

Listazad

Generated by Doxygen 1.8.6

Sun Mar 20 2016 13:00:10

Contents

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

ILista	??
Lista	??
BiegaczLista	??
IRunnable	??
Biegacz	??
BiegaczLista	??
IStoper	??
Stoper	??
Node	??
Sedzia	??

Chapter 2

Class Index

2.1 Class List

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

Biegacz	??
BiegaczLista		
	Przygotowuje liste i definiuje przebieg jej uzupełniania	??
ILista	??
IRunnable	??
IStoper	??
Lista		
	Deklaruje funkcje listy	??
Node		
	Pojedynczy element	??
Sedzia		
	Jest to klasa nadrzedna , ktora struje klasami Stoper i Biegacz poprzez odwołanie sie do interfejsow	??
Stoper		
	Zastepuje kilkukrotne wklejanie pliku naglowkowego	??

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

/home/patr95/Pulpit/PAMSI/1403/prog/prj/build/CMakeFiles/2.8.12.2/CompilerIdC/CMakeCCompilerId.c	??
/home/patr95/Pulpit/PAMSI/1403/prog/prj/build/CMakeFiles/2.8.12.2/CompilerIdCXX/CMakeCXXCompilerId.cpp	??
/home/patr95/Pulpit/PAMSI/1403/prog/prj/CMakeFiles/2.8.12.2/CompilerIdC/CMakeCCompilerId.c	??
/home/patr95/Pulpit/PAMSI/1403/prog/prj/CMakeFiles/2.8.12.2/CompilerIdCXX/CMakeCXXCompilerId.cpp	??
/home/patr95/Pulpit/PAMSI/1403/prog/prj/inc/Biegacz.h	??
/home/patr95/Pulpit/PAMSI/1403/prog/prj/inc/BiegaczLista.h	??
/home/patr95/Pulpit/PAMSI/1403/prog/prj/inc/ILista.h	??
/home/patr95/Pulpit/PAMSI/1403/prog/prj/inc/IRunnable.h	??
/home/patr95/Pulpit/PAMSI/1403/prog/prj/inc/IStoper.h	??
/home/patr95/Pulpit/PAMSI/1403/prog/prj/inc/Lista.h	??
/home/patr95/Pulpit/PAMSI/1403/prog/prj/inc/Sedzia.h	??
/home/patr95/Pulpit/PAMSI/1403/prog/prj/inc/Stoper.h	??
/home/patr95/Pulpit/PAMSI/1403/prog/prj/src/Biegacz.cpp	??
/home/patr95/Pulpit/PAMSI/1403/prog/prj/src/BiegaczLista.cpp	??
/home/patr95/Pulpit/PAMSI/1403/prog/prj/src/Lista.cpp	??
/home/patr95/Pulpit/PAMSI/1403/prog/prj/src/main.cpp	??
/home/patr95/Pulpit/PAMSI/1403/prog/prj/src/Sedzia.cpp	??
/home/patr95/Pulpit/PAMSI/1403/prog/prj/src/Stoper.cpp	??

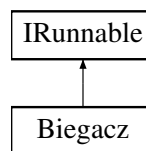
Chapter 4

Class Documentation

4.1 Biegacz Class Reference

```
#include <Biegacz.h>
```

Inheritance diagram for Biegacz:



Public Member Functions

- [Biegacz](#) ()
- virtual [~Biegacz](#) ()
- bool [prepare](#) (int size)
przygotowanie biegacza, czyli oczyszczenie z tablicy wynikow po wcześniejszym biegu i przygotowanie nowego dystansu.
- bool [run](#) ()

4.1.1 Detailed Description

Definition at line 7 of file Biegacz.h.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 Biegacz::Biegacz ()

stworzenie tablicy do powiekszenia

Definition at line 3 of file Biegacz.cpp.

4.1.2.2 Biegacz::~Biegacz () [virtual]

Definition at line 10 of file Biegacz.cpp.

4.1.3 Member Function Documentation

4.1.3.1 `bool Biegacz::prepare (int size) [virtual]`

Sprzątam po poprzednim "biegu", ustawiam wartosci poczatkowe

oczekiwana wart: 10,10³,10⁵...

Implements [IRunnable](#).

