lab1

0.0001

Wygenerowano przez Doxygen 1.8.6

Śr, 11 mar 2015 22:44:32

Spis treści

Rozdział 1

lab1

Autor

Filip Malinowski

Program sluzacy do uruchamiania algorytmow i badania ich szybkosci dzialania.

Format wywolania:

./prj/make clean

./prj/make

Rozdział 2

Indeks hierarchiczny

2.1 Hierarchia klas

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

Rozdział 3

Indeks klas

3.1 Lista klas

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

Rozdział 4

Indeks plików

4.1 Lista plików

Tutaj znajduje się lista wszystkich plików z ich krótkimi opisami:

Rozdział 5

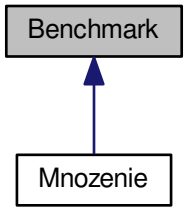
Dokumentacja klas

5.1 Dokumentacja klasy Benchmark

Klasa Benchmark modelujaca program benchmarkujacy. Obiekt tego typu reprezentuje program sprawdzajacy szybkosc wykonywania algorytmow.

`#include <benchmark.hh>`

Diagram dziedziczenia dla Benchmark



Metody publiczne

- Benchmark ()

  Konstrukor obiektu Benchmark.
- Benchmark (int _repeats)

  Konstruktor parametryczny obiektu Benchmark.
- ∼Benchmark ()

  Destruktor obiektu Benchmark.
- virtual void testAlgorithm (Benchmark ∗_algorithm) const

  Metoda testowania algorytmu. Metoda sluzy to testowania szybkosci dzialania algorytmu. Wykonuje testowany algorytm dla 5 kolejnych ilosci elementow. Wykonanie algorytmu dla danego zestawu liczb powtarza dwa razy i usrednia wynik. Otrzymany czas wraz z iloscia testowanych danych zapisuje w pliku ret_data.txt.
- virtual void runAlgorithm (int _border)

  Metoda uruchamiania algorytmu. Metoda sluzy do wykonywania danego algorytmu. W klasie Benchmark nie ma konkretnego dzialania.

Atrybuty prywatne

- int repeats

5.1.1 Opis szczegółowy

Definicja w linii 11 pliku benchmark.hh.

5.1.2 Dokumentacja konstruktora i destruktora

5.1.2.1 Benchmark::Benchmark ( ) [inline]

Definicja w linii 20 pliku benchmark.hh.

5.1.2.2 Benchmark::Benchmark (int *_repeats*) [inline]

Parametry

in	*_repeats*	- ilosc powtorzen testu dla pojedynczego zestawu danych.

Definicja w linii 26 pliku benchmark.hh.

5.1.2.3 Benchmark::~Benchmark ( ) [inline]

Definicja w linii 31 pliku benchmark.hh.

5.1.3 Dokumentacja funkcji składowych

5.1.3.1 virtual void Benchmark::runAlgorithm (int *_border*) [inline], [virtual]

Parametry

in	*_border*	- ilosc elementow dla ktorych algorytm ma wykonac swoje dzialanie.

Reimplementowana w Mnozenie.

Definicja w linii 50 pliku benchmark.hh.

Oto graf wywoływań tej funkcji:

5.1.3.2 void Benchmark::testAlgorithm (Benchmark * *_algorithm*) const [virtual]

Parametry

in	*_algorithm*	- testowany algorytm.

Definicja w linii 10 pliku benchmark.cpp.

Oto graf wywołań dla tej funkcji:

5.1.4 Dokumentacja atrybutów składowych

5.1.4.1 int Benchmark::repeats [private]

Definicja w linii 13 pliku benchmark.hh.

Dokumentacja dla tej klasy została wygenerowana z plików:

- benchmark.hh
- benchmark.cpp

5.2 Dokumentacja klasy Mnozenie

Klasa Mnozenie modelujaca algorytm potegowania. Obiekt tego typu reprezentuje algorytm wykonujacy dzialanie mnozenia kazdego elementu tablicy tab przez 2.

