

PAMSI

3

Wygenerowano przez Doxygen 1.8.9.1

N, 3 kwi 2016 23:42:07



## Spis treści



# Rozdział 1

## Indeks hierarchiczny

### 1.1 Hierarchia klas

Ta lista dziedziczenia posortowana jest z grubsza, choć nie całkowicie, alfabetycznie:

atp	??
atp2	??
tablica	??
kolejka	??
lista	??
stos	??
tablicatest	??
listatest	??
atp2::bad_index	??
atp2::empty	??
iRunnable	??
tablicatest	??
iRunnable	
listatest	??
stoper	??



## Rozdział 2

# Indeks klas

### 2.1 Lista klas

Tutaj znajdują się klasy, struktury, unie i interfejsy wraz z ich krótkimi opisami:

<b>atp</b>	??
<b>atp2</b>	??
<b>atp2::bad_index</b>	??
<b>atp2::empty</b>	??
<b>iRunnable</b>	??
<b>kolejka</b>	??
<b>lista</b>	??
<b>listatest</b>	??
<b>stoper</b>	??
<b>stos</b>	??
<b>tablica</b>	??
<b>tablicatest</b>	??





## Rozdział 3

# Indeks plików

### 3.1 Lista plików

Tutaj znajduje się lista wszystkich plików z ich krótkimi opisami:

<b>atp.hh</b>	..	??
<b>atp2.hh</b>	Plik zawiera definicje klasy wirtualnej <b>atp2</b> (str. ??) z ktorej skladaja sie zaimplementowane abstrakcyjne typy danych	??
<b>irunnable.hh</b>	Zawira interfejs do testowania zaimplementowanych algorytmow	??
<b>kolejka.cpp</b>	..	??
<b>kolejka.hh</b>	Plik zawiera definicje klasy kolejka	??
<b>lista.cpp</b>	..	??
<b>lista.hh</b>	Plik zawiera klase lista	??
<b>listatest.cpp</b>	..	??
<b>listatest.hh</b>	Plik zawiera testowa klase lista	??
<b>main.cpp</b>	..	??
<b>stoper.cpp</b>	..	??
<b>stoper.hh</b>	Zawiera klase stoper odpowiadajaca za mierzenie czasu	??
<b>stos.cpp</b>	..	??
<b>stos.hh</b>	Plik zawiera definicje klasy stos	??
<b>tablica.cpp</b>	..	??
<b>tablica.hh</b>	Plik zawiera klase tablica	??
<b>tablicatest.cpp</b>	..	??
<b>tablicatest.hh</b>	Plik zawiera testowa klase tablica	??



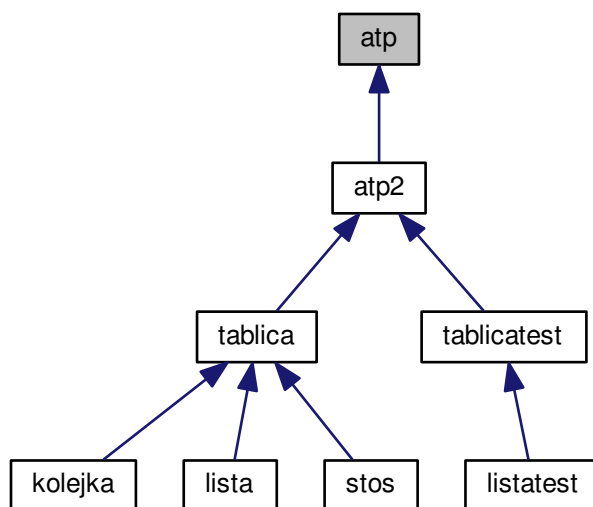
## Rozdział 4

# Dokumentacja klas

### 4.1 Dokumentacja klasy atp

```
#include <atp.hh>
```

Diagram dziedziczenia dla atp



#### Metody publiczne

- virtual void **remove** ()=0
- virtual void **add** (int dana)=0  
*usuwa element*
- virtual int **get** (int index)=0  
*dodaje element*
- virtual bool **isEmpty** ()=0  
*zwraca element*
- virtual int **size** ()=0

*sprawdza czy sa elementy*

#### 4.1.1 Opis szczegółowy

Definicja w linii 6 pliku atp.hh.

#### 4.1.2 Dokumentacja funkcji składowych

4.1.2.1 `virtual void atp::add ( int dana ) [pure virtual]`

usuwa element

Implementowany w **kolejka** (str. ??), **stos** (str. ??), **listatest** (str. ??) i **lista** (str. ??).

4.1.2.2 `virtual int atp::get ( int index ) [pure virtual]`

dodaje element

Implementowany w **kolejka** (str. ??), **stos** (str. ??), **listatest** (str. ??) i **lista** (str. ??).

4.1.2.3 `virtual bool atp::isEmpty ( ) [pure virtual]`

zwraca element

Implementowany w **atp2** (str. ??).

4.1.2.4 `virtual void atp::remove ( ) [pure virtual]`

Implementowany w **kolejka** (str. ??), **stos** (str. ??), **listatest** (str. ??) i **lista** (str. ??).

4.1.2.5 `virtual int atp::size ( ) [pure virtual]`

sprawdza czy sa elementy

Zwraca

true gdy sa elementy false gdy nie ma elementow

Implementowany w **atp2** (str. ??).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- **atp.hh**

## 4.2 Dokumentacja klasy atp2

```
#include <atp2.hh>
```

Diagram dziedziczenia dla atp2

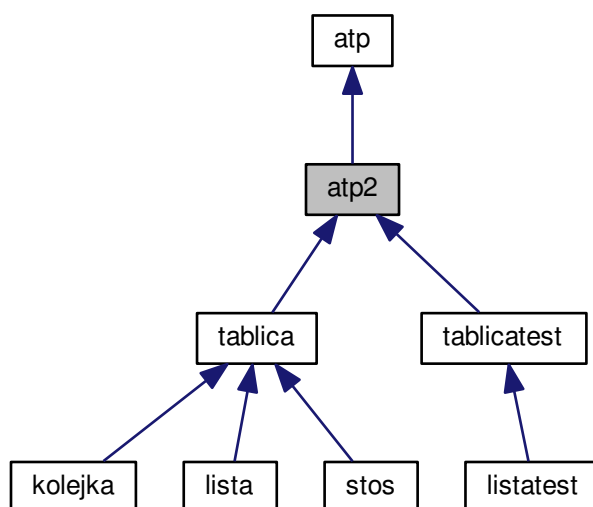
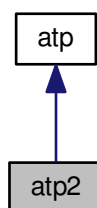


Diagram współpracy dla atp2:



### Komponenty

- class **bad\_index**
- class **empty**

### Metody publiczne

- int **size** ()  
*sprawdza czy sa elementy*
- int **ind** ()  
*metoda zwraca ilosc zaalokowanego miejsca*
- bool **isEmpty** ()  
*metoda zwraca indeks na którym jest ostatnia dana w kontenerze*

## Atrybuty chronione

- int **rozmiar**
- int **ile\_elem**

### 4.2.1 Opis szczegółowy

Definicja w linii 9 pliku atp2.hh.

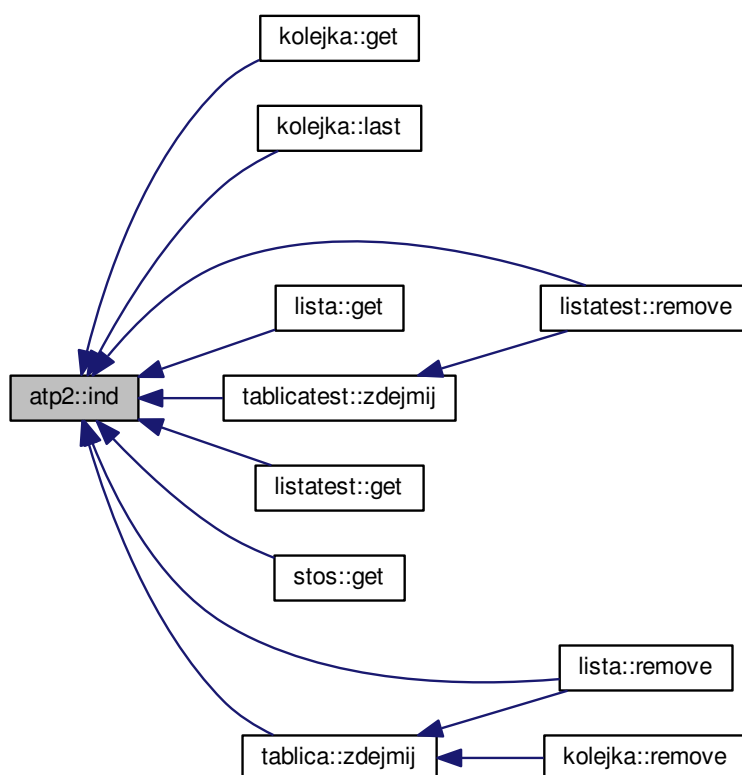
### 4.2.2 Dokumentacja funkcji składowych

#### 4.2.2.1 int atp2::ind ( ) [inline]

metoda zwraca ilosc zaalokowanego miejsca

Definicja w linii 17 pliku atp2.hh.

