Laboratorium 1

0.1

Wygenerowano przez Doxygen 1.8.6

Cz, 12 mar 2015 04:43:18

Rozdział 1

Laboratorium 1

Aplikacja umozliwia uzytkownikowi na przeprowadzenia algorytmu mnozenia

przez dwa na dowolnej liczbie elementow.

Najważniejsze cechy

Możliwość włączenia opcji benchmarkującej służącej do sprawdzenia

ile czasu wykonywal sie dany algorytm lub seria tego samego algorytmu

Argumenty wywołania

```
-n liczba       Ilość liczb do odczytania/przerobienia przez algorytm
-t liczba       Włącza opcje benchmarkującą dla seri powtorzen
-o tekst        Wprowadza nazwe pliku do zapisu
-i tekst        Wprowadza nazwe pliku do odczytu
-g              Generuje n liczb i zapisuje je do pliku (po wygenerowaniu konczy program)
```

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:

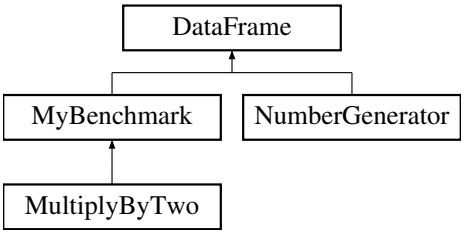
Rozdział 5

Dokumentacja klas

5.1 Dokumentacja klasy DataFrame

`#include <dataframe.h>`

Diagram dziedziczenia dla DataFrame



Metody publiczne

- int loadDataFromFile ()

  Ładuje dane z pliku.
- int saveDataToFile ()

  Zapisuje dane do pliku.
- DataFrame operator= (DataFrame dataframe)

  Kopiuje elementy roznych obiektow.
- virtual ~DataFrame ()

Atrybuty publiczne

- int * tableOfData

  Zawiera adres do tablicy {size} elementów.
- char * outputFileName

  Zawiera nazwe pliku do zapisu.
- char * inputFileName

  Zawiera nazwe pliku do odczytu.
- unsigned int sizeOfTable

  Rozmiar tablicy tableOfData.

5.1.1 Opis szczegółowy

Definicja w linii 15 pliku dataframe.h.

5.1.2 Dokumentacja konstruktora i destruktora

5.1.2.1 virtual DataFrame::~DataFrame ( ) [inline], [virtual]

Definicja w linii 60 pliku dataframe.h.

5.1.3 Dokumentacja funkcji składowych

5.1.3.1 int DataFrame::loadDataFromFile ( )

Wczytuje dane z pliku i zapisuje je dynamicznie do tablicy jednowymiarowej, na ktora wskazuje wskaźnik ∗tableOfData

Rozmiar tablicy jest przechowywany w sizeOfTable

Definicja w linii 13 pliku dataframe.cpp.

5.1.3.2 DataFrame DataFrame::operator= (DataFrame *dataframe*)

Zapisuje kolejne liczby do pliku o nazwie outputFileName

Definicja w linii 37 pliku dataframe.cpp.

5.1.3.3 int DataFrame::saveDataToFile ( )

Wczytuje liczby z pliku o nazwie intputFileName

Definicja w linii 25 pliku dataframe.cpp.

5.1.4 Dokumentacja atrybutów składowych

5.1.4.1 char∗ DataFrame::inputFileName

Definicja w linii 29 pliku dataframe.h.

5.1.4.2 char∗ DataFrame::outputFileName

Definicja w linii 25 pliku dataframe.h.

5.1.4.3 unsigned int DataFrame::sizeOfTable

Definicja w linii 34 pliku dataframe.h.

5.1.4.4 int∗ DataFrame::tableOfData

Definicja w linii 21 pliku dataframe.h.

Dokumentacja dla tej klasy została wygenerowana z plików:

- dataframe.h

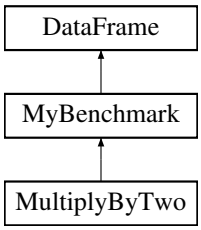
- dataframe.cpp

5.2 Dokumentacja klasy MultiplyByTwo

Algorytm mnozy kazda liczbe razy 2.

`#include <multiplybytwo.h>`

Diagram dziedziczenia dla MultiplyByTwo



Metody publiczne

- void executeAlgorithm ()

  Wykonuje algorytm mnozenie x2.
- ~MultiplyByTwo ()

Dodatkowe Dziedziczone Składowe

5.2.1 Opis szczegółowy

Algorytm mnozy kazda kolejna liczbe przez 2

Definicja w linii 20 pliku multiplybytwo.h.

5.2.2 Dokumentacja konstruktora i destruktora

5.2.2.1 MultiplyByTwo::~MultiplyByTwo ( ) [inline]

Definicja w linii 29 pliku multiplybytwo.h.

5.2.3 Dokumentacja funkcji składowych

5.2.3.1 void MultiplyByTwo::executeAlgorithm ( ) [virtual]

Implementuje MyBenchmark.

Definicja w linii 11 pliku multiplybytwo.cpp.