Definition at line 50 of file Biegacz.cpp.

4.1.3.2 `bool Biegacz::run () [virtual]`

uzupelnianie tablicy zerami.

Implements [IRunnable](#).

Definition at line 61 of file Biegacz.cpp.

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

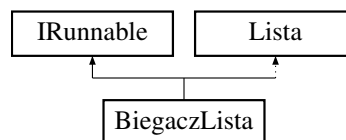
- </home/patr95/Pulpit/PAMSI/1403/prog/prj/inc/Biegacz.h>
- </home/patr95/Pulpit/PAMSI/1403/prog/prj/src/Biegacz.cpp>

4.2 BiegaczLista Class Reference

Przygotowuje liste i definiuje przebieg jej uzupelniania.

```
#include <BiegaczLista.h>
```

Inheritance diagram for BiegaczLista:



Public Member Functions

- `bool prepare (int rozmiar_docelowy)`
- `bool run ()`

4.2.1 Detailed Description

Definition at line 10 of file BiegaczLista.h.

4.2.2 Member Function Documentation

4.2.2.1 `bool BiegaczLista::prepare (int rozmiar_docelowy) [virtual]`

Klasa pomocnicza majaca na celu zdefiniowanie metod z tablicowego Biegacza, dla listy. Usuwa wczesniej modyfikowana liste. Funkcja prepare zwiksza rozmiar listy do rozmiaru docelowego, wypelnia ja losowymi slowami, oraz jednym slowem ktorego bedziemy szukac. sprzatanie po poprzedniej liscie.

wypelnienie listy losowymi slowami/elementami(stale)

wrzucanie wyrazu na losowe miejsce w liscie

Implements [IRunnable](#).

Definition at line 3 of file BiegaczLista.cpp.

4.2.2.2 bool BiegaczLista::run () [virtual]

Funkcja run wyszukuje slowo i zwraca prawde lub falsz w zaleznosci ,czy znalazlo czy nie. przeszukuje liste by znalezc slowo podane nizej.

Implements [IRunnable](#).

Definition at line 21 of file BiegaczLista.cpp.

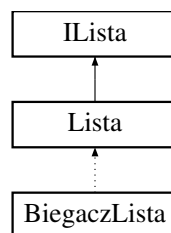
The documentation for this class was generated from the following files:

- /home/patr95/Pulpit/PAMSI/1403/prog/prj/inc/[BiegaczLista.h](#)
- /home/patr95/Pulpit/PAMSI/1403/prog/prj/src/[BiegaczLista.cpp](#)

4.3 ILista Class Reference

```
#include <ILista.h>
```

Inheritance diagram for ILista:



Public Member Functions

- virtual void [Add](#) (std::string item, int index)=0
- virtual void [Remove](#) (int index)=0
- virtual bool [IsEmpty](#) ()=0
- virtual std::string [Get](#) (int index)=0
- virtual int [Size](#) ()=0

4.3.1 Detailed Description

Definition at line 5 of file ILista.h.

4.3.2 Member Function Documentation

4.3.2.1 virtual void ILista::Add (std::string *item*, int *index*) [pure virtual]

Implemented in [Lista](#).

4.3.2.2 virtual std::string ILista::Get (int *index*) [pure virtual]

Implemented in [Lista](#).

4.3.2.3 `virtual bool ILista::IsEmpty () [pure virtual]`

Implemented in [Lista](#).

4.3.2.4 `virtual void ILista::Remove (int index) [pure virtual]`

Implemented in [Lista](#).

4.3.2.5 `virtual int ILista::Size () [pure virtual]`

Implemented in [Lista](#).

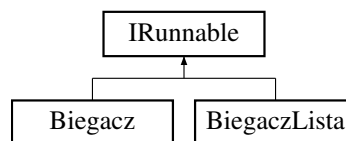
The documentation for this class was generated from the following file:

- </home/patr95/Pulpit/PAMSI/1403/prog/prj/inc/ILista.h>

4.4 IRunnable Class Reference

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

Inheritance diagram for IRunnable:



Public Member Functions

- `virtual bool prepare (int size)=0`
- `virtual bool run ()=0`

4.4.1 Detailed Description

Definition at line 3 of file `IRunnable.h`.