`#include <algorithm1.hh>`

Diagram dziedziczenia dla Mnozenie

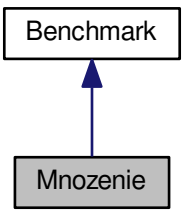
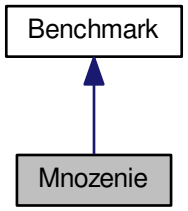


Diagram współpracy dla Mnozenie:



Metody publiczne

- Mnozenie ()

  Konstruktor obiektu Mnozenie.
- Mnozenie (int _tab[SIZE])

  Konstruktor parametryczny obiektu Mnozenie.
- ~Mnozenie ()

  Destruktor obiektu Mnozenie.
- virtual void testAlgorithm (Benchmark *_algorithm)

  Metoda testowania algorytmu. Metoda sluzy do testowania szybkosci dzialania algorytmu. W klasie Mnozenie nie ma konkretnego dzialania.
- virtual void runAlgorithm (int _border)

  Metoda uruchamiania algorytmu. Metoda sluzy to wykonywania danego algorytmu. Mnozy kazdy element tablicy przez liczbe 2.

Atrybuty prywatne

- int tab [SIZE]

5.2.1 Opis szczegółowy

Definicja w linii 9 pliku algorithm1.hh.

5.2.2 Dokumentacja konstruktora i destruktora

5.2.2.1 Mnozenie::Mnozenie ( ) [inline]

Definicja w linii 17 pliku algorithm1.hh.

5.2.2.2 Mnozenie::Mnozenie (int *_tab[SIZE]*) [inline]

Parametry

in	*_tab*	- tablica przechowujaca dane wejsciowe.

Definicja w linii 23 pliku algorithm1.hh.

5.2.2.3 Mnozenie::~Mnozenie ( ) [inline]

Definicja w linii 28 pliku algorithm1.hh.

5.2.3 Dokumentacja funkcji składowych

5.2.3.1 void Mnozenie::runAlgorithm (int *_border*) [virtual]

Parametry

in	*_border*	- ilosc elementow dla ktorych algorytm ma wykonac swoje dzialanie.

Reimplementowana z Benchmark.

Definicja w linii 7 pliku algorithm1.cpp.

5.2.3.2 virtual void Mnozenie::testAlgorithm (Benchmark * *_algorithm*) [inline], [virtual]

Parametry

in	*_algorithm*	- testowany algorytm.

Definicja w linii 36 pliku algorithm1.hh.

5.2.4 Dokumentacja atrybutów składowych

5.2.4.1 int Mnozenie::tab[SIZE] [private]

Definicja w linii 11 pliku algorithm1.hh.

Dokumentacja dla tej klasy została wygenerowana z plików:

- algorithm1.hh
- algorithm1.cpp

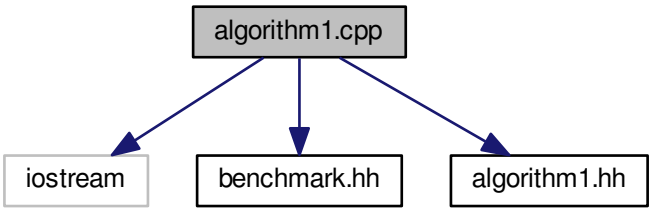
Rozdział 6

Dokumentacja plików

6.1 Dokumentacja pliku algorithm1.cpp

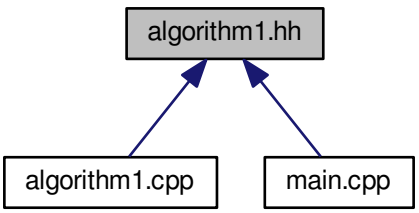
```
#include <iostream>
#include "benchmark.hh"
#include "algorithm1.hh"
```

Wykres zależności załączania dla algorithm1.cpp:



6.2 Dokumentacja pliku algorithm1.hh

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



Komponenty

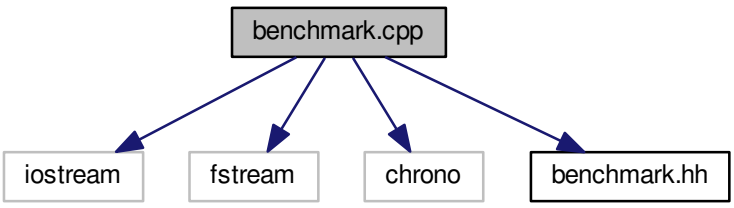
- class Mnozenie

  Klasa Mnozenie modelujaca algorytm potegowania. Obiekt tego typu reprezentuje algorytm wykonujacy dzialanie mnozenia kazdego elementu tablicy tab przez 2.

