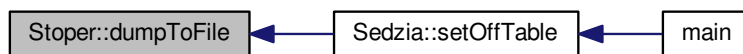
<i>file_name</i>	Nazwa pliku, do którego będą zapisane dane. Nazwa nie powinna zawierać rozszerzenia. Rozszerzenie jest dodawane w funkcji.
------------------	--

Implements [IStoper](#).

Here is the call graph for this function:



Here is the caller graph for this function:



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

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

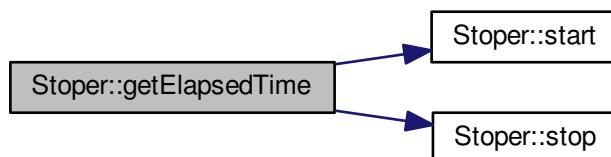
Oblicza czas pomiędzy czasem zapisanym w zmiennych `start_time` i `stop_time`.

Returns

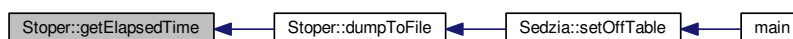
Zwraca zmierzony czas - różnica pomiędzy polem `start_time` a polem `stop_time`. Zwraca wynik w mikrosekundach.

Implements [IStoper](#).

Here is the call graph for this function:



Here is the caller graph for this function:



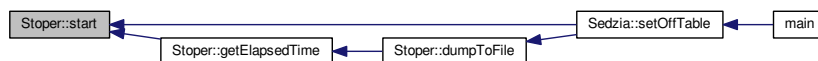
5.4.3.3 `void Stoper::start () [virtual]`

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

Zapisuje moment uruchomienia stopera. Korzysta z metody `gettimeofday()`.

Implements [IStoper](#).

Here is the caller graph for this function:



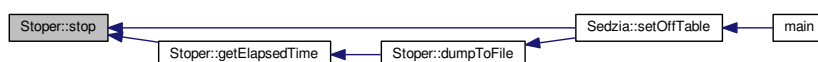
5.4.3.4 void Stoper::stop () [virtual]

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

Zapisuje moment zatrzymania stopera. Korzysta z metody gettimeofday().

Implements [IStoper](#).

Here is the caller graph for this function:



5.4.4 Member Data Documentation

5.4.4.1 timeval* Stoper::start_time [private]

Moment startu stopera.

Element przechowujący informacje o czasie systemowym w momencie uruchomienia stopera. Element timeval. Nazwa zgodna konwencja podręcznika "Google C++ Style Guide".

5.4.4.2 timeval* Stoper::stop_time [private]

Moment zatrzymania stopera.

Element przechowujący informacje o czasie systemowym w momencie zatrzymania stopera. Element typu timeval. Nazwa zgodna konwencja podręcznika "Google C++ Style Guide".

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

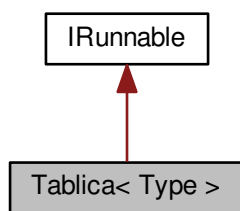
- [/home/kkuczaj/PAMSI/lab02_07.03/prj/inc/Stoper.h](#)
- [/home/kkuczaj/PAMSI/lab02_07.03/prj/src/Stoper.cpp](#)

5.5 Tablica< Type > Class Template Reference

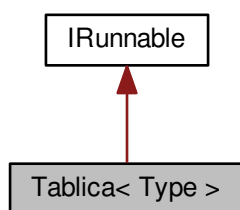
Klasa [Tablica](#), w której odbywa się zapis dynamiczny elementów.

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

Inheritance diagram for Tablica< Type >:



Collaboration diagram for Tablica< Type >:



Public Member Functions

- `Tablica (uint x=10)`
Konstruktor parametryczny.
- `~Tablica ()`
Destruktor.
- `virtual void prepare (uint size)`
Implementacja funkcji `prepare()` interfesju `IRunnable`.
- `virtual void run ()`
Implementacja funkcji `run()` interfesju `IRunnable`.
- `uint 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

- Type * [elements](#)
Wskaznik do poczatku tablicy dynamicznej.
- [uint current_size](#)
Okresla aktualny rozmiar stosu.
- [uint desired_size](#)
Okresla pozadany rozmiar stosu.
- unsigned int [index](#)
Okresla aktualny indeks.