4.4.2 Member Function Documentation

4.4.2.1 `virtual bool IRunnable::prepare (int size) [pure virtual]`

Implemented in [Biegacz](#), and [BiegaczLista](#).

4.4.2.2 `virtual bool IRunnable::run () [pure virtual]`

Implemented in [Biegacz](#), and [BiegaczLista](#).

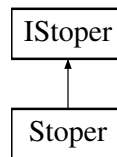
The documentation for this class was generated from the following file:

- </home/patr95/Pulpit/PAMSI/1403/prog/prj/inc/IRunnable.h>

4.5 IStoper Class Reference

```
#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](#) (std::string const &)=0

4.5.1 Detailed Description

Definition at line 5 of file IStoper.h.

4.5.2 Member Function Documentation

4.5.2.1 virtual void IStoper::dumpToFile (std::string const &) [pure virtual]

Implemented in [Stoper](#).

4.5.2.2 virtual double IStoper::getElapsedTime () [pure virtual]

Implemented in [Stoper](#).

4.5.2.3 virtual void IStoper::Start () [pure virtual]

Implemented in [Stoper](#).

4.5.2.4 virtual void IStoper::Stop () [pure virtual]

Implemented in [Stoper](#).

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

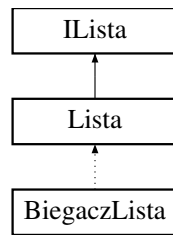
- /home/patr95/Pulpit/PAMSI/1403/prog/prj/inc/[IStoper.h](#)

4.6 Lista Class Reference

Deklaruje funkcje listy.

