Podstawowe struktury danych

1

Wygenerowano przez Doxygen 1.8.6

So, 14 mar 2015 15:43:11

Spis treści

1	Spra	wozdar	nie	1
	1.1	Zadani	e	1
	1.2	Wyniki		1
	1.3	Podsur	mowanie	2
2	Inde	ks hiera	archiczny	3
	2.1	Hierard	chia klas	3
3	Inde	ks klas		5
	3.1	Lista k	as	5
4	Inde	ks pliká	bw	7
	4.1	Lista p	lików	7
5	Dok	umenta	cja klas	9
	5.1	Dokum	entacja klasy benchmark	9
		5.1.1	Opis szczegółowy	9
		5.1.2	Dokumentacja funkcji składowych	10
			5.1.2.1 analyze	10
			5.1.2.2 test	11
	5.2	Dokum	entacja klasy tabx2	12
		5.2.1	Opis szczegółowy	13
		5.2.2	Dokumentacja konstruktora i destruktora	13
			5.2.2.1 tabx2	13
			5.2.2.2 ~tabx2	13
		5.2.3	Dokumentacja funkcji składowych	13
			5.2.3.1 test	13
		5.2.4	Dokumentacja atrybutów składowych	13
			5.2.4.1 size	13
			5.2.4.2 tab	13
6	Dok	umenta	cja plików	15
	6.1	Dokum	entacja pliku benchmark.cpp	15

iv SPIS TREŚCI

6.2	Dokum	mentacja pliku benchmark.hh	15
6.3	Dokum	mentacja pliku generator.cpp	16
	6.3.1	Dokumentacja funkcji	17
		6.3.1.1 data_generator	17
6.4	Dokum	mentacja pliku generator.hh	17
	6.4.1	Dokumentacja funkcji	18
		6.4.1.1 data_generator	18
6.5	Dokum	mentacja pliku main.cpp	18
	6.5.1	Dokumentacja funkcji	19
		6.5.1.1 main	19
6.6	Dokum	mentacja pliku strona.dox	19
6.7	Dokum	mentacja pliku tabx2.cpp	19
6.8	Dokum	mentacja pliku tabx2.hh	20
	6.8.1	Opis szczegółowy	21
	6.8.2	Dokumentacja definicji	21
		6.8.2.1 TABX2_HH	21
Indeks			22

Sprawozdanie

Data

11.03.2015r.

Wersja

0.1

1.1 Zadanie

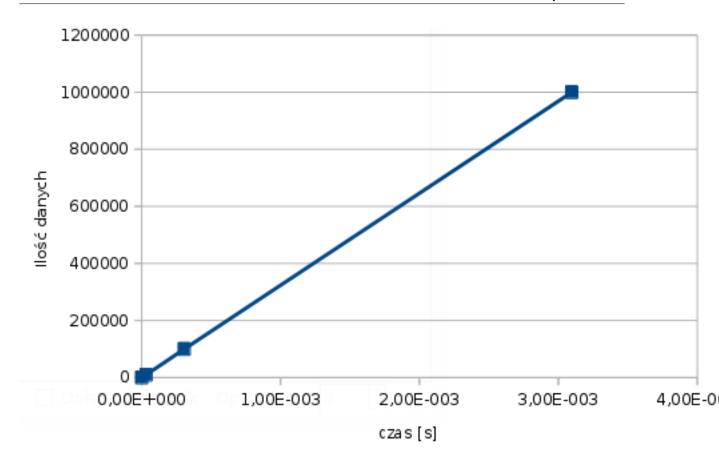
Celem ćwiczenia było stworzenie programu benchmarkującego , który dla wybranych danych będzie zliczał średni czas wykonania dowolnego algorytmu (w tym przypadku mnożenia elementów tablicy przez 2). Należało również stworzyć program generujący losowe liczby.

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 maxymalną liczbą danych dla jakiej przeprowadzany był test

Na podstawie otrzymanych danych mamy :

2 Sprawozdanie



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 wiekszej ilości danych wyniki są poprawne Wydaje się , że zbyt mało danych jest obecnych w środkowej częsci 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).

Indeks hierarchiczny

2.1 Hierarchia klas

ıaı	ista dziedziczei	nıa	ı po	os	or	tov	иa	na	. je	≀st	Z	gr	ub	SZ	a,	cr	100	n	ıe	ca	lłKC	WI	CIE	9, 8	alta	be	ety	CZI	nie	:						
	benchmark																																			9
	tabx2																																			12

4	Indeks hierarchiczny

Indeks klas

\sim	4	 5 - A -	11.1	l
3	п.	 ista	- I/-	20

ıtaj znajdują się klasy, struktury, unie i interfejsy wraz z ich krótkimi opisami:	
benchmark	 ,
tabx2	 12

6 Indeks klas

Indeks plików

4.1 Lista plików

Tutaj znajduje się lista wszystkich plików z ich krótkimi opisami:

benchmark.cpp	
Deklaracja funkzji z klasy Benchmark	15
benchmark.hh	
Definicja klasy Benchmark	15
generator.cpp	
Deklaracja funckji generujacej liczby losowe	16
generator.hh	
Definicja generatora liczb losowych	17
main.cpp	18
tabx2.cpp	
Deklaracja klasy main	19
tabx2.hh	
Definicja klasy tabx2	2(

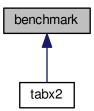
Indeks plików 8

Dokumentacja klas

5.1 Dokumentacja klasy benchmark

#include <benchmark.hh>

Diagram dziedziczenia dla benchmark



Metody publiczne

• void analyze (int repeat, int data_amount)

Metoda pozwala przeprowadzić analize czasu trwania funkcji test w następujący sposób : wykonuje funckję test repeat razy , po czym uśrednia jej wynik. Cały proces jest wykonywany dla data_amount razy , gdzie data amount jest potęgą liczby 10.

Metody prywatne

• virtual void test (int length)=0

Klasa benchmark umożliwia przprowadzenie pomiaru czasu trwania programu dla określonych danych.

5.1.1 Opis szczegółowy

Definicja w linii 11 pliku benchmark.hh.

10 Dokumentacja klas

5.1.2 Dokumentacja funkcji składo	lowych	skład	funkcii	Dokumentacia	5.1.2
-----------------------------------	--------	-------	---------	--------------	-------

5.1.2.1 void benchmark::analyze (int repeat, int data_amount)

Parametry

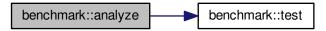
in	repeat	- ilość powtórzeń testu który chcemy wykonać
in	data_amount	- ilosc wynikowych danych podawana jako potega liczby 10

Zwraca

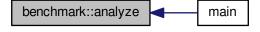
plik .csv z czasami poszczegolnych pomiarow oraz ilosc testowanych danych

Definicja w linii 7 pliku benchmark.cpp.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



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

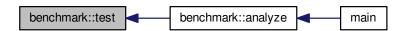
Metoda wirtualna, reprezentuje czynność której czas chcemy zmierzyc

Parametry

ir	1	length	- ilosc danych dla jakich przeprowadzamy test

Implementowany w tabx2.

Oto graf wywoływań tej funkcji:



Dokumentacja dla tej klasy została wygenerowana z plików:

12 Dokumentacja klas

- benchmark.hh
- benchmark.cpp

5.2 Dokumentacja klasy tabx2

#include <tabx2.hh>

Diagram dziedziczenia dla tabx2

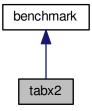
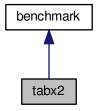


Diagram współpracy dla tabx2:



Metody publiczne

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

zwykly destruktor

Atrybuty prywatne

· int size

Modeluje pojęcie tabx2.

int * tab

Wskaznik na tablice przechowujaca 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 wczytujacy okreslona ilosc danych i alokujacy je dynamicznie

Parametry

in	size	- dlugosc tablice
----	------	-------------------

Definicja w linii 15 pliku tabx2.cpp.

5.2.2.2 tabx2:: \sim tabx2 ()

Definicja w linii 35 pliku tabx2.cpp.

5.2.3 Dokumentacja funkcji składowych

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

Mnozy okreslona ilosc danych przez 2

Parametry

in	length	- ilosc danych do przemnozenia
----	--------	--------------------------------

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 alokowana dynamicznie o pojemnosci wybranej przez uzytkownika , umozliwia wykonywanie mnozenia przez 2 wszystkich elementow 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

14 Dokumentacja klas

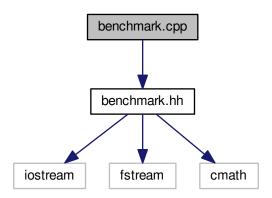
Dokumentacja plików

6.1 Dokumentacja pliku benchmark.cpp

Deklaracja funkzji z klasy Benchmark.

#include "benchmark.hh"

Wykres zależności załączania dla benchmark.cpp:



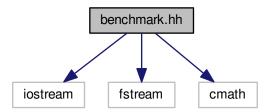
6.2 Dokumentacja pliku benchmark.hh

Definicja klasy Benchmark.

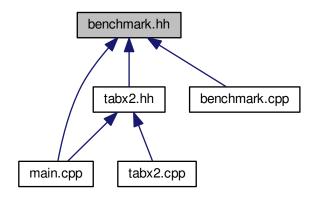
```
#include <iostream>
#include <fstream>
#include <cmath>
```

16 Dokumentacja plików

Wykres zależności załączania dla benchmark.hh:



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



Komponenty

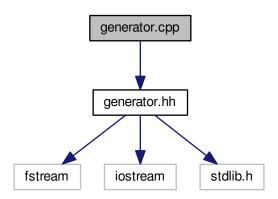
· class benchmark

6.3 Dokumentacja pliku generator.cpp

Deklaracja funckji generujacej liczby losowe.

#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, ktore nastepnie sa zapisywane do pliku random_-data.dat

Parametry

in	data_amount	- ilosc liczb wynikowych ktore chcemy uzyskac

Definicja w linii 9 pliku generator.cpp.

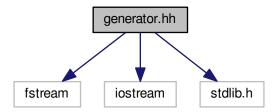
6.4 Dokumentacja pliku generator.hh

Definicja generatora liczb losowych.

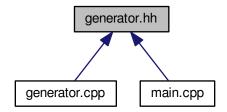
```
#include <fstream>
#include <iostream>
#include <stdlib.h>
```

18 Dokumentacja plików

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, ktore nastepnie sa zapisywane do pliku random_-data.dat

Parametry

in	data_amount	- ilosc liczb wynikowych ktore chcemy uzyskac

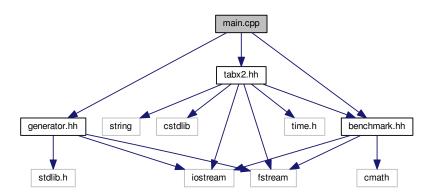
Definicja w linii 9 pliku generator.cpp.

6.5 Dokumentacja pliku main.cpp

#include "tabx2.hh"

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

Wykres zależności załączania dla main.cpp:



Funkcje

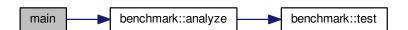
• int main ()

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

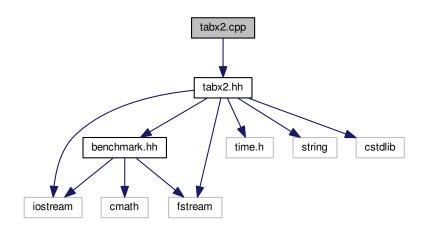
6.7 Dokumentacja pliku tabx2.cpp

Deklaracja klasy main.

20 Dokumentacja plików

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

Wykres zależności załączania dla tabx2.cpp:

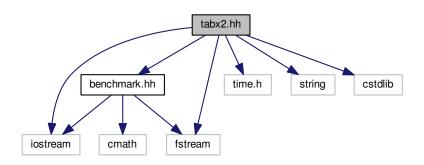


6.8 Dokumentacja pliku tabx2.hh

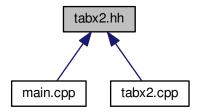
Definicja klasy tabx2.

```
#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 glowne pojecia zwiazane 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

```
\simtabx2
    tabx2, 13
analyze
    benchmark, 10
benchmark, 9
    analyze, 10
    test, 11
benchmark.cpp, 15
benchmark.hh, 15
data_generator
    generator.cpp, 17
    generator.hh, 18
generator.cpp, 16
    data_generator, 17
generator.hh, 17
    data_generator, 18
main
     main.cpp, 19
main.cpp, 18
    main, 19
size
    tabx2, 13
strona.dox, 19
TABX2_HH
    tabx2.hh, 21
tab
    tabx2, 13
tabx2, 12
    \simtabx2, 13
    size, 13
    tab, 13
    tabx2, 13
    test, 13
tabx2.cpp, 19
tabx2.hh, 20
    TABX2_HH, 21
test
    benchmark, 11
    tabx2, 13
```