5.5.1 Detailed Description

`template<class Type>class Tablica< Type >`

Klasa [Tablica](#), w której odbywa się zapis dynamiczny elementów.

Implementuje metody interfejsu [IRunnable](#). Zajmuje się dynamiczną alokacją pamięci. Elastyczna a propos typów wskutek zastosowania szablonów.

5.5.2 Constructor & Destructor Documentation

5.5.2.1 `template<class Type> Tablica< Type >::Tablica (uint x = 10) [inline],[explicit]`

Konstruktor parametryczny.

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

Parameters

x	Okresla początkowa wielkość przydzielonej pamięci. Domyślna wartość w przypadku braku podania to 10.
---	--

5.5.2.2 `template<class Type> Tablica< Type >::~~Tablica () [inline]`

Destruktor.

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

5.5.3 Member Function Documentation

5.5.3.1 `template<class Type> uint Tablica< Type >::getSize () [inline]`

Zwraca aktualny rozmiar tablicy dynamicznej.

Zwraca wartość pola `current_size`.

Returns

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

5.5.3.2 `template<class Type> void Tablica< Type >::increaseSize () [inline],[private]`

Zwiększa rozmiar przydzielonej pamięci na sterce.

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

Here is the caller graph for this function:



5.5.3.3 `template<class Type> bool Tablica< Type >::isFull () [inline],[private]`

Pozwala prosto określić, czy należy przydzielić pamięć.

Metoda prywatna. Służy do określania czy należy wywołać metodę [increaseSize\(\)](#).

Returns

`true` Pamięć pełna. Należy zwiększyć rozmiar.

`false` Jest jeszcze wolne miejsce.

Here is the caller graph for this function:



5.5.3.4 `template<class Type> virtual void Tablica< Type >::prepare (uint size) [inline],[virtual]`

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

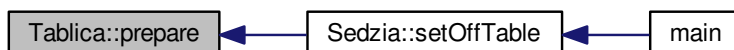
Zapisuje pożądaną rozmiar do pola `desired_size`.

Parameters

<i>size</i>	Parametr typu unsigned int, gdyż rozmiar nie powinien nigdy być ujemny. Jego wartość zapisywana jest do pola <code>desired_size</code> .
-------------	--

Implements [IRunnable](#).

Here is the caller graph for this function:



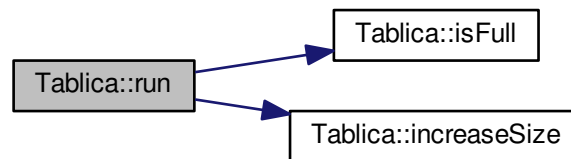
5.5.3.5 `template<class Type> virtual void Tablica< Type >::run () [inline],[virtual]`

Implementacja funkcji `run()` interfejsu `IRunnable`.

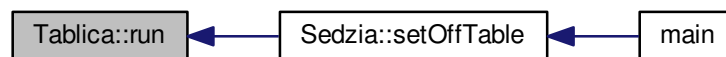
Uruchamia "bieg", w którym następuje zapis elementów do poszczególnych elementów tablicy dynamicznej. Tam odbywa się alokacja pamięci oraz instrukcje warunkowe.

Implements `IRunnable`.

Here is the call graph for this function:



Here is the caller graph for this function:



5.5.4 Member Data Documentation

5.5.4.1 `template<class Type> uint Tablica< Type >::current_size [private]`

Okresla aktualny rozmiar stosu.

Pole prywatne typu unsigned int, gdyż rozmiar nigdy nie powinien być ujemny.

5.5.4.2 `template<class Type> uint Tablica< Type >::desired_size [private]`

Okresla pozadany rozmiar stosu.

Pole prywatne typu unsigned int, gdyż rozmiar nigdy nie powinien być ujemny. Zadawane w funkcji `prepare()`.

5.5.4.3 `template<class Type> Type* Tablica< Type >::elements [private]`

Wskaźnik do początku tablicy dynamicznej.

Wskazuje na adres w pamięci serty. Pole prywatne.

5.5.4.4 `template<class Type> unsigned int Tablica< Type >::index` [private]

Okresla aktualny indeks.