```
#include <Lista.h>
```

Inheritance diagram for Lista:



Public Member Functions

- void [Add](#) (std::string item, int index)
- void [Remove](#) (int index)
- bool [IsEmpty](#) ()
- std::string [Get](#) (int index)
- int [Size](#) ()
- [Lista](#) ()

4.6.1 Detailed Description

Definition at line 17 of file Lista.h.

4.6.2 Constructor & Destructor Documentation

4.6.2.1 `Lista::Lista ()` `[inline]`

Definition at line 62 of file Lista.h.

4.6.3 Member Function Documentation

4.6.3.1 `void Lista::Add (std::string item, int index)` `[virtual]`

Funkcja Add dodaje kolejny element listy. Jesli rozmiar listy jest za maly w stosunku do podawanego indexu to wyswietla komunikat. Funkcja dziala w taki sposob, ze tworzy nowy "Node" do tablicy, wskaznik z wczesniejszego elementu wskazuje na nowy [Node](#) nastepnie z [Node](#) wskazujemy na kolejny element tablicy, aby byla ciaglosc w liscie. W tej funkcji rozpatrujemy 3 przypadki zwiekszenia listy, dostawianie elementu na pocz.koniec lub gdzieś pomiedzy w liscie. Jesli dostawiamy na poczatek tworzymy nowy element i wskazujemy na niego wskaznikiem poczatkowym i koncowym. Jesli tworzymy ostatni element, to dodajemy nowego Noda, i wskazujemy na niego wskaznikiem KONcowym. Ostatnim przypadkiem jest elementu gdzieś pomiedzy POCZ i KON listy. Tutaj musimy pamietac ze lista liczy swoje miejsce od 0, wiec w petli for mamy index-1. Tworzymy tymczasowy wskaznik i ustawiamy go na pierwszy element. nastepnie w petli przesuwamy o tyle elementów ile zadamy-1. Przesuwamy ten wskaznik na kolejny element, tworzymy pomocniczy wskaznik ktory ma wskazywac na nowo utworzony element. W nowo utworzonym elemencie dodajemy wskaznik ktory wskazuje na nastepny element ktory byl stworzony przed naszym nowym Nodem. Wszystko po to by zachowac ciaglosc listy i wskazywanie na kolejne elementy. dostawiam element na poczatku i wskazuje na wczesniejszy element

aktualizowanie poczatkowego wskaznika na nowo dodany element

aktualizowanie koncowego+ wskaznika na nowo dodany element

wskaznik koncowy wskazuje na podpiete nowe pole w liscie

temp to tymczasowy wskaznik dodawanej listy

przestawiamy sie na kolejny element

pomocniczy wskaznik

wskazuje na nowo stworzony element

w nowo dodanym elemencie ustawiamy wskaźnik na wcześniejszy (który jest następnym) elementem.

Implements [ILista](#).

Definition at line 12 of file Lista.cpp.

4.6.3.2 `std::string Lista::Get (int index) [virtual]`

W tej funkcji tworzymy wskaźnik tymczasowy na 1 element listy, przesuwamy o `index-1`, po znalezieniu interesującego nas elementu zwracamy jego wartość. `temp` to tymczasowy wskaźnik dodawanej listy

Implements [ILista](#).

Definition at line 70 of file Lista.cpp.

4.6.3.3 `bool Lista::IsEmpty () [virtual]`

Zwraca `true` jeśli lista jest pusta lub `false` jeśli nie jest.

Implements [ILista](#).

Definition at line 8 of file Lista.cpp.

4.6.3.4 `void Lista::Remove (int index) [virtual]`

`Remove` działa w podobny sposób. Jeśli zadamy usunięcie 1 elementu, to zapamiętujemy coś na co wskazuje 1 element. Usuwaemy to, by następnie ustawić wskaźnik początkowy na coś, na co wskazywał pierwszy element. Jeśli usuwamy ostatni element to tworzymy wskaźnik tymczasowy na pierwszy element, przesuwamy do ostatniego, gdzie `index` zmniejszamy `-2`, ponieważ indeksujemy listę od zera, oraz interesuje nas wcześniejszy element który wskazuje na ostatni. Gdy osiągniemy przedostatni element wskazujemy jego wskaźnikiem na null pointer oraz usuwamy wcześniejszy ostatni element z listy na którego nic nie wskazuje. W ostatnim przypadku tworzymy tymczasowy następnik, który jest wskaźnikiem na wskaźnik. Przeszukujemy listę od początku, usuwamy element i nasz wskaźnik dalej wskazuje na kolejny element dzięki następnikowi. Zachowana jest ciągłość wskazywania na siebie elementów z listy. zapamiętaj drugi element listy

usun pierwszy element

przestaw drugi na pierwszy

`temp` to tymczasowy wskaźnik dodawanej listy

przesuwamy się na kolejny element

`temp` to tymczasowy wskaźnik dodawanej listy

Implements [ILista](#).

Definition at line 43 of file Lista.cpp.

4.6.3.5 `int Lista::Size () [virtual]`

`Size` zwraca nam zmienną rozmiar.

Implements [ILista](#).

Definition at line 3 of file Lista.cpp.

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

- `/home/patr95/Pulpit/PAMSI/1403/prog/prj/inc/Lista.h`
- `/home/patr95/Pulpit/PAMSI/1403/prog/prj/src/Lista.cpp`

4.7 Node Struct Reference

pojedynczy element