Oto graf wywoływań tej funkcji:



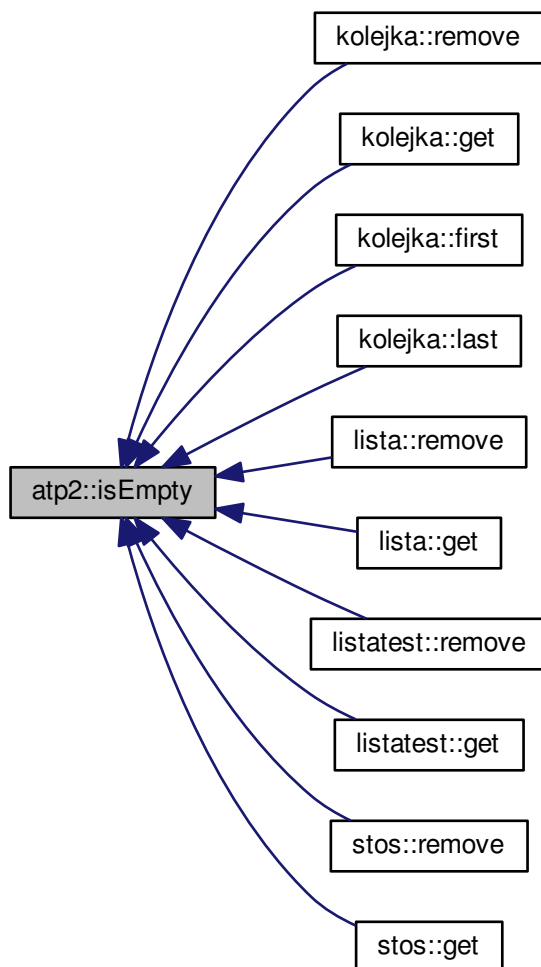
#### 4.2.2.2 bool atp2::isEmpty ( ) [inline],[virtual]

metoda zwraca indeks na którym jest ostatnia dana w kontenerze

Implementuje **atp** (str. ??).

Definicja w linii 18 pliku atp2.hh.

Oto graf wywołań tej funkcji:



**4.2.2.3** `int atp2::size ( ) [inline],[virtual]`

sprawdza czy sa elementy

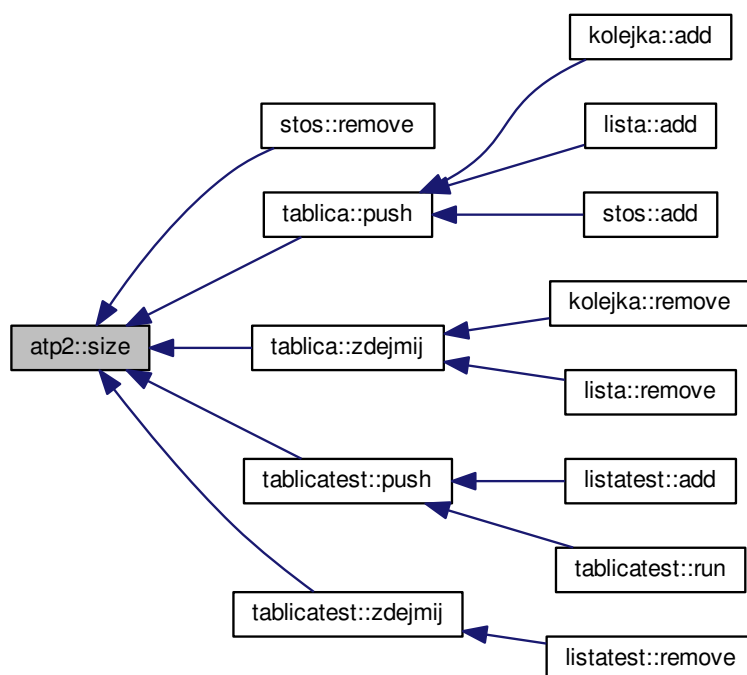
**Zwraca**

true gdy sa elementy false gdy nie ma elementow

Implementuje **atp** (str. ??).

Definicja w linii 16 pliku atp2.hh.

Oto graf wywoływań tej funkcji:



### 4.2.3 Dokumentacja atrybutów składowych

#### 4.2.3.1 `int atp2::ile_elem` `[protected]`

Definicja w linii 12 pliku atp2.hh.

#### 4.2.3.2 `int atp2::rozmiar` `[protected]`

Definicja w linii 11 pliku atp2.hh.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- **atp2.hh**

## 4.3 Dokumentacja klasy `atp2::bad_index`

```
#include <atp2.hh>
```



#### 4.3.1 Opis szczegółowy

Definicja w linii 15 pliku atp2.hh.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- **atp2.hh**

### 4.4 Dokumentacja klasy atp2::empty

```
#include <atp2.hh>
```

#### 4.4.1 Opis szczegółowy

Definicja w linii 14 pliku atp2.hh.

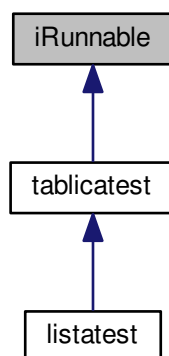
Dokumentacja dla tej klasy została wygenerowana z pliku:

- **atp2.hh**

### 4.5 Dokumentacja klasy iRunnable

```
#include <irunnable.hh>
```

Diagram dziedziczenia dla iRunnable



#### Metody publiczne

- virtual void **run** ()=0
- void **prepare** (int, int)

#### 4.5.1 Opis szczegółowy

Definicja w linii 6 pliku irunnable.hh.

## 4.5.2 Dokumentacja funkcji składowych

4.5.2.1 `void iRunnable::prepare ( int , int )`

4.5.2.2 `virtual void iRunnable::run ( ) [pure virtual]`

Implementowany w **tablicatest** (str. ??) i **listatest** (str. ??).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- **irunnable.hh**

## 4.6 Dokumentacja klasy kolejka

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

Diagram dziedziczenia dla kolejka

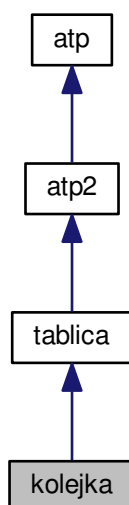
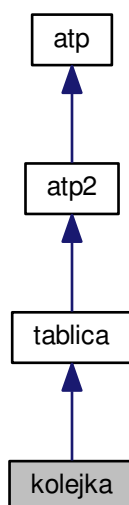


Diagram współpracy dla kolejki:



### Metody publiczne

- **kolejka** ()  
*Klasa kolejka - jeden z abstrakcyjnych typów danych.*
- void **remove** ()
- void **add** (int)  
*usuwa pierwsza dana w kolejce*
- int **get** (int)  
*dodaje dana na koniec kolejki*
- int **first** ()  
*zwraca wartosc, mozliwe jest tylko sprawdzenie ostatniej i pierwszej danej w kolejce, by sprawdzic ostatnia, jako argument podaj **ind()** (str. ??) lub skorzystaj z metod **first()** (str. ??) i **last()** (str. ??)*
- int **last** ()  
*zwraca wartosc pierwszej danej*

### Dodatkowe Dziedziczone Składowe

#### 4.6.1 Opis szczegółowy

Definicja w linii 9 pliku kolejka.hh.

#### 4.6.2 Dokumentacja konstruktora i destruktor

##### 4.6.2.1 kolejka::kolejka ( )

Klasa kolejka - jeden z abstrakcyjnych typów danych.