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

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

- /home/kkuczaj/PAMSI/lab02_07.03/prj/inc/Tablica.h

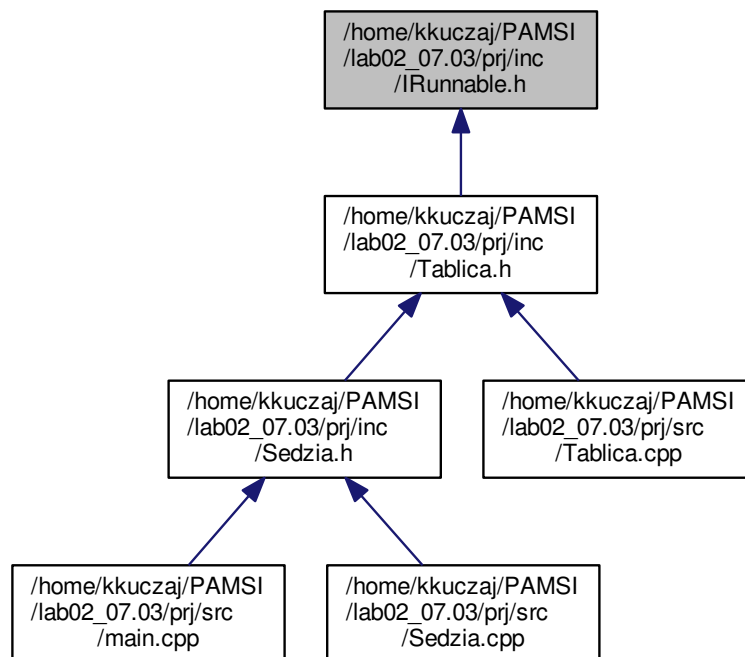
Chapter 6

File Documentation

6.1 /home/kkuczaj/PAMSI/lab02_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.

6.1.1 Detailed Description

Naglowek zawierajacy interfejs dla biegacza.

Author

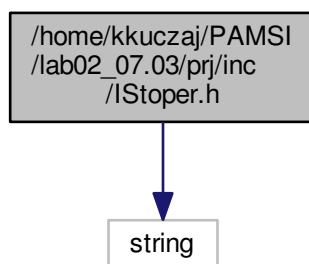
Kamil Kuczaj

6.2 /home/kkuczaj/PAMSI/lab02_07.03/prj/inc/IStoper.h File Reference

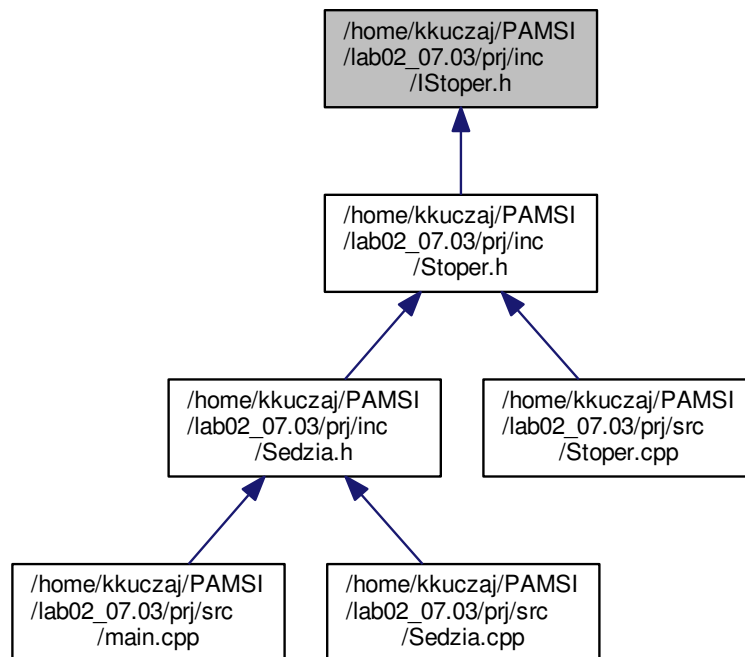
Naglowek zawierajacy interfejs dla stopera.

```
#include <string>
```

Include dependency graph for IStoper.h:



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



Classes

- class [IStoper](#)

Interfejs dla stopera.