```
#include <Lista.h>
```

Public Attributes

- [Node](#) * [wskaznik](#)
- std::string [wartosc](#)

4.7.1 Detailed Description

Definition at line 8 of file [Lista.h](#).

4.7.2 Member Data Documentation

4.7.2.1 std::string Node::wartosc

Definition at line 11 of file [Lista.h](#).

4.7.2.2 Node* Node::wskaznik

Definition at line 10 of file [Lista.h](#).

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

- [/home/patr95/Pulpit/PAMSI/1403/prog/prj/inc/Lista.h](#)

4.8 Sedzia Class Reference

Jest to klasa nadrzedna , ktora struje klasami [Stoper](#) i [Biegacz](#) poprzez odwołanie sie do interfejsow.

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

Public Member Functions

- [Sedzia](#) ([IRunnable](#) &biegacz, [IStoper](#) &stoper)
- void [release](#) ()

4.8.1 Detailed Description

Definition at line 16 of file [Sedzia.h](#).

4.8.2 Constructor & Destructor Documentation

4.8.2.1 Sedzia::Sedzia ([IRunnable](#) & *biegacz*, [IStoper](#) & *stoper*) [\[inline\]](#)

Definition at line 21 of file [Sedzia.h](#).

4.8.3 Member Function Documentation

4.8.3.1 void Sedzia::release ()

Przez funkcje release wywołujemy kolejno funkcje biegacz prepare->Start->run->Stop->getElapsedTime->dumpToFile ktore sa zdefiniowane w klasach [Stoper](#) i [Biegacz](#). Mamy taka mozliwosc poprzez interfejsy.

Definition at line 3 of file Sedzia.cpp.

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

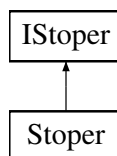
- /home/patr95/Pulpit/PAMSI/1403/prog/prj/inc/[Sedzia.h](#)
- /home/patr95/Pulpit/PAMSI/1403/prog/prj/src/[Sedzia.cpp](#)

4.9 Stoper Class Reference

zastępuje kilkakrotne wklejanie pliku naglowkowego

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

Inheritance diagram for Stoper:



Public Types

- using [clock](#) = std::chrono::high_resolution_clock
const& =unikamy kopi
- using [time_point](#) = clock::time_point
czas systemowy
- using [time_type](#) = std::chrono::milliseconds

Public Member Functions

- void [Start](#) ()
- void [Stop](#) ()
- double [getElapsedTime](#) ()
duration cast=operator porownujacy, pozwala odjac 2 punkty w czasie
- void [dumpToFile](#) (std::string const &)

4.9.1 Detailed Description

Mierzy czas, wyswietla czas okrazenia, zwraca wyniki do pliku.

Definition at line 13 of file Stoper.h.

4.9.2 Member Typedef Documentation

4.9.2.1 using Stoper::clock = std::chrono::high_resolution_clock

Definition at line 34 of file Stoper.h.

4.9.2.2 using Stoper::time_point = clock::time_point

Definition at line 35 of file Stoper.h.

4.9.2.3 using Stoper::time_type = std::chrono::milliseconds

Definition at line 36 of file Stoper.h.

4.9.3 Member Function Documentation

4.9.3.1 void Stoper::dumpToFile (std::string const & *nazwaPliku*) [virtual]

dumpToFile wysyla wyniki do pliku tekstowego. Jesli go nie am to go tworzy. Wyniki wysylane sa w kolejnosci wykonania petli.

Implements [IStoper](#).

Definition at line 19 of file Stoper.cpp.

4.9.3.2 double Stoper::getElapsedTime () [virtual]

getElapsedTime wykonuje roznice (Stop-Start) i zwraca czas w milisekundach

Implements [IStoper](#).

Definition at line 13 of file Stoper.cpp.

4.9.3.3 void Stoper::Start () [virtual]

Start pobiera czas systemowy przed wykonaniem zadania

Implements [IStoper](#).

Definition at line 3 of file Stoper.cpp.

4.9.3.4 void Stoper::Stop () [virtual]

Stop pobiera czas systemowy po wykonaniu zadania

Implements [IStoper](#).

Definition at line 8 of file Stoper.cpp.

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

- [/home/patr95/Pulpit/PAMSI/1403/prog/prj/inc/Stoper.h](#)
- [/home/patr95/Pulpit/PAMSI/1403/prog/prj/src/Stoper.cpp](#)

Chapter 5

File Documentation

5.1 /home/patr95/Pulpit/PAMSI/1403/prog/prj/build/CMakeFiles/2.8.12.2/CompilerIdC/C-MakeCCompilerId.c File Reference

Macros

- `#define COMPILER_ID ""`
- `#define PLATFORM_ID ""`
- `#define ARCHITECTURE_ID ""`
- `#define DEC(n)`
- `#define HEX(n)`

Functions

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

Variables

- `char const * info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"`
- `char const * info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"`
- `char const * info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"`

5.1.1 Macro Definition Documentation

5.1.1.1 `#define ARCHITECTURE_ID ""`

Definition at line 320 of file CMakeCCompilerId.c.

5.1.1.2 `#define COMPILER_ID ""`

Definition at line 200 of file CMakeCCompilerId.c.

5.1.1.3 `#define DEC(n)`

Value:

```
( '0' + ((n) / 10000000) % 10 ), \
( '0' + ((n) / 1000000) % 10 ), \
( '0' + ((n) / 100000) % 10 ), \
( '0' + ((n) / 10000) % 10 ), \
( '0' + ((n) / 1000) % 10 ), \
( '0' + ((n) / 100) % 10 ), \
( '0' + ((n) / 10) % 10 ), \
( '0' + ((n) % 10) )
```

Definition at line 324 of file CMakeCCompilerId.c.

5.1.1.4 #define HEX(n)

Value:

```
( '0' + ((n) >> 28 & 0xF) ), \
( '0' + ((n) >> 24 & 0xF) ), \
( '0' + ((n) >> 20 & 0xF) ), \
( '0' + ((n) >> 16 & 0xF) ), \
( '0' + ((n) >> 12 & 0xF) ), \
( '0' + ((n) >> 8 & 0xF) ), \
( '0' + ((n) >> 4 & 0xF) ), \
( '0' + ((n) & 0xF) )
```

Definition at line 335 of file CMakeCCompilerId.c.

5.1.1.5 #define PLATFORM_ID ""

Definition at line 287 of file CMakeCCompilerId.c.

5.1.2 Function Documentation

5.1.2.1 int main (int argc, char * argv[])

Definition at line 377 of file CMakeCCompilerId.c.

5.1.3 Variable Documentation

5.1.3.1 char const* info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"

Definition at line 368 of file CMakeCCompilerId.c.

5.1.3.2 char const* info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"

Definition at line 208 of file CMakeCCompilerId.c.

5.1.3.3 char const* info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"

Definition at line 367 of file CMakeCCompilerId.c.

5.2 /home/patr95/Pulpit/PAMSI/1403/prog/prj/CMakeFiles/2.8.12.2/CompilerIdC/CMakeC-CompilerId.c File Reference

Macros

- #define [COMPILER_ID](#) ""

- `#define PLATFORM_ID ""`
- `#define ARCHITECTURE_ID ""`
- `#define DEC(n)`
- `#define HEX(n)`

Functions

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

Variables

- `char const * info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"`
- `char const * info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"`
- `char const * info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"`

5.2.1 Macro Definition Documentation

5.2.1.1 `#define ARCHITECTURE_ID ""`

Definition at line 320 of file CMakeCCompilerId.c.

5.2.1.2 `#define COMPILER_ID ""`

Definition at line 200 of file CMakeCCompilerId.c.

5.2.1.3 `#define DEC(n)`

Value:

```
('0' + ((n) / 10000000) % 10), \
('0' + ((n) / 1000000) % 10), \
('0' + ((n) / 100000) % 10), \
('0' + ((n) / 10000) % 10), \
('0' + ((n) / 1000) % 10), \
('0' + ((n) / 100) % 10), \
('0' + ((n) / 10) % 10), \
('0' + ((n) % 10))
```

Definition at line 324 of file CMakeCCompilerId.c.

5.2.1.4 `#define HEX(n)`

Value:

```
('0' + ((n) >> 28 & 0xF)), \
('0' + ((n) >> 24 & 0xF)), \
('0' + ((n) >> 20 & 0xF)), \
('0' + ((n) >> 16 & 0xF)), \
('0' + ((n) >> 12 & 0xF)), \
('0' + ((n) >> 8 & 0xF)), \
('0' + ((n) >> 4 & 0xF)), \
('0' + ((n) & 0xF))
```

Definition at line 335 of file CMakeCCompilerId.c.

5.2.1.5 `#define PLATFORM_ID ""`

Definition at line 287 of file CMakeCCompilerId.c.

5.2.2 Function Documentation

5.2.2.1 `int main (int argc, char * argv[])`

Definition at line 377 of file CMakeCCompilerId.c.

5.2.3 Variable Documentation

5.2.3.1 `char const* info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"`

Definition at line 368 of file CMakeCCompilerId.c.

5.2.3.2 `char const* info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"`

Definition at line 208 of file CMakeCCompilerId.c.

5.2.3.3 `char const* info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"`

Definition at line 367 of file CMakeCCompilerId.c.

5.3 /home/patr95/Pulpit/PAMSI/1403/prog/prj/build/CMakeFiles/2.8.12.2/CompilerIdCXX/-CMakeCXXCompilerId.cpp File Reference

Macros

- `#define COMPILER_ID ""`
- `#define PLATFORM_ID ""`
- `#define ARCHITECTURE_ID ""`
- `#define DEC(n)`
- `#define HEX(n)`

Functions

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

Variables

- `char const * info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"`
- `char const * info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"`
- `char const * info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"`

5.3.1 Macro Definition Documentation

5.3.1.1 `#define ARCHITECTURE_ID ""`

Definition at line 313 of file CMakeCXXCompilerId.cpp.

5.3.1.2 `#define COMPILER_ID ""`

Definition at line 193 of file CMakeCXXCompilerId.cpp.

Reference

5.3.1.3 #define DEC(n)

Value:

```
( '0' + ((n) / 10000000) % 10 ), \
( '0' + ((n) / 1000000) % 10 ), \
( '0' + ((n) / 100000) % 10 ), \
( '0' + ((n) / 10000) % 10 ), \
( '0' + ((n) / 1000) % 10 ), \
( '0' + ((n) / 100) % 10 ), \
( '0' + ((n) / 10) % 10 ), \
( '0' + ((n) % 10) )
```

Definition at line 317 of file CMakeCXXCompilerId.cpp.

5.3.1.4 #define HEX(n)

Value:

```
( '0' + ((n) >> 28 & 0xF) ), \
( '0' + ((n) >> 24 & 0xF) ), \
( '0' + ((n) >> 20 & 0xF) ), \
( '0' + ((n) >> 16 & 0xF) ), \
( '0' + ((n) >> 12 & 0xF) ), \
( '0' + ((n) >> 8 & 0xF) ), \
( '0' + ((n) >> 4 & 0xF) ), \
( '0' + ((n) & 0xF) )
```

Definition at line 328 of file CMakeCXXCompilerId.cpp.

5.3.1.5 #define PLATFORM_ID ""

Definition at line 280 of file CMakeCXXCompilerId.cpp.

5.3.2 Function Documentation

5.3.2.1 int main (int argc, char * argv[])

Definition at line 367 of file CMakeCXXCompilerId.cpp.

5.3.3 Variable Documentation

5.3.3.1 char const* info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"

Definition at line 361 of file CMakeCXXCompilerId.cpp.

5.3.3.2 char const* info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"

Definition at line 201 of file CMakeCXXCompilerId.cpp.

5.3.3.3 char const* info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"

Definition at line 360 of file CMakeCXXCompilerId.cpp.

5.4 /home/patr95/Pulpit/PAMSI/1403/prog/prj/CMakeFiles/2.8.12.2/CompilerIdCXX/C-MakeCXXCompilerId.cpp File Reference

Macros

- `#define COMPILER_ID ""`
- `#define PLATFORM_ID ""`
- `#define ARCHITECTURE_ID ""`
- `#define DEC(n)`
- `#define HEX(n)`

Functions

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

Variables

- `char const * info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"`
- `char const * info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"`
- `char const * info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"`

5.4.1 Macro Definition Documentation

5.4.1.1 `#define ARCHITECTURE_ID ""`

Definition at line 313 of file CMakeCXXCompilerId.cpp.

5.4.1.2 `#define COMPILER_ID ""`

Definition at line 193 of file CMakeCXXCompilerId.cpp.

5.4.1.3 `#define DEC(n)`

Value:

```
('0' + ((n) / 10000000) % 10), \
('0' + ((n) / 1000000) % 10), \
('0' + ((n) / 100000) % 10), \
('0' + ((n) / 10000) % 10), \
('0' + ((n) / 1000) % 10), \
('0' + ((n) / 100) % 10), \
('0' + ((n) / 10) % 10), \
('0' + ((n) % 10))
```

Definition at line 317 of file CMakeCXXCompilerId.cpp.

5.4.1.4 `#define HEX(n)`

Value:

```
('0' + ((n) >> 28 & 0xF)), \
('0' + ((n) >> 24 & 0xF)), \
('0' + ((n) >> 20 & 0xF)), \
('0' + ((n) >> 16 & 0xF)), \
('0' + ((n) >> 12 & 0xF)), \
('0' + ((n) >> 8 & 0xF)), \
('0' + ((n) >> 4 & 0xF)), \
('0' + ((n) & 0xF))
```

Definition at line 328 of file CMakeCXXCompilerId.cpp.

5.4.1.5 `#define PLATFORM_ID ""`

Definition at line 280 of file CMakeCXXCompilerId.cpp.

5.4.2 Function Documentation

5.4.2.1 `int main (int argc, char * argv[])`

Definition at line 367 of file CMakeCXXCompilerId.cpp.

5.4.3 Variable Documentation

5.4.3.1 `char const* info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"`

Definition at line 361 of file CMakeCXXCompilerId.cpp.

5.4.3.2 `char const* info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"`

Definition at line 201 of file CMakeCXXCompilerId.cpp.

5.4.3.3 `char const* info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"`

Definition at line 360 of file CMakeCXXCompilerId.cpp.

5.5 /home/patr95/Pulpit/PAMSI/1403/prog/prj/inc/Biegacz.h File Reference

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

Classes

- class [Biegacz](#)

Variables

- const int [początkowy_rozmiar_tablicy](#) = 10

5.5.1 Variable Documentation

5.5.1.1 `const int początkowy_rozmiar_tablicy = 10`

Definition at line 5 of file Biegacz.h.

5.6 /home/patr95/Pulpit/PAMSI/1403/prog/prj/inc/BiegaczLista.h File Reference

```
#include "Lista.h"
#include "IRunnable.h"
#include <random>
```

Classes

- class [BiegaczLista](#)
Przygotowuje liste i definiuje przebieg jej uzupełniania.

5.7 /home/patr95/Pulpit/PAMSI/1403/prog/prj/inc/ILista.h File Reference

```
#include <string>
```

Classes

- class [ILista](#)

5.8 /home/patr95/Pulpit/PAMSI/1403/prog/prj/inc/IRunnable.h File Reference

Classes

- class [IRunnable](#)

5.9 /home/patr95/Pulpit/PAMSI/1403/prog/prj/inc/IStoper.h File Reference

```
#include <string>
```

Classes

- class [IStoper](#)

5.10 /home/patr95/Pulpit/PAMSI/1403/prog/prj/inc/Lista.h File Reference

```
#include <string>
#include <iostream>
#include "ILista.h"
```

Classes

- struct [Node](#)
pojedynczy element

- class [Lista](#)

Deklaruje funkcje listy.

5.11 /home/patr95/Pulpit/PAMSI/1403/prog/prj/inc/Sedzia.h File Reference

```
#include "IRunnable.h"
#include "IStoper.h"
#include <fstream>
#include <string>
#include <iostream>
```

Classes

- class [Sedzia](#)

Jest to klasa nadrzedna , ktora struje klasami [Stoper](#) i [Biegacz](#) poprzez odwołanie sie do interfejsow.

Variables

- const std::string [NAZWA_PLIKU_Z_ILOSCIA_ELEMENTOW](#) = "dystans.txt"

5.11.1 Variable Documentation

5.11.1.1 const std::string [NAZWA_PLIKU_Z_ILOSCIA_ELEMENTOW](#) = "dystans.txt"

Definition at line 10 of file Sedzia.h.

5.12 /home/patr95/Pulpit/PAMSI/1403/prog/prj/inc/Stoper.h File Reference

```
#include <iostream>
#include <chrono>
#include <string>
#include <fstream>
#include "IStoper.h"
```

Classes

- class [Stoper](#)

zastepuje kilkukrotne wklejanie pliku naglowkowego

5.13 /home/patr95/Pulpit/PAMSI/1403/prog/prj/src/Biegacz.cpp File Reference

```
#include "Biegacz.h"
```

5.14 /home/patr95/Pulpit/PAMSI/1403/prog/prj/src/BiegaczLista.cpp File Reference

```
#include "BiegaczLista.h"
```

5.15 /home/patr95/Pulpit/PAMSI/1403/prog/prj/src/Lista.cpp File Reference

```
#include "Lista.h"
```

5.16 /home/patr95/Pulpit/PAMSI/1403/prog/prj/src/main.cpp File Reference

```
#include <iostream>
#include "Sedzia.h"
#include "BiegaczLista.h"
#include "Stoper.h"
```

Functions

- int [main](#) ()

5.16.1 Function Documentation

5.16.1.1 int main ()

Definition at line 9 of file main.cpp.

5.17 /home/patr95/Pulpit/PAMSI/1403/prog/prj/src/Sedzia.cpp File Reference

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

5.18 /home/patr95/Pulpit/PAMSI/1403/prog/prj/src/Stoper.cpp File Reference

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