Klasa ma w swoim składzie metody służące do zarządzania kolejką, wszystkie operacje poza dodaniem nowej danej do pustej kolejki zgłaszają wyjątek

Definicja w linii 3 pliku kolejka.cpp.

### 4.6.3 Dokumentacja funkcji składowych

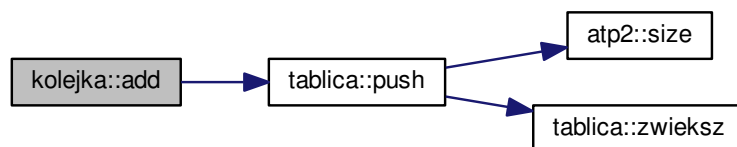
#### 4.6.3.1 void kolejka::add ( int x ) [virtual]

usuwa pierwsza dana w kolejce

Implementuje **atp** (str. ??).

Definicja w linii 6 pliku kolejka.cpp.

Oto graf wywołań dla tej funkcji:

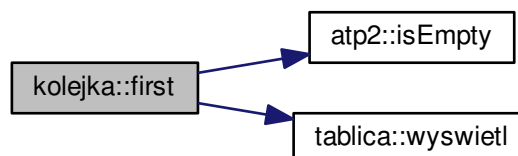


#### 4.6.3.2 int kolejka::first ( )

zwraca wartosc, mozliwe jest tylko sprawdzenie ostatniej i pierwszej danej w kolejce, by sprawdzic ostatnia, jako argument podaj **ind()** (str. ??) lub skorzystaj z metod **first()** (str. ??) i **last()** (str. ??)

Definicja w linii 26 pliku kolejka.cpp.

Oto graf wywołań dla tej funkcji:



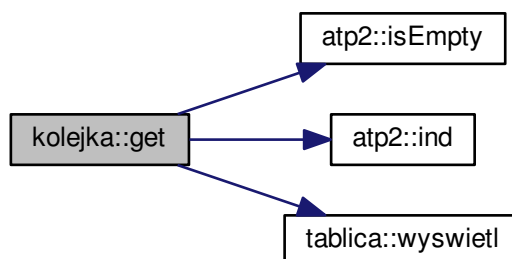
#### 4.6.3.3 int kolejka::get ( int i ) [virtual]

dodaje dana na koniec kolejki

Implementuje **atp** (str. ??).

Definicja w linii 17 pliku kolejka.cpp.

Oto graf wywołań dla tej funkcji:

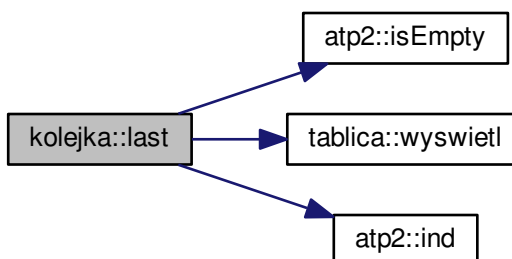


#### 4.6.3.4 `int kolejka::last ( )`

zwraca wartosc pierwszej danej

Definicja w linii 33 pliku kolejka.cpp.

Oto graf wywołań dla tej funkcji:

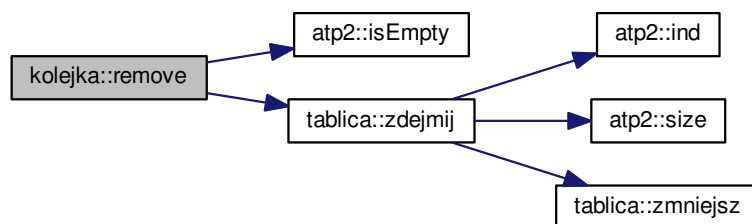


#### 4.6.3.5 `void kolejka::remove ( ) [virtual]`

Implementuje **atp** (str. ??).

Definicja w linii 10 pliku kolejka.cpp.

Oto graf wywołań dla tej funkcji:



Dokumentacja dla tej klasy została wygenerowana z plików:

- `kolejka.hh`
- `kolejka.cpp`

## 4.7 Dokumentacja klasy lista

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

Diagram dziedziczenia dla lista

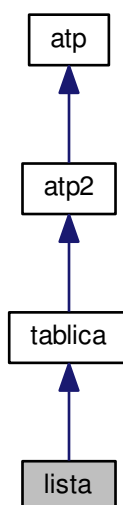
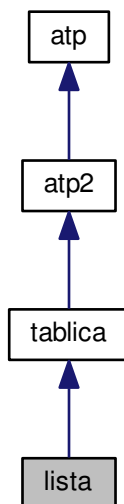


Diagram współpracy dla lista:



### Metody publiczne

- **lista** ()
- void **remove** (int)
- void **remove** ()  
*usuwa dana o podanym indeksie, a nastepnie przestawia zmienne*
- void **add** (int, int)  
*usuwa ostatnia dana w kolejce*
- void **add** (int)  
*dodaje dana na miejscu o podanym indeksie, jezeli nie moze byc tam wstawiona zglosi wyjatek, gdy miejsce jest zajete przestawia dane*
- int **get** (int)  
*dodaje dana na koniec listy*

### Dodatkowe Dziedziczone Składowe

#### 4.7.1 Opis szczegółowy

Definicja w linii 8 pliku lista.hh.

#### 4.7.2 Dokumentacja konstruktora i destruktor

##### 4.7.2.1 lista::lista ( )

Definicja w linii 3 pliku lista.cpp.

### 4.7.3 Dokumentacja funkcji składowych

#### 4.7.3.1 void lista::add ( int x, int index )

usuwa ostatnia dana w kolejce

Definicja w linii 10 pliku lista.cpp.

Oto graf wywołań dla tej funkcji:



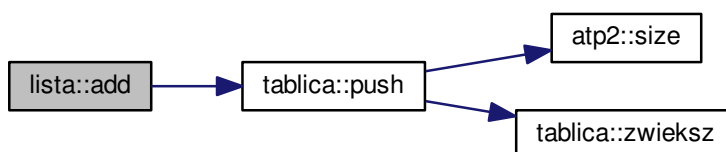
#### 4.7.3.2 void lista::add ( int x ) [virtual]

dodaje dana na miejscu o podanym indeksie, jezeli nie moze byc tam wstawiona zglosi wyjatek, gdy miejsce jest zajete przestawia dane

Implementuje **atp** (str. ??).

Definicja w linii 6 pliku lista.cpp.

Oto graf wywołań dla tej funkcji:



#### 4.7.3.3 int lista::get ( int i ) [virtual]

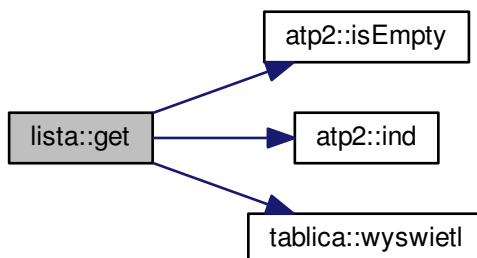
dodaje dana na koniec listy

Implementuje **atp** (str. ??).

Definicja w linii 33 pliku lista.cpp.



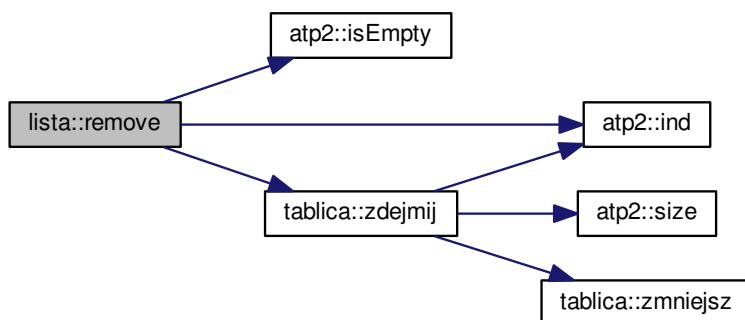
Oto graf wywołań dla tej funkcji:



#### 4.7.3.4 void lista::remove ( int i )

Definicja w linii 17 pliku lista.cpp.