6.2.1 Detailed Description

Nagłówek zawierający interfejs dla stopera.

Author

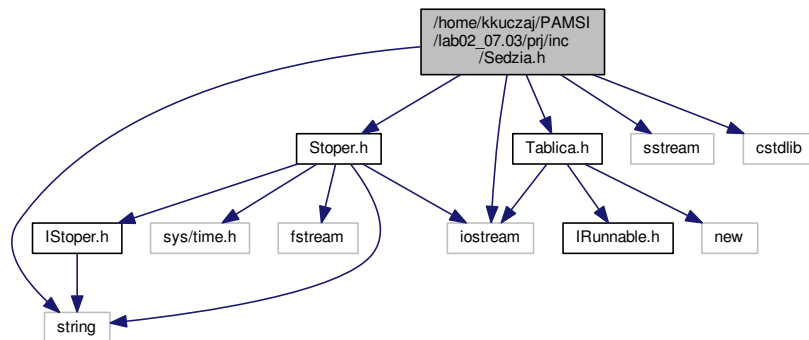
Kamil Kuczaj

6.3 /home/kkuczaj/PAMSI/lab02_07.03/prj/inc/Sedzia.h File Reference

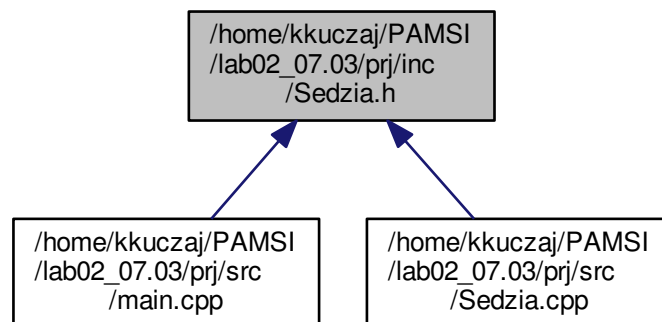
Nagłówek opisujący implementację Sedziego.

```
#include "Stoper.h"
#include "Tablica.h"
#include <iostream>
#include <sstream>
#include <string>
#include <cstdlib>
```

Include dependency graph for Sedzia.h:



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



Classes

- class [Sedzia](#)
Implementacja klasy [Sedzia](#).

6.3.1 Detailed Description

Nagłówek opisujący implementację Sedziego.

