

## Podstawowe struktury danych

1

Wygenerowano przez Doxygen 1.8.6

Śr, 18 mar 2015 08:07:00



# Spis treści

<b>1</b>	<b>Sprawozdanie</b>	<b>1</b>
1.1	Zadanie . . . . .	1
1.2	Wyniki . . . . .	1
1.3	Podsumowanie . . . . .	2
<b>2</b>	<b>Indeks hierarchiczny</b>	<b>3</b>
2.1	Hierarchia klas . . . . .	3
<b>3</b>	<b>Indeks klas</b>	<b>5</b>
3.1	Lista klas . . . . .	5
<b>4</b>	<b>Indeks plików</b>	<b>7</b>
4.1	Lista plików . . . . .	7
<b>5</b>	<b>Dokumentacja klas</b>	<b>9</b>
5.1	Dokumentacja klasy benchmark . . . . .	9
5.1.1	Opis szczegółowy . . . . .	9
5.1.2	Dokumentacja funkcji składowych . . . . .	9
5.1.2.1	analize . . . . .	9
5.1.2.2	test . . . . .	10
5.2	Dokumentacja klasy tabx2 . . . . .	11
5.2.1	Opis szczegółowy . . . . .	11
5.2.2	Dokumentacja konstruktora i destruktora . . . . .	12
5.2.2.1	tabx2 . . . . .	12
5.2.2.2	~tabx2 . . . . .	12
5.2.3	Dokumentacja funkcji składowych . . . . .	12
5.2.3.1	test . . . . .	12
5.2.4	Dokumentacja atrybutów składowych . . . . .	12
5.2.4.1	size . . . . .	12
5.2.4.2	tab . . . . .	12
<b>6</b>	<b>Dokumentacja plików</b>	<b>13</b>
6.1	Dokumentacja pliku benchmark.cpp . . . . .	13

6.2	Dokumentacja pliku benchmark.hh . . . . .	13
6.3	Dokumentacja pliku generator.cpp . . . . .	14
6.3.1	Dokumentacja funkcji . . . . .	15
6.3.1.1	data_generator . . . . .	15
6.4	Dokumentacja pliku generator.hh . . . . .	15
6.4.1	Dokumentacja funkcji . . . . .	16
6.4.1.1	data_generator . . . . .	16
6.5	Dokumentacja pliku main.cpp . . . . .	16
6.5.1	Dokumentacja funkcji . . . . .	17
6.5.1.1	main . . . . .	17
6.6	Dokumentacja pliku strona.dox . . . . .	17
6.7	Dokumentacja pliku tabx2.cpp . . . . .	17
6.8	Dokumentacja pliku tabx2.hh . . . . .	18
6.8.1	Opis szczegółowy . . . . .	19
6.8.2	Dokumentacja definicji . . . . .	19
6.8.2.1	TABX2_HH . . . . .	19
<b>Indeks</b>		<b>20</b>

# Rozdział 1

## Sprawozdanie

Data

11.03.2015r.

Wersja

0.1

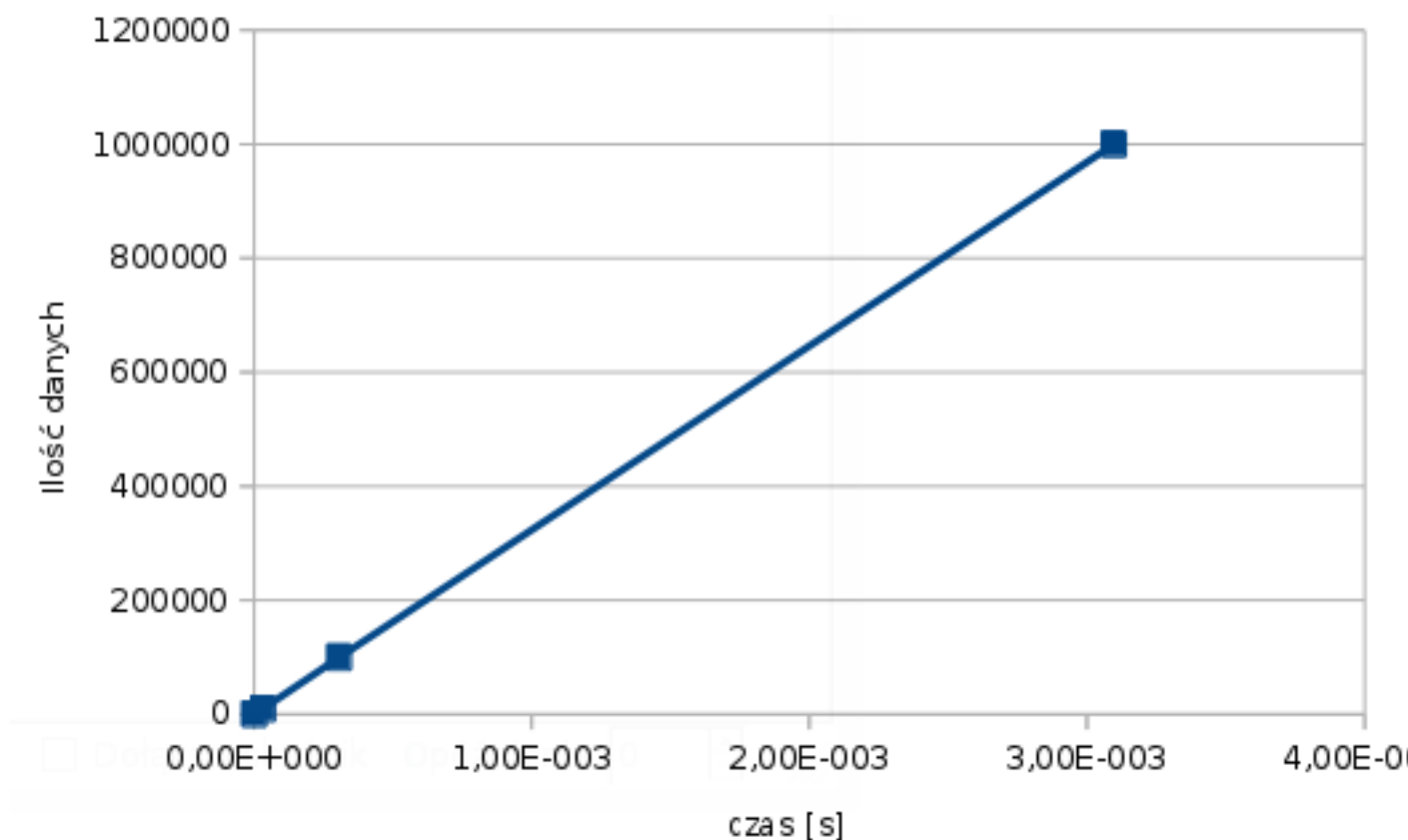
### 1.1 Zadanie

Celem ćwiczenia było stworzenie programu benchmarkującego , który dla wybranych danych będzie zliczał średni czas wykonania dowolnego algorytmu ( w tym przypadku mnożenia elementów tablicy przez 2). Należało również stworzyć program generujący losowe liczby.

### 1.2 Wyniki

Dla dziesięciu milionów liczb program zwraca 7 danych wyjściowych ( zgodnie z algorytmem  $10^n$  , gdzie n jest równocześnie ilością zwracanych czasów oraz maksymalną liczbą danych dla jakiej przeprowadzany był test

Na podstawie otrzymanych danych mamy :



### 1.3 Podsumowanie

Wykres dodany do dokumentacji z niewiadomych względów nie jest wyświetlany poprawnie (dodano sprawozdanie również w formacie pdf). Zgodnie z przewidywaniami złożoność obliczeniowa jest liniowa, jedyną rzeczą która zwraca uwagę jest fakt iż czas wykonania jednej operacji jest dłuższy od czasu wykonania 10 operacji. Dla większej ilości danych wyniki są poprawne. Wydaje się, że zbyt mało danych jest obecnych w środkowej części wykresu co powinno być zostać zmienione w celu poprawy jakości odbioru wykresu (dla charakterystyki liniowej jest to akurat bez znaczenia ale np dla logarytmicznej było by widoczne).

## Rozdział 2

# Indeks hierarchiczny

### 2.1 Hierarchia klas

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

benchmark . . . . .	9
tabx2 . . . . .	11





## Rozdział 3

# Indeks klas

### 3.1 Lista klas

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

<a href="#">benchmark</a>	.....	9
<a href="#">tabx2</a>	.....	11



## Rozdział 4

# Indeks plików

### 4.1 Lista plików

Tutaj znajduje się lista wszystkich plików z ich krótkimi opisami:

<a href="#">benchmark.cpp</a>		
	Deklaracja funkcji z klasy Benchmark . . . . .	13
<a href="#">benchmark.hh</a>		
	Definicja klasy Benchmark . . . . .	13
<a href="#">generator.cpp</a>		
	Deklaracja funkcji generującej liczby losowe . . . . .	14
<a href="#">generator.hh</a>		
	Definicja generatora liczb losowych . . . . .	15
<a href="#">main.cpp</a>		16
<a href="#">tabx2.cpp</a>		
	Deklaracja klasy <a href="#">tabx2</a> . . . . .	17
<a href="#">tabx2.hh</a>		
	Definicja klasy <a href="#">tabx2</a> . . . . .	18



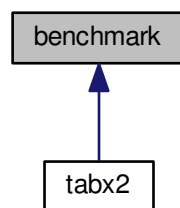
## Rozdział 5

# Dokumentacja klas

### 5.1 Dokumentacja klasy benchmark

```
#include <benchmark.hh>
```

Diagram dziedziczenia dla benchmark



#### Metody publiczne

- void `analyze` (int repeat, int data\_amount)  
*Metoda analyze zlicza czas wykonywania funkcji `test()`*

#### Metody prywatne

- virtual void `test` (int length)=0

#### 5.1.1 Opis szczegółowy

Definicja w linii 11 pliku benchmark.hh.

#### 5.1.2 Dokumentacja funkcji składowych

5.1.2.1 void benchmark::analyze ( int repeat, int data\_amount )

## Parametry

in	<i>repeat</i>	- ilość powtórzeń testu który chcemy wykonać
in	<i>data_amount</i>	- ilość wynikowych danych podawana jako potęga liczby 10

## Zwraca

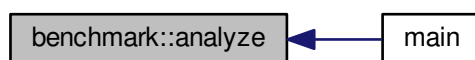
plik .csv z czasami poszczególnych pomiarów oraz ilość testowanych danych

Definicja w linii 7 pliku benchmark.cpp.

Oto graf wywołań dla tej funkcji:



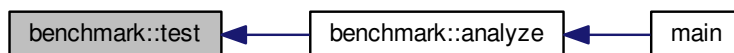
Oto graf wywołań tej funkcji:



#### 5.1.2.2 virtual void benchmark::test ( int *length* ) [private],[pure virtual]

Implementowany w [tabx2](#).

Oto graf wywołań tej funkcji:



Dokumentacja dla tej klasy została wygenerowana z plików:

- [benchmark.hh](#)
- [benchmark.cpp](#)

## 5.2 Dokumentacja klasy tabx2

```
#include <tabx2.hh>
```

Diagram dziedziczenia dla tabx2

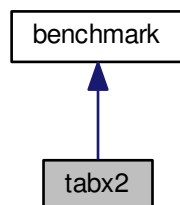
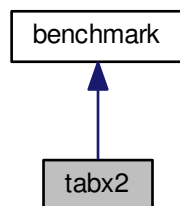


Diagram współpracy dla tabx2:



### Metody publiczne

- void `test` (int length)
- `tabx2` (int size)
- `~tabx2` ()

*zwykły destruktor*

### Atrybuty prywatne

- int `size`  
*Modeluje pojęcie `tabx2`.*

- int \* `tab`  
*Wskaźnik na tablicę przechowująca dane.*

#### 5.2.1 Opis szczegółowy

Definicja w linii 15 pliku tabx2.hh.

## 5.2.2 Dokumentacja konstruktora i destruktora

### 5.2.2.1 `tabx2::tabx2 ( int size )`

Konstruktor wczytujący określoną ilość danych i alokujący je dynamicznie

Parametry

<code>in</code>	<code>size</code>	- długość tablice
-----------------	-------------------	-------------------

Definicja w linii 15 pliku `tabx2.cpp`.

### 5.2.2.2 `tabx2::~~tabx2 ( )`

Definicja w linii 35 pliku `tabx2.cpp`.

## 5.2.3 Dokumentacja funkcji składowych

### 5.2.3.1 `void tabx2::test ( int length ) [virtual]`

Mnoży określoną ilość danych przez 2

Parametry

<code>in</code>	<code>length</code>	- ilość danych do przemnożenia
-----------------	---------------------	--------------------------------

Implementuje [benchmark](#).

Definicja w linii 7 pliku `tabx2.cpp`.

## 5.2.4 Dokumentacja atrybutów składowych

### 5.2.4.1 `int tabx2::size [private]`

Tworzy tablice alokowaną dynamicznie o pojemności wybranej przez użytkownika, umożliwia wykonywanie mnożenia przez 2 wszystkich elementów tablicy

Rozmiar tablicy

Definicja w linii 27 pliku `tabx2.hh`.

### 5.2.4.2 `int* tabx2::tab [private]`

Definicja w linii 30 pliku `tabx2.hh`.

Dokumentacja dla tej klasy została wygenerowana z plików:

- [tabx2.hh](#)
- [tabx2.cpp](#)



## Rozdział 6

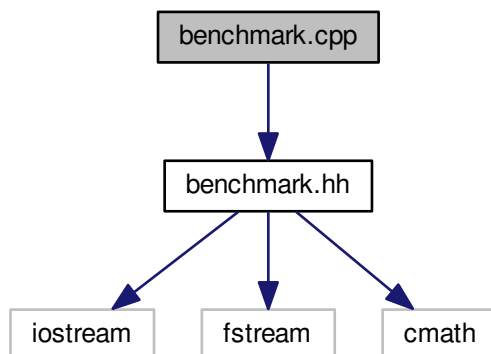
# Dokumentacja plików

### 6.1 Dokumentacja pliku benchmark.cpp

Deklaracja funkcji z klasy Benchmark.

```
#include "benchmark.hh"
```

Wykres zależności załączania dla benchmark.cpp:

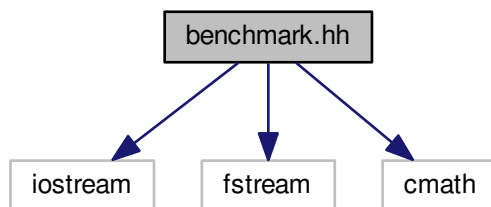


### 6.2 Dokumentacja pliku benchmark.hh

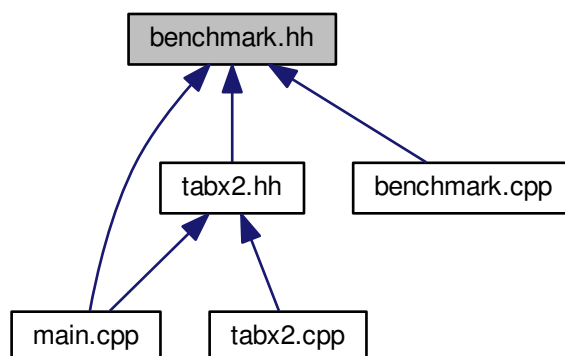
Definicja klasy Benchmark.

```
#include <iostream>
#include <fstream>
#include <cmath>
```

Wykres zależności załączania dla benchmark.hh:

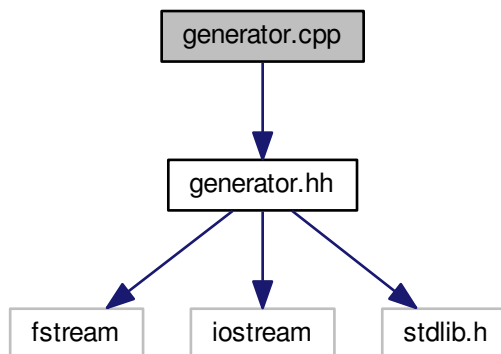


Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



```
#include "generator.hh"
```

Wykres zależności załączania dla generator.cpp:



## Funkcje

- bool `data_generator` (int `data_amount`)

*Generuje liczby losowe.*

### 6.3.1 Dokumentacja funkcji

#### 6.3.1.1 bool `data_generator` ( int `data_amount` )

Funkcja generuje naturalne liczby losowe z przedziału 0-100, które następnie są zapisywane do pliku `random_data.dat`

**Parametry**

<code>in</code>	<code>data_amount</code>	- ilość liczb wynikowych które chcemy uzyskać
-----------------	--------------------------	---

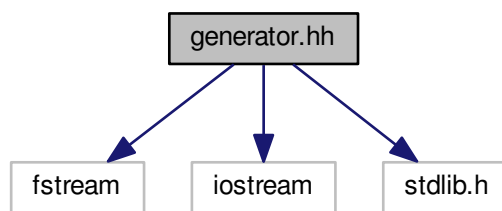
Definicja w linii 9 pliku `generator.cpp`.

## 6.4 Dokumentacja pliku generator.hh

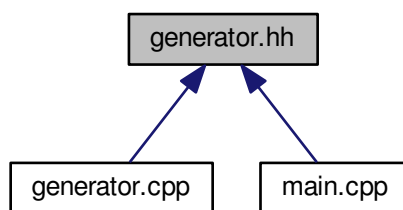
Definicja generatora liczb losowych.

```
#include <fstream>
#include <iostream>
#include <stdlib.h>
```

Wykres zależności załączania dla generator.hh:



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



## Funkcje

- bool `data_generator` (int `data_amount`)  
*Generuje liczby losowe.*

### 6.4.1 Dokumentacja funkcji

#### 6.4.1.1 bool data\_generator ( int data\_amount )

Funkcja generuje naturalne liczby losowe z przedziału 0-100, które następnie są zapisywane do pliku random\_data.dat

Parametry

in	<code>data_amount</code>	- ilość liczb wynikowych które chcemy uzyskać
----	--------------------------	---

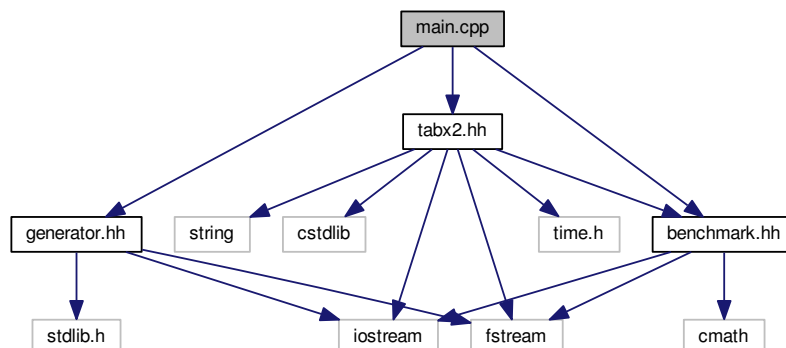
Definicja w linii 9 pliku generator.cpp.

## 6.5 Dokumentacja pliku main.cpp

```
#include "tabx2.hh"
```

```
#include "benchmark.hh"
#include "generator.hh"
```

Wykres zależności załączania dla main.cpp:



## Funkcje

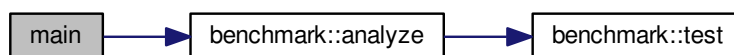
- int `main` ()

### 6.5.1 Dokumentacja funkcji

#### 6.5.1.1 int main ( )

Definicja w linii 6 pliku main.cpp.

Oto graf wywołań dla tej funkcji:



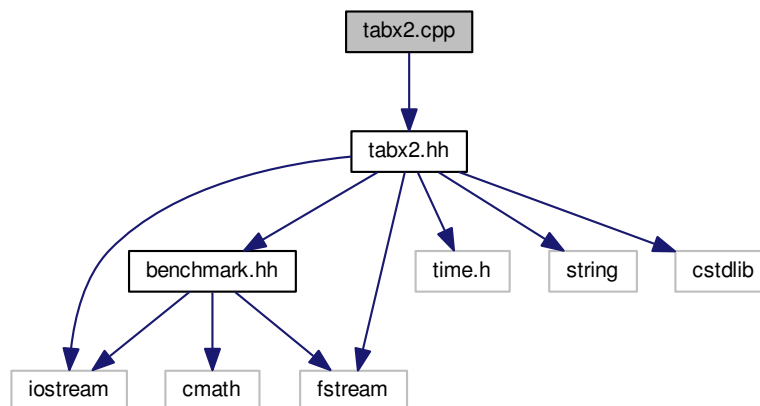
## 6.6 Dokumentacja pliku strona.dox

## 6.7 Dokumentacja pliku tabx2.cpp

Deklaracja klasy `tabx2`.

```
#include "tabx2.hh"
```

Wykres zależności załączania dla tabx2.cpp:

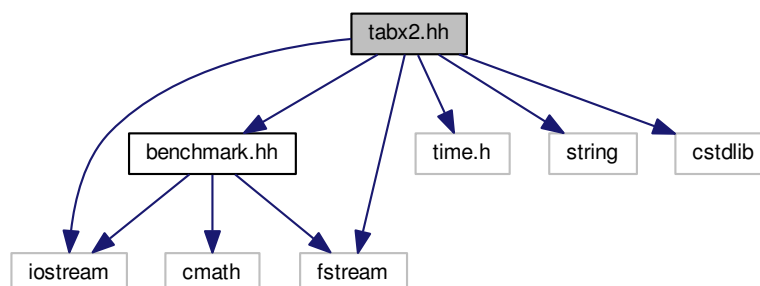


## 6.8 Dokumentacja pliku tabx2.hh

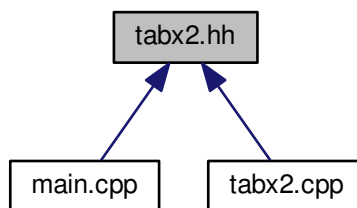
Definicja klasy [tabx2](#).

```
#include <iostream>
#include <time.h>
#include <string>
#include <fstream>
#include <cstdlib>
#include "benchmark.hh"
```

Wykres zależności załączania dla tabx2.hh:



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



### Komponenty

- class `tabx2`

### Definicje

- `#define TABX2_HH`

#### 6.8.1 Opis szczegółowy

Plik zawiera definicję klasy `tabx2`. Modeluje główne pojęcia związane z programem

Definicja w pliku `tabx2.hh`.

#### 6.8.2 Dokumentacja definicji

##### 6.8.2.1 `#define TABX2_HH`

Definicja w linii 2 pliku `tabx2.hh`.

# Skorowidz

- ~tabx2
  - tabx2, [12](#)
- analyze
  - benchmark, [9](#)
- benchmark, [9](#)
  - analyze, [9](#)
  - test, [10](#)
- benchmark.cpp, [13](#)
- benchmark.hh, [13](#)
- data\_generator
  - generator.cpp, [15](#)
  - generator.hh, [16](#)
- generator.cpp, [14](#)
  - data\_generator, [15](#)
- generator.hh, [15](#)
  - data\_generator, [16](#)
- main
  - main.cpp, [17](#)
- main.cpp, [16](#)
  - main, [17](#)
- size
  - tabx2, [12](#)
- strona.dox, [17](#)
- TABX2\_HH
  - tabx2.hh, [19](#)
- tab
  - tabx2, [12](#)
- tabx2, [11](#)
  - ~tabx2, [12](#)
  - size, [12](#)
  - tab, [12](#)
  - tabx2, [12](#)
  - test, [12](#)
- tabx2.cpp, [17](#)
- tabx2.hh, [18](#)
  - TABX2\_HH, [19](#)
- test
  - benchmark, [10](#)
  - tabx2, [12](#)