Oto graf wywołań dla tej funkcji:



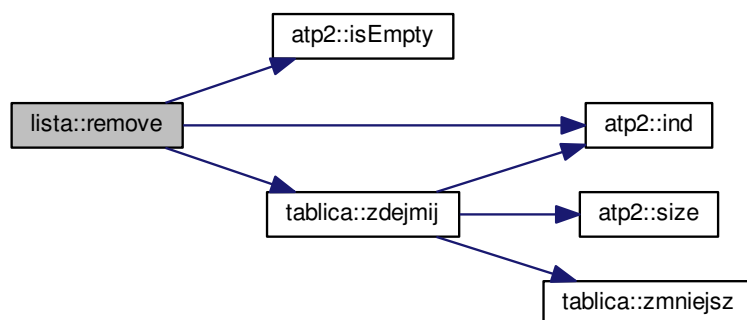
#### 4.7.3.5 void lista::remove ( ) [virtual]

usuwa dana o podanym indeksie, a następnie przestawia zmienne

Implementuje **atp** (str. ??).

Definicja w linii 26 pliku lista.cpp.

Oto graf wywołań dla tej funkcji:



Dokumentacja dla tej klasy została wygenerowana z plików:

- `lista.hh`
- `lista.cpp`

## 4.8 Dokumentacja klasy listatest

```
#include <listatest.hh>
```

Diagram dziedziczenia dla listatest

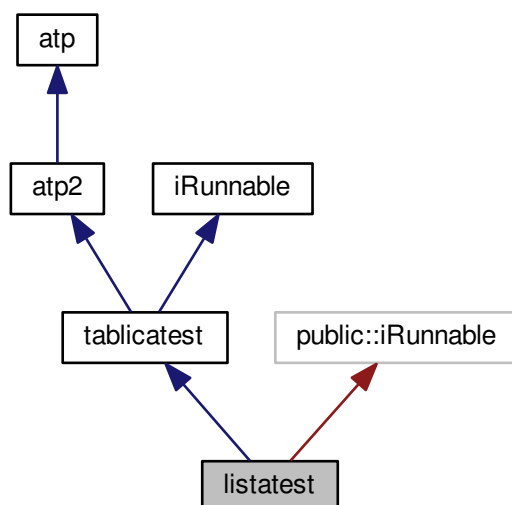
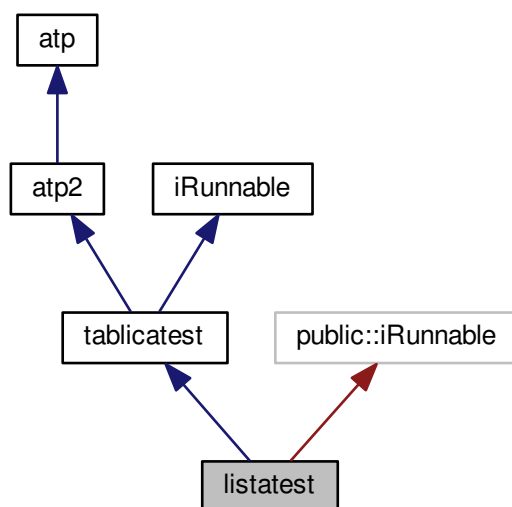


Diagram współpracy dla listatest:



### Metody publiczne

- **listatest** ()
- void **run** ()
- void **remove** (int)
- void **remove** ()
  - usuwa dana o podanym indeksie, a nastepnie przestawia zmienne*
- void **add** (int, int)
  - usuwa ostatnia dana w kolejce*
- void **add** (int)
  - dodaje dana na miejscu o podanym indeksie, jezeli nie moze byc tam wstawiona zglosi wyjatek, gdy miejsce jest zajete przestawia dane*
- int **get** (int)
  - dodaje dana na koniec listy*

### Dodatkowe Dziedziczone Składowe

#### 4.8.1 Opis szczegółowy

Definicja w linii 8 pliku listatest.hh.

#### 4.8.2 Dokumentacja konstruktora i destruktoru

##### 4.8.2.1 listatest::listatest ( )

Definicja w linii 3 pliku listatest.cpp.

### 4.8.3 Dokumentacja funkcji składowych

#### 4.8.3.1 void listatest::add ( int x, int index )

usuwa ostatnia dana w kolejce

Definicja w linii 10 pliku listatest.cpp.

Oto graf wywołań dla tej funkcji:



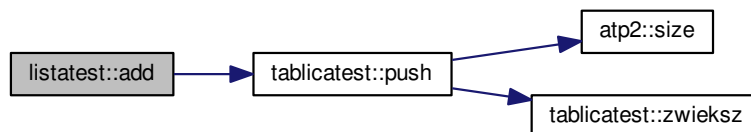
#### 4.8.3.2 void listatest::add ( int x ) [virtual]

dodaje dana na miejscu o podanym indeksie, jezeli nie moze byc tam wstawiona zglosi wyjatek, gdy miejsce jest zajete przestawia dane

Implementuje **atp** (str. ??).

Definicja w linii 6 pliku listatest.cpp.

Oto graf wywołań dla tej funkcji:



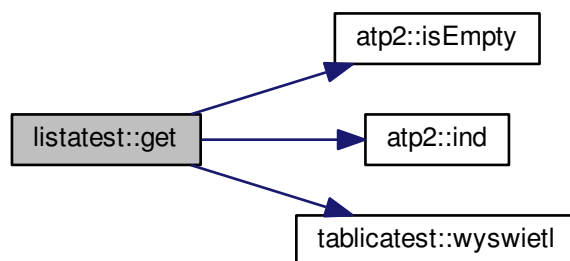
#### 4.8.3.3 int listatest::get ( int i ) [virtual]

dodaje dana na koniec listy

Implementuje **atp** (str. ??).

Definicja w linii 33 pliku listatest.cpp.

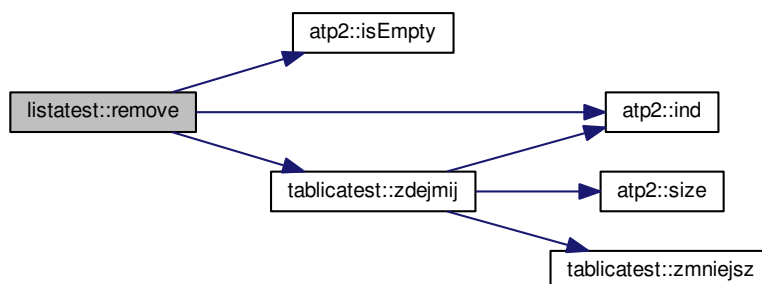
Oto graf wywołań dla tej funkcji:



#### 4.8.3.4 void listatest::remove ( int i )

Definicja w linii 17 pliku listatest.cpp.

Oto graf wywołań dla tej funkcji:



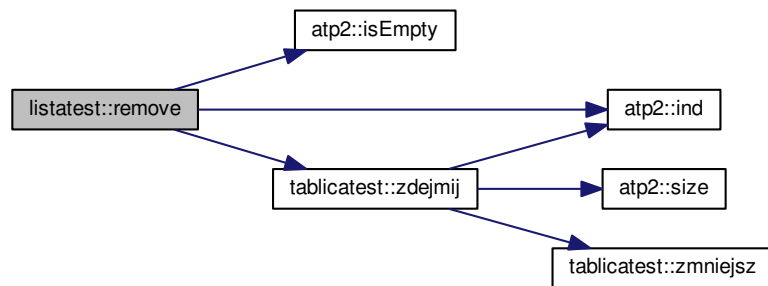
#### 4.8.3.5 void listatest::remove ( ) [virtual]

usuwa dana o podanym indeksie, a następnie przestawia zmienne

Implementuje **atp** (str. ??).

Definicja w linii 26 pliku listatest.cpp.

Oto graf wywołań dla tej funkcji:

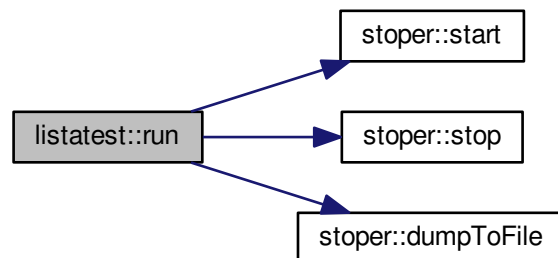


#### 4.8.3.6 void listatest::run ( ) [virtual]

Implementuje **iRunnable** (str. ??).

Definicja w linii 42 pliku `listatest.cpp`.

Oto graf wywołań dla tej funkcji:



Dokumentacja dla tej klasy została wygenerowana z plików:

- `listatest.hh`
- `listatest.cpp`

## 4.9 Dokumentacja klasy stoper

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

### Metody publiczne

- void **start** ()

- void **stop** ()
- std::chrono::duration< double > **getElapsedTime** ()
- std::chrono::duration< double > **getTime** ()
- bool **dumpToFile** (std::string)

#### 4.9.1 Opis szczegółowy

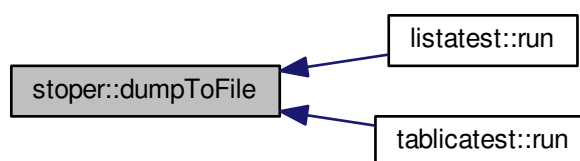
Definicja w linii 10 pliku stoper.hh.

#### 4.9.2 Dokumentacja funkcji składowych

##### 4.9.2.1 bool stoper::dumpToFile ( std::string *Nazwa* )

Definicja w linii 23 pliku stoper.cpp.

Oto graf wywołań tej funkcji:



##### 4.9.2.2 std::chrono::duration< double > stoper::getElapsedTime ( )

Definicja w linii 11 pliku stoper.cpp.

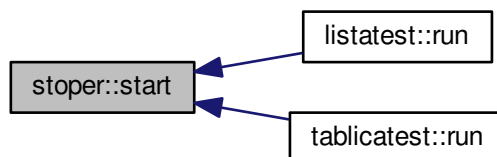
##### 4.9.2.3 std::chrono::duration< double > stoper::getTime ( )

Definicja w linii 18 pliku stoper.cpp.

##### 4.9.2.4 void stoper::start ( )

Definicja w linii 3 pliku stoper.cpp.

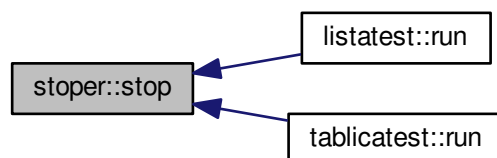
Oto graf wywoływań tej funkcji:



#### 4.9.2.5 void stoper::stop ( )

Definicja w linii 7 pliku `stoper.cpp`.

Oto graf wywoływań tej funkcji:



Dokumentacja dla tej klasy została wygenerowana z plików:

- **stoper.hh**
- **stoper.cpp**

## 4.10 Dokumentacja klasy stos

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



Diagram dziedziczenia dla stos

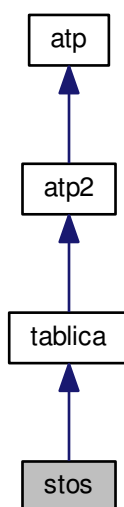
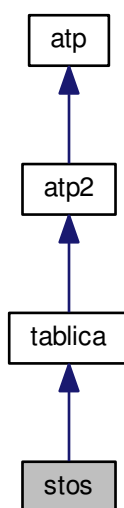


Diagram współpracy dla stos:



### Metody publiczne

- **stos ()**

*Klasa stos - jeden z abstrakcyjnych typów danych.*

- void **remove** ()

*konstruktor bezparametryczny*

- void **add** (int)

*metoda **remove()** (str. ??)- nie przyjmuje wartosci, usuwa najwyzej polozona na stosie dana*

- int **get** (int=0)

*metoda **add(int)** (str. ??)- przyjmuje wartosc int, bedaca nowa liczba do dodania na szczyt stosu*

- int **get** ()

*metoda get z parametrem int nie powinna byc uzywana, mozliwe jest uzycie jedynie zerowego indeksu, wszystkie inne zwracaja wyjatek, bezpiecniej jest uzywac **get()** (str. ??)*

## Dodatkowe Dziedziczone Składowe

### 4.10.1 Opis szczegółowy

Definicja w linii 8 pliku stos.hh.

### 4.10.2 Dokumentacja konstruktora i destruktor

#### 4.10.2.1 stos::stos ( )

Klasa stos - jeden z abstrakcyjnych typow danych.

Klasa ma w swoim skladzie metody sluzace do zarzadzania stosem, mozliwe jest jedynie manipulowanie najwyzej polozonym elementem na stosie, inne operacje poza dodaniem nowej danej do niego na pustym stosie zgłaszają wyjątek

Definicja w linii 3 pliku stos.cpp.

### 4.10.3 Dokumentacja funkcji składowych

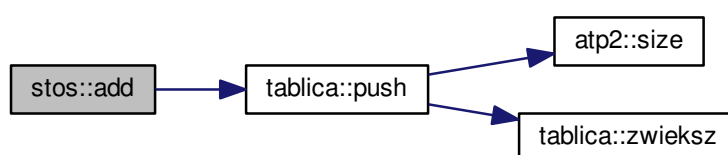
#### 4.10.3.1 void stos::add ( int x ) [virtual]

metoda **remove()** (str. ??)- nie przyjmuje wartosci, usuwa najwyzej polozona na stosie dana

Implementuje **atp** (str. ??).

Definicja w linii 5 pliku stos.cpp.

Oto graf wywołań dla tej funkcji:



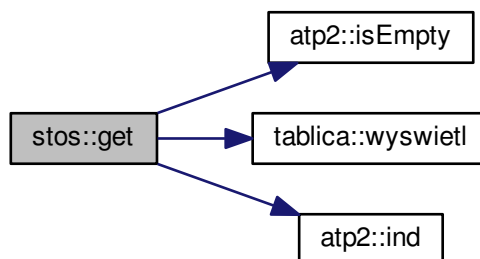
#### 4.10.3.2 `int stos::get ( int i = 0 ) [virtual]`

metoda **add(int)** (str. ??)- przyjmuje wartosc int, bedaca nowa liczba do dodania na szczyt stosu

Implementuje **atp** (str. ??).

Definicja w linii 18 pliku stos.cpp.

Oto graf wywołań dla tej funkcji:

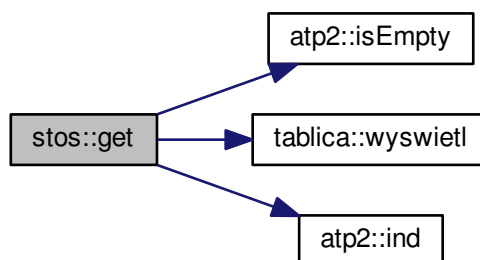


#### 4.10.3.3 `int stos::get ( )`

metoda `get` z parametrem `int` nie powinna byc uzywana, mozliwe jest uzycie jedynie zerowego indeksu, wszystkie inne zwracaja wyjatke, bezpieczniej jest uzywac **get()** (str. ??)

Definicja w linii 27 pliku stos.cpp.

Oto graf wywołań dla tej funkcji:



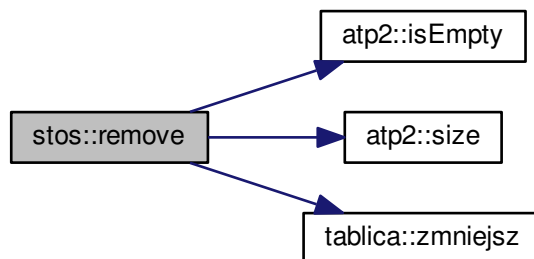
#### 4.10.3.4 `void stos::remove ( ) [virtual]`

konstruktor bezparametryczny

Implementuje **atp** (str. ??).

Definicja w linii 9 pliku stos.cpp.

Oto graf wywołań dla tej funkcji:



Dokumentacja dla tej klasy została wygenerowana z plików:

- **stos.hh**
- **stos.cpp**

## 4.11 Dokumentacja klasy tablica

```
#include <tablica.hh>
```

Diagram dziedziczenia dla tablica

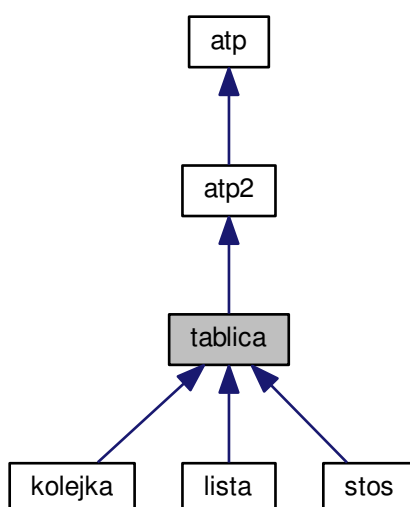
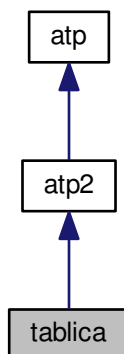


Diagram współpracy dla `tablica`:



### Metody chronione

- `tablica()`
- `tablica(int n)`
- `int wyswietl(int n)`
- `void push(int)`
- `void zwieksz(int)`
- `void przypisz(int, int)`
- `void zmniejsz()`
- `void zdejmij(int)`

### Dodatkowe Dziedziczone Składowe

#### 4.11.1 Opis szczegółowy

Definicja w linii 8 pliku `tablica.hh`.

#### 4.11.2 Dokumentacja konstruktora i destruktor

##### 4.11.2.1 `tablica::tablica( )` `[inline]`, `[protected]`

Definicja w linii 16 pliku `tablica.hh`.

##### 4.11.2.2 `tablica::tablica( int n )` `[inline]`, `[protected]`

Definicja w linii 17 pliku `tablica.hh`.

#### 4.11.3 Dokumentacja funkcji składowych

##### 4.11.3.1 `void tablica::przypisz( int dana, int miejsce )` `[protected]`

Definicja w linii 22 pliku `tablica.cpp`.

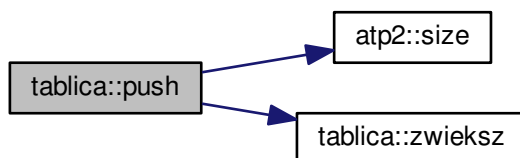
Oto graf wywołań tej funkcji:



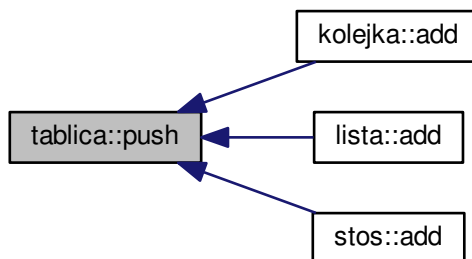
#### 4.11.3.2 void tablica::push ( int *dana* ) [protected]

Definicja w linii 2 pliku `tablica.cpp`.

Oto graf wywołań dla tej funkcji:



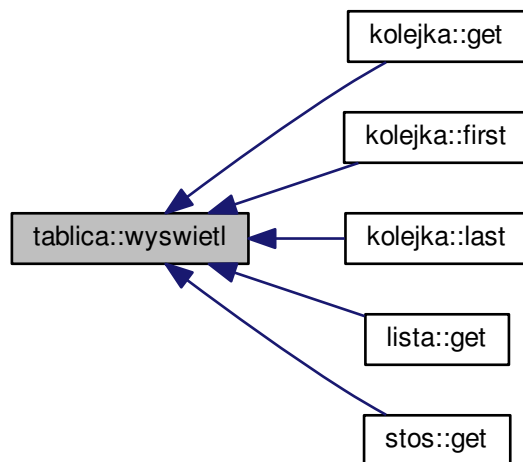
Oto graf wywołań tej funkcji:



#### 4.11.3.3 int tablica::wyswietl ( int *n* ) [inline], [protected]

Definicja w linii 19 pliku `tablica.hh`.

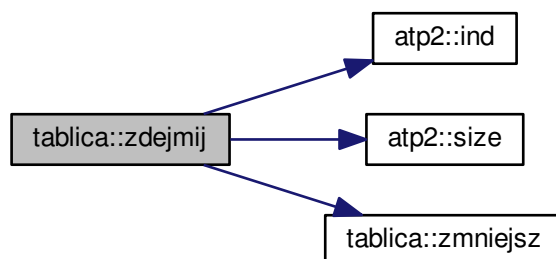
Oto graf wywoływań tej funkcji:



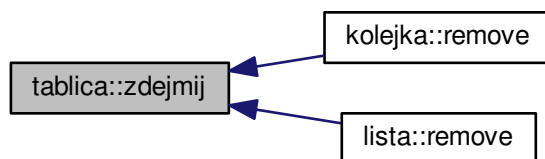
#### 4.11.3.4 void tablica::zdejmij ( int *index* ) [protected]

Definicja w linii 35 pliku `tablica.cpp`.

Oto graf wywołań dla tej funkcji:



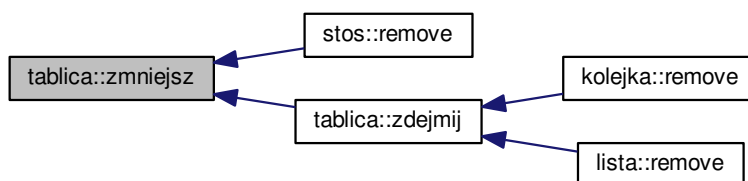
Oto graf wywoływań tej funkcji:



#### 4.11.3.5 `void tablica::zmniejsz ( )` [protected]

Definicja w linii 52 pliku `tablica.cpp`.

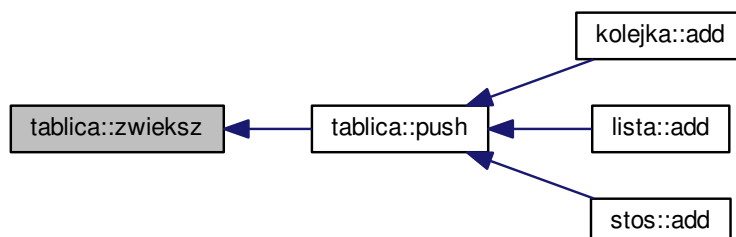
Oto graf wywoływań tej funkcji:



#### 4.11.3.6 `void tablica::zwieksz ( int dana )` [protected]

Definicja w linii 10 pliku `tablica.cpp`.

Oto graf wywoływań tej funkcji:





Dokumentacja dla tej klasy została wygenerowana z plików:

- **tablica.hh**
- **tablica.cpp**

## 4.12 Dokumentacja klasy tablicatest

```
#include <tablicatest.hh>
```

Diagram dziedziczenia dla tablicatest

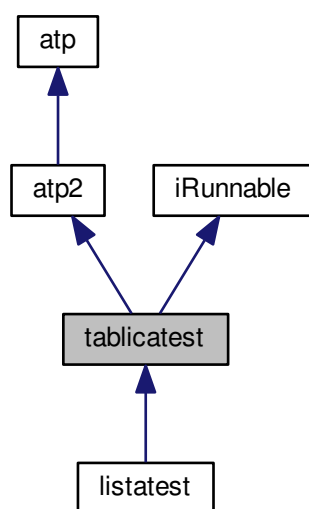
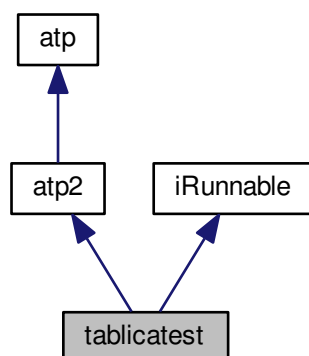


Diagram współpracy dla tablicatest:



## Metody chronione

- **tablicatest** ()
- **tablicatest** (int *n*)
- void **run** ()
- int **wyświetl** (int *n*)
- void **push** (int)
- void **zwiększ** (int)
- void **przypisz** (int, int)
- void **zmniejsz** ()
- void **zdejmij** (int)

## Dodatkowe Dziedziczone Składowe

### 4.12.1 Opis szczegółowy

Definicja w linii 8 pliku tablicatest.hh.

### 4.12.2 Dokumentacja konstruktora i destruktora

#### 4.12.2.1 `tablicatest::tablicatest ( )` [inline], [protected]

Definicja w linii 16 pliku tablicatest.hh.

#### 4.12.2.2 `tablicatest::tablicatest ( int n )` [inline], [protected]

Definicja w linii 17 pliku tablicatest.hh.

### 4.12.3 Dokumentacja funkcji składowych

#### 4.12.3.1 `void tablicatest::przypisz ( int dana, int miejsce )` [protected]

Definicja w linii 22 pliku tablicatest.cpp.

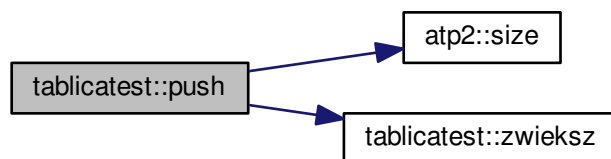
Oto graf wywoływań tej funkcji:



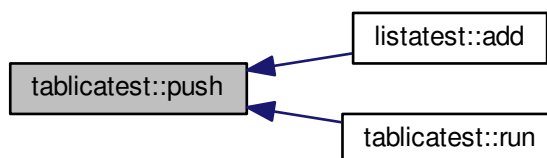
#### 4.12.3.2 `void tablicatest::push ( int dana )` [protected]

Definicja w linii 2 pliku tablicatest.cpp.

Oto graf wywołań dla tej funkcji:



Oto graf wywołań tej funkcji:

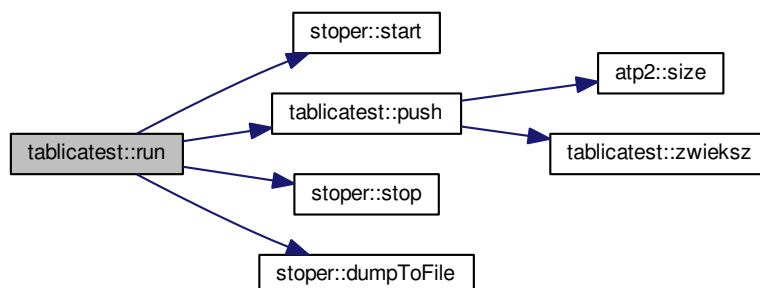


#### 4.12.3.3 `void tablicatest::run ( ) [protected], [virtual]`

Implementuje **iRunnable** (str. ??).

Definicja w linii 61 pliku `tablicatest.cpp`.

Oto graf wywołań dla tej funkcji:



#### 4.12.3.4 `int tablicatest::wyswietl ( int n ) [inline],[protected]`

Definicja w linii 20 pliku `tablicatest.hh`.

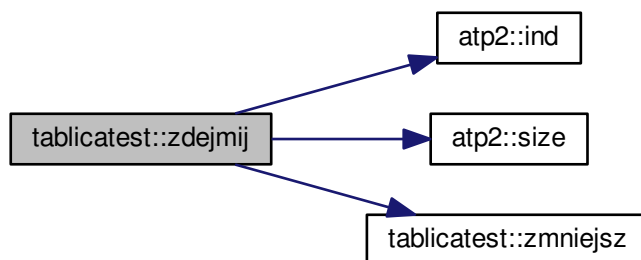
Oto graf wywołań tej funkcji:



#### 4.12.3.5 `void tablicatest::zdejmij ( int index ) [protected]`

Definicja w linii 35 pliku `tablicatest.cpp`.

Oto graf wywołań dla tej funkcji:



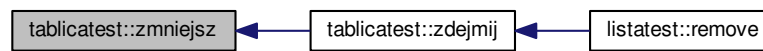
Oto graf wywołań tej funkcji:



#### 4.12.3.6 `void tablicatest::zmniejsz ( ) [protected]`

Definicja w linii 52 pliku `tablicatest.cpp`.

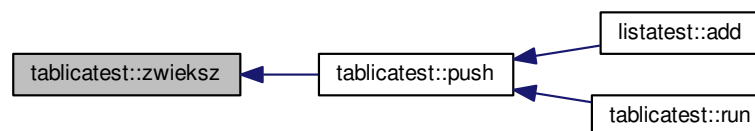
Oto graf wywoływań tej funkcji:



#### 4.12.3.7 void tablicatest::zwiększ ( int *dana* ) [protected]

Definicja w linii 10 pliku tablicatest.cpp.

Oto graf wywoływań tej funkcji:



Dokumentacja dla tej klasy została wygenerowana z plików:

- **tablicatest.hh**
- **tablicatest.cpp**

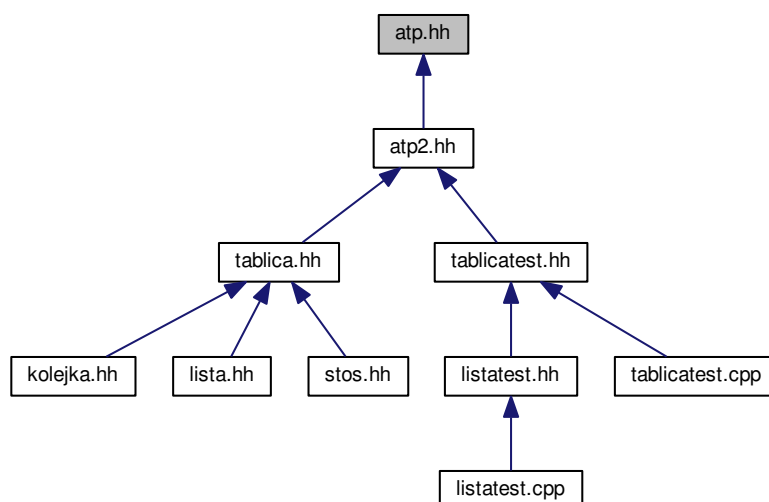


## Rozdział 5

# Dokumentacja plików

### 5.1 Dokumentacja pliku atp.hh

Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



#### Komponenty

- class **atp**

#### 5.1.1 Opis szczegółowy

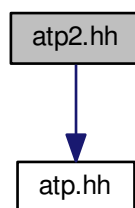
zawiera interfejs dla kolejki, listy i stosu

## 5.2 Dokumentacja pliku atp2.hh

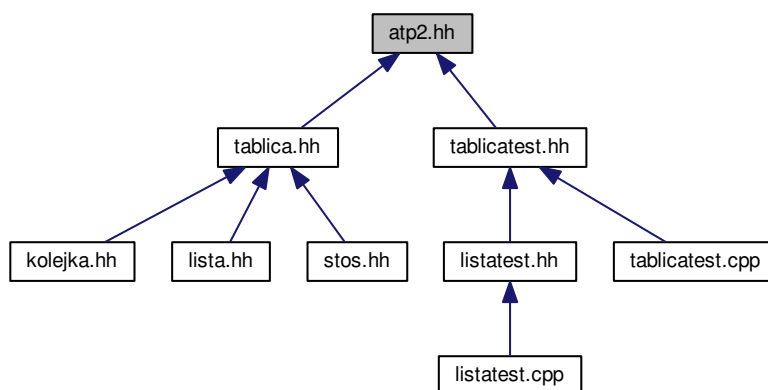
plik zawiera definicje klasy wirtualnej **atp2** (str. ??) z której składają się zaimplementowane abstrakcyjne typy danych

```
#include "atp.hh"
```

Wykres zależności załączania dla atp2.hh:



Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



### Komponenty

- class **atp2**
- class **atp2::empty**
- class **atp2::bad\_index**

#### 5.2.1 Opis szczegółowy

plik zawiera definicje klasy wirtualnej **atp2** (str. ??) z której składają się zaimplementowane abstrakcyjne typy danych



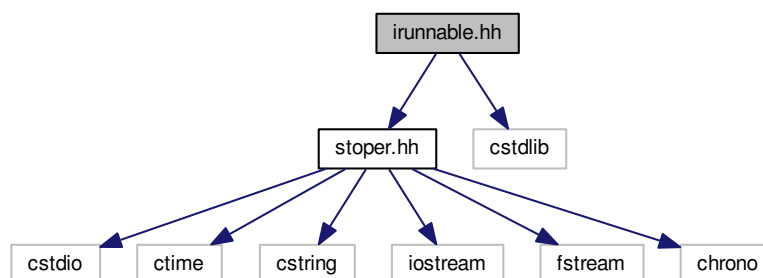
## 5.3 Dokumentacja pliku irunnable.hh

zawiera interfejs do testowania zaimplementowanych algorytmów

```
#include "stoper.hh"
```

```
#include <cstdlib>
```

Wykres zależności załączania dla irunnable.hh:



### Komponenty

- class **iRunnable**

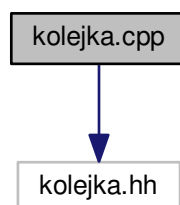
#### 5.3.1 Opis szczegółowy

zawiera interfejs do testowania zaimplementowanych algorytmów

## 5.4 Dokumentacja pliku kolejka.cpp

```
#include "kolejka.hh"
```

Wykres zależności załączania dla kolejka.cpp:

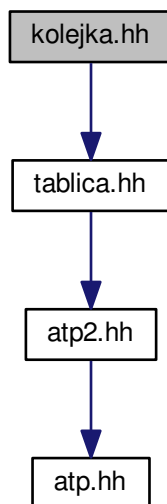


## 5.5 Dokumentacja pliku kolejka.hh

plik zawiera definicje klasy kolejka

```
#include "tablica.hh"
```

Wykres zależności załączania dla kolejka.hh:



### Komponenty

- class **kolejka**

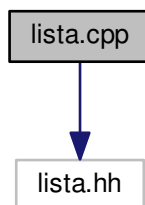
#### 5.5.1 Opis szczegółowy

plik zawiera definicje klasy kolejka

## 5.6 Dokumentacja pliku lista.cpp

```
#include "lista.hh"
```

Wykres zależności załączania dla lista.cpp:

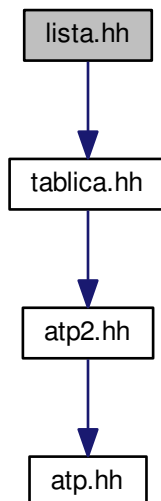


## 5.7 Dokumentacja pliku lista.hh

plik zawiera klasę lista

```
#include "tablica.hh"
```

Wykres zależności załączania dla lista.hh:



### Komponenty

- class **lista**

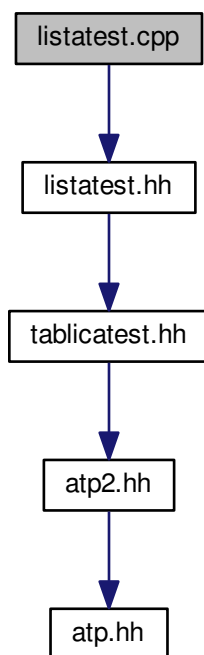
### 5.7.1 Opis szczegółowy

plik zawiera klasę lista

## 5.8 Dokumentacja pliku listatest.cpp

```
#include "listatest.hh"
```

Wykres zależności załączania dla listatest.cpp:

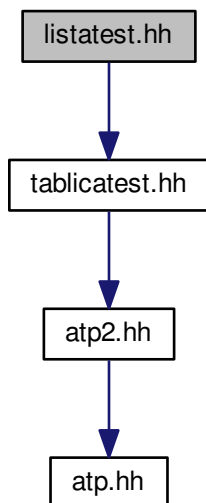


## 5.9 Dokumentacja pliku listatest.hh

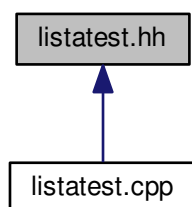
plik zawiera testową klasę lista

```
#include "tablicatest.hh"
```

Wykres zależności załączania dla listatest.hh:



Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



## Komponenty

- class **listatest**

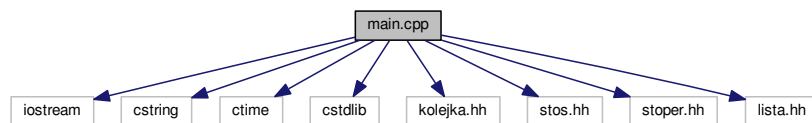
### 5.9.1 Opis szczegółowy

plik zawiera testową klasę lista

## 5.10 Dokumentacja pliku main.cpp

```
#include <iostream>
#include <cstring>
#include <ctime>
#include <cstdlib>
#include "kolejka.hh"
#include "stos.hh"
#include "stoper.hh"
#include "lista.hh"
```

Wykres zależności załączania dla main.cpp:



### Funkcje

- int **main** (void)

### 5.10.1 Dokumentacja funkcji

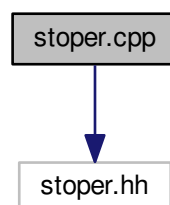
#### 5.10.1.1 int main ( void )

Definicja w linii 9 pliku main.cpp.

## 5.11 Dokumentacja pliku stoper.cpp

```
#include "stoper.hh"
```

Wykres zależności załączania dla stoper.cpp:

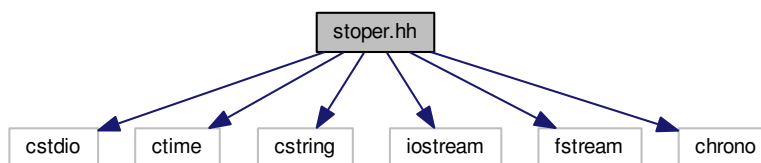


## 5.12 Dokumentacja pliku stoper.hh

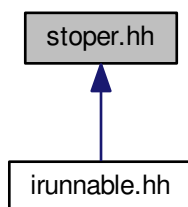
zawiera klasę stoper odpowiadającą za mierzenie czasu

```
#include <cstdio>
#include <ctime>
#include <cstring>
#include <iostream>
#include <fstream>
#include <chrono>
```

Wykres zależności załączania dla stoper.hh:



Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



### Komponenty

- class **stoper**

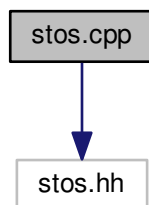
#### 5.12.1 Opis szczegółowy

zawiera klasę stoper odpowiadającą za mierzenie czasu

### 5.13 Dokumentacja pliku stos.cpp

```
#include "stos.hh"
```

Wykres zależności załączania dla stos.cpp:

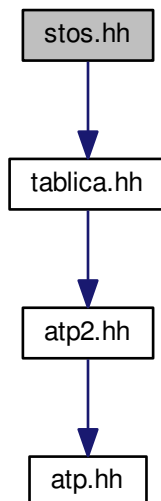


### 5.14 Dokumentacja pliku stos.hh

plik zawiera definicje klasy `stos`

```
#include "tablica.hh"
```

Wykres zależności załączania dla stos.hh:



#### Komponenty

- class **stos**



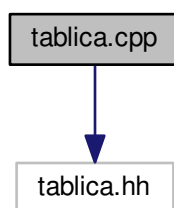
### 5.14.1 Opis szczegółowy

plik zawiera definicje klasy stos

## 5.15 Dokumentacja pliku tablica.cpp

```
#include "tablica.hh"
```

Wykres zależności załączania dla tablica.cpp:

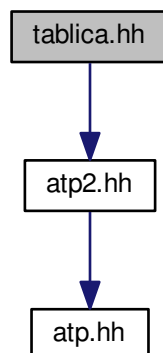


## 5.16 Dokumentacja pliku tablica.hh

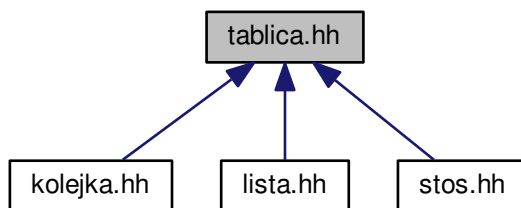
plik zawiera klasę `tablica`

```
#include "atp2.hh"
```

Wykres zależności załączania dla tablica.hh:



Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



## Komponenty

- class **tablica**

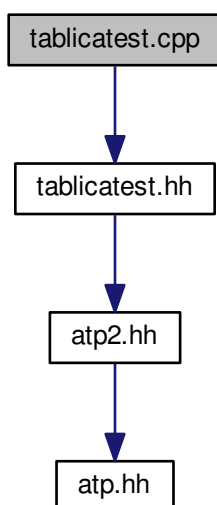
### 5.16.1 Opis szczegółowy

plik zawiera klasę `tablica`

## 5.17 Dokumentacja pliku `tablicatest.cpp`

```
#include "tablicatest.hh"
```

Wykres zależności załączania dla `tablicatest.cpp`:

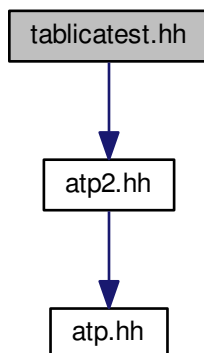


## 5.18 Dokumentacja pliku tablicatest.hh

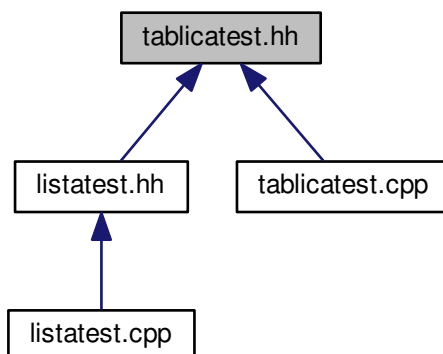
plik zawiera testowa klase tablica

```
#include "atp2.hh"
```

Wykres zależności załączania dla tablicatest.hh:



Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



### Komponenty

- class **tablicatest**

#### 5.18.1 Opis szczegółowy

plik zawiera testowa klase tablica