Author

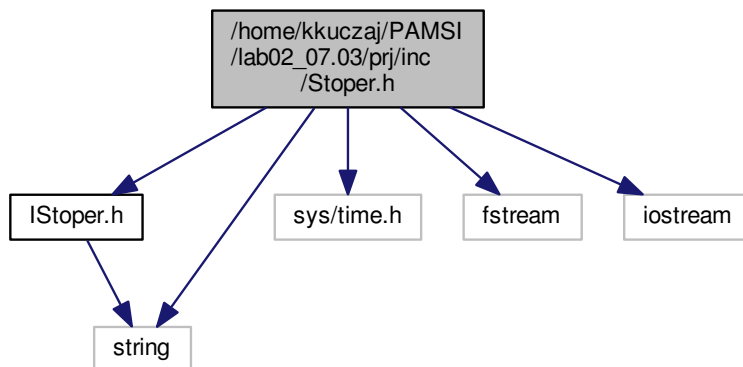
Kamil Kuczaj

6.4 /home/kkuczaj/PAMSI/lab02_07.03/prj/inc/Stoper.h File Reference

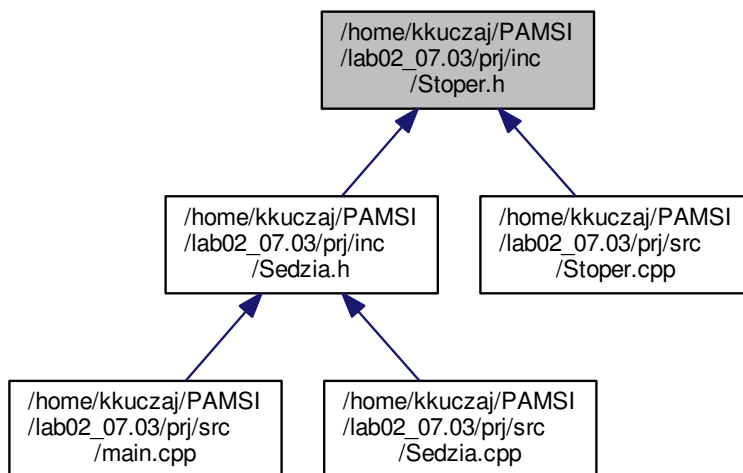
Implementacja interfejsu [IStoper](#) w klasie [Stoper](#).

```
#include "IStoper.h"
#include <sys/time.h>
#include <fstream>
#include <iostream>
#include <string>
```

Include dependency graph for Stoper.h:



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



Classes

- class [Stoper](#)

Implementacja klasy [Stoper](#).

6.4.1 Detailed Description

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

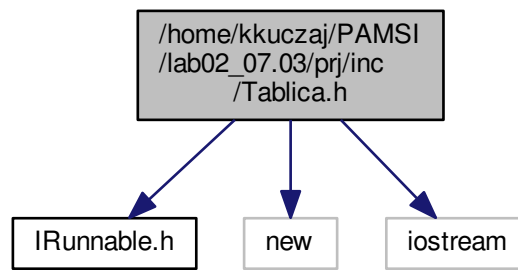
Author

Kamil Kuczaj

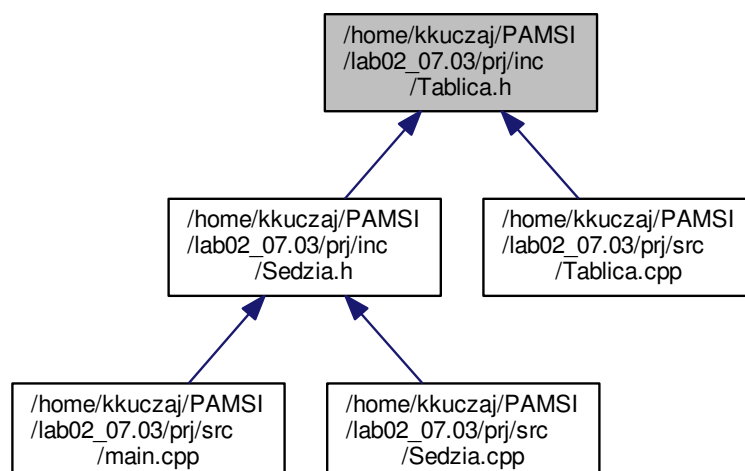
6.5 `/home/kkuczaj/PAMSI/lab02_07.03/prj/inc/Tablica.h` File Reference

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< Type >](#)
Klasa [Tablica](#), w której odbywa się zapis dynamiczny elementów.

Typedefs

- typedef unsigned int [uint](#)
Skraca zapis.

6.5.1 Detailed Description

Implementacja interfejsu [IRunnable](#).

Author

Kamil Kuczaj

6.5.2 Typedef Documentation

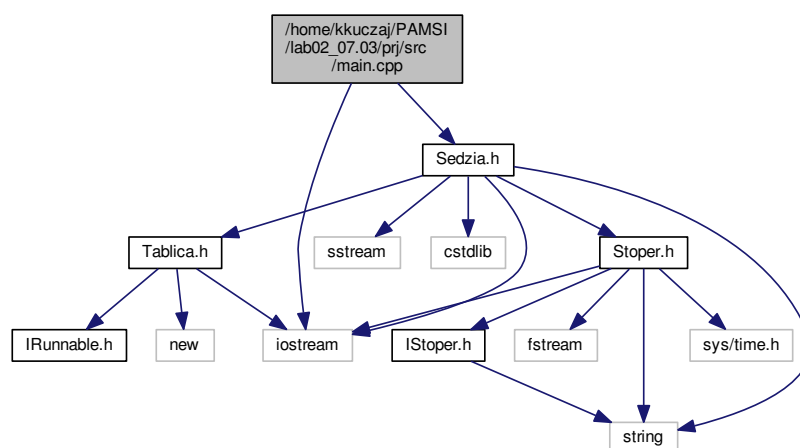
6.5.2.1 typedef unsigned int uint

Skraca zapis.