6.3 Dokumentacja pliku benchmark.cpp

```
#include <iostream>
#include <fstream>
#include <chrono>
#include "benchmark.hh"
```

Wykres zależności załączania dla benchmark.cpp:



Definicje

- #define LENGTH 5

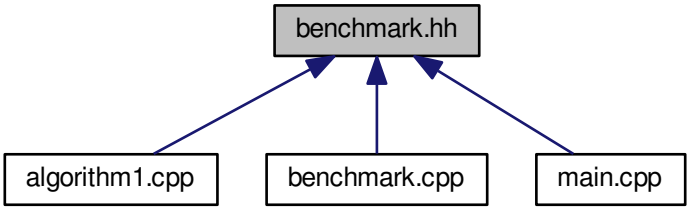
6.3.1 Dokumentacja definicji

6.3.1.1 #define LENGTH 5

Definicja w linii 7 pliku benchmark.cpp.

6.4 Dokumentacja pliku benchmark.hh

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



Komponenty

- class Benchmark

  Klasa Benchmark modelujaca program benchmarkujacy. Obiekt tego typu reprezentuje program sprawdzajacy szybkosc wykonywania algorytmow.

Definicje

- #define SIZE 10000000

6.4.1 Dokumentacja definicji

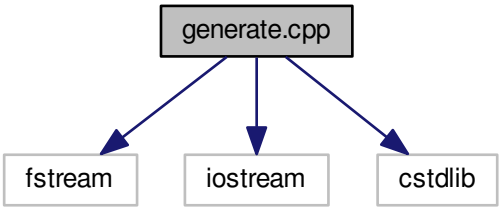
6.4.1.1 #define SIZE 10000000

Definicja w linii 4 pliku benchmark.hh.

6.5 Dokumentacja pliku generate.cpp

```
#include <fstream>
#include <iostream>
#include <cstdlib>
```

Wykres zależności załączania dla generate.cpp:



Definicje

- #define SIZE 10000000

Funkcje

- int main ()

  Funkcja generowania pliku z danymi wejsciowymi. Generuje liczby losowe od 1 do 51 i zapisuje je do pliku o nazwie data.txt.

6.5.1 Dokumentacja definicji

6.5.1.1 #define SIZE 10000000

Definicja w linii 5 pliku generate.cpp.

6.5.2 Dokumentacja funkcji

6.5.2.1 int main ( )

Zwracane wartości

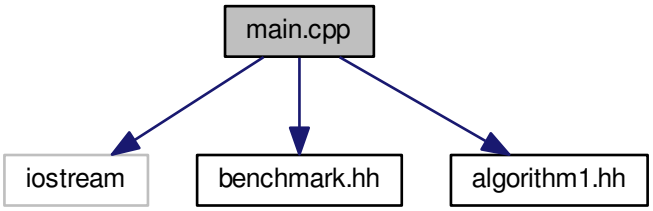
0	- gdy funkcja zadziala poprawnie.
1	- gdy wystapi blad otwarcia pliku do zapisu.

Definicja w linii 14 pliku generate.cpp.

6.6 Dokumentacja pliku main.cpp

```
#include <iostream>
#include "benchmark.hh"
#include "algorithm1.hh"
```

Wykres zależności załączania dla main.cpp:



Funkcje

- int main ()

  Funkcja tworzaca i testujaca algorytm. Wczytuje dane otrzymane na strumien wejsciowy do tablicy data[]. Nastepnie tworzy obiekt Benchmark oraz obiekt Potegowanie. Pozniej uruchamia metode testujaca w obiekcie klasy Benchmark dla obiektu klasy Potegowanie.

6.6.1 Dokumentacja funkcji

6.6.1.1 int main ( )

Zwracane wartości

0	- domyslna wartosc zwracana przez funkcje.

Definicja w linii 14 pliku main.cpp.

6.7 Dokumentacja pliku strona-glowna.dox

Skorowidz