Dokumentacja dla tej klasy została wygenerowana z plików:

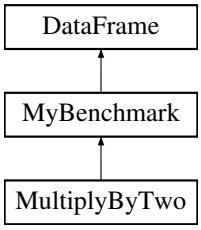
- multiplybytwo.h
- multiplybytwo.cpp

5.3 Dokumentacja klasy MyBenchmark

Klasa bazowa/interface do testowania algorytmu.

`#include <mybenchmark.h>`

Diagram dziedziczenia dla MyBenchmark



Metody publiczne

- double testAlgorithm (unsigned int repetition)

  Benchmarkuje algorytm główny.
- virtual ~MyBenchmark ()

  Usuwam obiekt test biorąc pod uwage jego prawdziwy typ.

Metody chronione

- virtual void executeAlgorithm ()=0

  Interface metody algorytmu glownego.

Dodatkowe Dziedziczone Składowe

5.3.1 Opis szczegółowy

Używana jako interface dla wszystkich algorytmow aby testowac czas wykonywanego algorymtu.

Definicja w linii 20 pliku mybenchmark.h.

5.3.2 Dokumentacja konstruktora i destruktora

5.3.2.1 virtual MyBenchmark::~MyBenchmark ( ) [inline], [virtual]

Definicja w linii 49 pliku mybenchmark.h.

5.3.3 Dokumentacja funkcji składowych

5.3.3.1 virtual void MyBenchmark::executeAlgorithm ( ) [protected], [pure virtual]

Metoda abstrakcyjna, ktora jest interfacem do implementacji przez glowny algorytm. To znaczy, ze kazdy algorytm ma byc uruchamiany tą funkcja

Implementowany w MultiplyByTwo.

5.3.3.2 double MyBenchmark::testAlgorithm (unsigned int *repetition*)

Obliczam czas wykonywanego algorytmu dzięki zastosowaniu metody abstrakcyjnej executeAlgorithm() i zaimplementowaniu tego interfacu w algorytmie głównym

Definicja w linii 12 pliku mybenchmark.cpp.

Dokumentacja dla tej klasy została wygenerowana z plików:

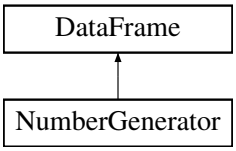
- mybenchmark.h
- mybenchmark.cpp

5.4 Dokumentacja klasy NumberGenerator

Klasa generujaca losowe liczby.

`#include <numbergenerator.h>`

Diagram dziedziczenia dla NumberGenerator



Metody publiczne

- void generateNumbers ()
  Generuje losowe liczby.
- ~NumberGenerator ()

Dodatkowe Dziedziczone Składowe

5.4.1 Opis szczegółowy

Klasa generujaca losowe liczby na podstawie czasu maszyny na ktorym jest uruchomiona Wszystkie funkcje zapisu pliku dziedziczy z klasy DataFrame

Definicja w linii 23 pliku numbergenerator.h.

5.4.2 Dokumentacja konstruktora i destruktora

5.4.2.1 NumberGenerator::~NumberGenerator ( ) [inline]

Definicja w linii 44 pliku numbergenerator.h.

5.4.3 Dokumentacja funkcji składowych

5.4.3.1 void NumberGenerator::generateNumbers ( ) [inline]

Generuje losowe liczby na podstawie czasu maszyny

Definicja w linii 31 pliku numbergenerator.h.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- numbergenerator.h

Rozdział 6

Dokumentacja plików

6.1 Dokumentacja pliku dataframe.cpp

```
#include "dataframe.h"
```

6.2 Dokumentacja pliku dataframe.d

6.3 Dokumentacja pliku dataframe.h

```
#include <fstream>
```

Komponenty

- class DataFrame

6.4 Dokumentacja pliku main.cpp

```
#include <iostream>
#include <unistd.h>
#include <stdlib.h>
#include "multiplybytwo.h"
#include "numbergenerator.h"
#include "dataframe.h"
```

Funkcje

- int main (int argc, char ∗argv[])

6.4.1 Dokumentacja funkcji

6.4.1.1 int main (int *argc,* char ∗ *argv[]*)

Ilosc powtorzen przez algorytmu

Zmienna uzywana przez GETOPT

Flaga ktora mowi o tym czy wlaczyc generator liczb losowych

Definicja w linii 16 pliku main.cpp.

6.5 Dokumentacja pliku main.d

6.6 Dokumentacja pliku multiplybytwo.cpp

```
#include "multiplybytwo.h"
```

6.7 Dokumentacja pliku multiplybytwo.d

6.8 Dokumentacja pliku multiplybytwo.h

```
#include "mybenchmark.h"
#include "dataframe.h"
```

Komponenty

- class MultiplyByTwo

  Algorytm mnozy kazda liczbe razy 2.

6.9 Dokumentacja pliku mybenchmark.cpp

```
#include "mybenchmark.h"
```

6.10 Dokumentacja pliku mybenchmark.d

6.11 Dokumentacja pliku mybenchmark.h

```
#include <ctime>
#include "dataframe.h"
```

Komponenty

- class MyBenchmark

  Klasa bazowa/interface do testowania algorytmu.

6.12 Dokumentacja pliku numbergenerator.h

```
#include <stdlib.h>
#include <time.h>
#include <iostream>
#include "dataframe.h"
```

Komponenty

- class NumberGenerator

  Klasa generujaca losowe liczby.

6.13 Dokumentacja pliku strona-glowna.dox