Zdefiniowanie własnego typu - pozwala na krótszy zapis

6.6 /home/kkuczaj/PAMSI/lab02_07.03/prj/src/main.cpp File Reference

```
#include "Sedzia.h"
#include <iostream>
Include dependency graph for main.cpp:
```



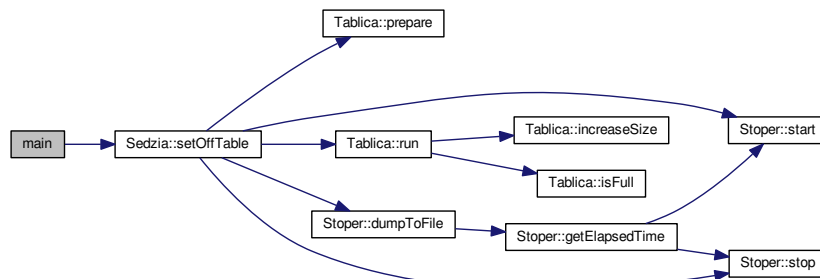
Functions

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

6.6.1 Function Documentation

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

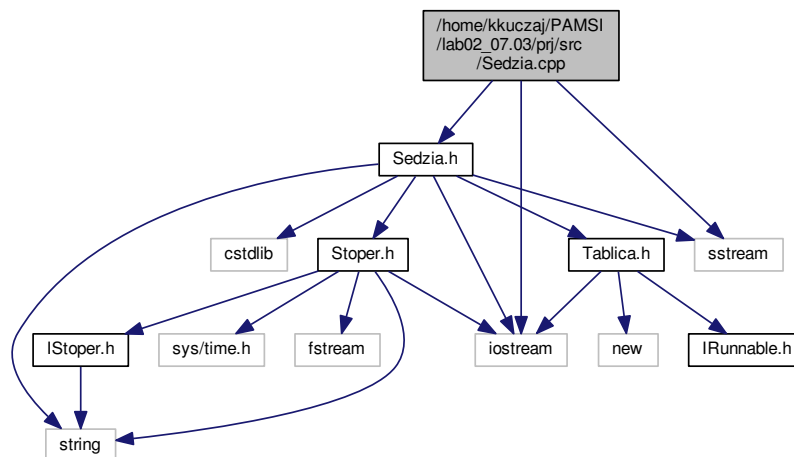
Here is the call graph for this function:



6.7 /home/kkuczaj/PAMSI/lab02_07.03/prj/src/Sedzia.cpp File Reference

```
#include "Sedzia.h"
#include <iostream>
#include <sstream>
```

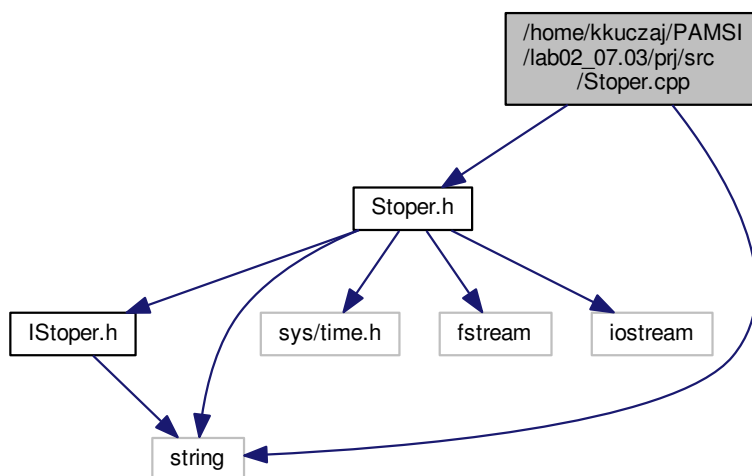
Include dependency graph for Sedzia.cpp:



6.8 /home/kkuczaj/PAMSI/lab02_07.03/prj/src/Stoper.cpp File Reference

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

Include dependency graph for Stoper.cpp:



6.9 /home/kkuczaj/PAMSI/lab02_07.03/prj/src/Tablica.cpp File Reference

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

Include dependency graph for Tablica.cpp:

