

# Tester algorytmow

## 1.2

Generated by Doxygen 1.8.1.2

Sun Mar 9 2014 22:42:50



# Contents

<b>1</b>	<b>Tester algorytmow</b>	<b>1</b>
<b>2</b>	<b>Class Index</b>	<b>3</b>
2.1	Class List . . . . .	3
<b>3</b>	<b>Class Documentation</b>	<b>5</b>
3.1	Dzialanie Class Reference . . . . .	5
3.1.1	Detailed Description . . . . .	5
3.1.2	Member Function Documentation . . . . .	5
3.1.2.1	sprawdz . . . . .	5
3.1.2.2	wczytajDaneWejscowe . . . . .	6
3.1.2.3	wez_rozmiar . . . . .	6
3.1.2.4	wlaczStoper . . . . .	6
3.1.2.5	wykonajAlgorytm . . . . .	6
3.1.2.6	wylaczStoper . . . . .	6
3.2	Kontener Class Reference . . . . .	7
3.2.1	Detailed Description . . . . .	7
3.2.2	Member Function Documentation . . . . .	8
3.2.2.1	dodaj_element . . . . .	8
3.2.2.2	dodaj_elementy . . . . .	8
3.2.2.3	operator+ . . . . .	8
3.2.2.4	operator= . . . . .	8
3.2.2.5	operator== . . . . .	8
3.2.2.6	operator[] . . . . .	9
3.2.2.7	wczytajDane . . . . .	9
3.2.2.8	wez_dane . . . . .	9
3.2.2.9	wez_rozmiar . . . . .	9
3.2.2.10	zamien_elementy . . . . .	9
3.3	Tester Class Reference . . . . .	10
3.3.1	Detailed Description . . . . .	10
3.3.2	Member Function Documentation . . . . .	10
3.3.2.1	zamienNazwy . . . . .	10



# Chapter 1

## Tester algorytmow

### Author

Mariusz Dajczak 200403

### Date

02.03.2014

### Version

1.2

Aplikacja wykonuje test algorytmu pod wzgledem czasu dzialania dla roznych danych wejsciowych. Nalezy dostarczyc conajmniej jeden zestaw danych aby program dzialal.

Zestaw danych to:

- plik z danymi wejsciowymi, ktorego format jest scisle okreslony:

- nazwa: wejście%.txt, gdzie % to numer zestawu

- plik z wynikiem:

- nazwa: wynik%.txt, gdzie % to numer zestawu

Zestawy nalezy numerowac od 0.

Rozmiar problemu dla danego zestawu musi byc identyczny.

Wynikiem dzialania programu jest plik CSV, w ktorym zapisany jest rezultat testow.

Plik nosi nazwe rezultat.csv

Jego format wyglada nastepujaca:

- w kazdym wierszu znajduja sie takie dane:

- rozmiar\_problemu,ilosc\_powtorzen,sredni\_czas



## Chapter 2

# Class Index

### 2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">Dzialanie</a>	Klasa modelujaca gowna czesc programu . . . . .	5
<a href="#">Kontener</a>	Klasa Dane . . . . .	7
<a href="#">Tester</a>	Klasa <a href="#">Tester</a> Klasa modeluje narzedzie do benchmarkowania algorytmu. Poprzez wielokrotne wynonywanie algorytmu mozemy wyciagnac sredni czas dzialania, co jest dokladniejszym pomiarem . . . . .	10





## Chapter 3

# Class Documentation

### 3.1 Działanie Class Reference

Klasa modelująca gówna czesc programu.

```
#include <dzialanie.hh>
```

#### Public Member Functions

- void [wczytajDaneWejscowe](#) (string nazwa)  
*metoda wczytuje dane do tablicy znajdujacej sie w zmiennej wejscie.*
- void [wczytajWynik](#) (string nazwa)
- LARGE\_INTEGER [włączStoper](#) ()  
*Metoda uruchamia pomiar czasu.*
- LARGE\_INTEGER [wylaczStoper](#) ()  
*Metoda konczy pomiar czasu.*
- void [wykonajAlgorytm](#) ()  
*Metoda wykonuje algorytm na danych wejscowych ( tablicy)*
- bool [sprawdz](#) ()  
*Metoda sprawdza poprawnosc algorytmu.*
- int [uruchom](#) ()  
*Metoda wykonuje jednorazowy test algorytmu Metoda: -włącza zegar -wykonuje algorytm -wylacza zegar -sprawdza poprawnosc algorytmu return czas wykonywania algorytmu w milisekundach.*
- unsigned int [wez\\_rozmiar](#) ()  
*Metoda zwracajaca rozmiar problemu.*

#### 3.1.1 Detailed Description

Klasa modelująca gówna czesc programu.

Klasa modeluje gówna czesc porgramu, ktorego zadaniem jest: -wczytanie danych -zmierzenie czasu dzialania algorytmu -sprawdzenie poprawnosc tego algorytmu, majac oczekiwany wynik

#### 3.1.2 Member Function Documentation

##### 3.1.2.1 bool Działanie::sprawdz ( )

Metoda sprawdza poprawnosc algorytmu.

Wczytywane sa poprawne dane wynikowe, a nastepnie sa one porownywane z tymi otrzymanymi przez wykonanie algorytmu

#### Returns

0 - gdy algorytm jest poprawny, -1 - gdy nie.

#### 3.1.2.2 void Dzialanie::wczytajDaneWejsciove ( string nazwa )

metoda wczytuje dane do tablicy znajdujacej sie w zmiennej wejscie.

Format danych w pliku jest nastepujacy: pierwszy wiersz - ilosc elementow, a nastepnie w kolumnie kolejne wartosci tablicy.

#### Parameters

<i>nazwaPliku</i>	- nazwa pliku do otwarcia
-------------------	---------------------------

#### Returns

void

#### 3.1.2.3 unsigned int Dzialanie::wez\_rozmiar ( ) [inline]

Metoda zwracajaca rozmiar problemu.

#### Returns

rozmiar tablicy

#### 3.1.2.4 LARGE\_INTEGER Dzialanie::wlaczStoper ( )

Metoda uruchamia pomiar czasu.

Czas jest mierzony w milisekundach

#### Returns

czas, w ktorym zostal wlaczony stoper

#### 3.1.2.5 void Dzialanie::wykonajAlgorytm ( )

Metoda wykonuje algorytm na danych wejsciovych ( tablicy)

Algorytm do wykonania : pomnoz kazdy element razy 2.

#### Returns

void

#### 3.1.2.6 LARGE\_INTEGER Dzialanie::wylaczStoper ( )

Metoda konczy pomiar czasu.

Czas jest mierzony w milisekundach

### Returns

czas, w którym stoper został wyłączony

The documentation for this class was generated from the following files:

- inc/dzialanie.hh
- scr/dzialanie.cpp

## 3.2 Kontener Class Reference

Klasa Dane.

```
#include <kontener.hh>
```

### Public Member Functions

- unsigned int [wez\\_rozmiar](#) ()  
*Metoda zwracająca rozmiar tablicy.*
- vector< int > & [wez\\_dane](#) ()  
*Metoda zwracająca referencje do tablicy danych Metoda pozwala na dostęp do tablicy i jej modyfikacje.*
- void [wczytajDane](#) (string nazwaPliku)  
*metoda wczytuje dane do tablicy z pliku*
- void [zamien\\_elementy](#) (unsigned int i, unsigned int j)  
*Metoda zamienia ze sobą dwa elementy tablicy.*
- void [odwroc\\_kolejnosc](#) ()  
*Metoda odwraca zawartość tablicy.*
- void [dodaj\\_element](#) (int e)  
*Metoda dodaje element na koniec tablicy.*
- void [dodaj\\_elementy](#) (Kontener tab)  
*Metoda dodaje na koniec tablicy zawartość innej tablicy.*
- int & [operator\[\]](#) (int index)  
*Operator indeksujący tablice.*
- Kontener & [operator+](#) (Kontener tab)  
*Operator dodawania tablic Operator pozwala na dodanie 2 tablic.*
- Kontener & [operator=](#) (Kontener tab)  
*Operator przypisania Operator pozwala na przypisanie do tablicy zawartości innej tablicy.*
- bool [operator==](#) (Kontener tab)  
*Operator porównania 2 tablic Operator pozwala na porównanie 2 tablic. Sprawdza on czy są takie same pod względem zawartości.*

### Friends

- ostream & [operator<<](#) (ostream &out, Kontener Tab)  
*Operator wypisywania Metoda pozwala na wypisanie zawartości tablicy na standardowe wyjście.*

### 3.2.1 Detailed Description

Klasa Dane.

Klasa posiada 2 pola: -tablice (vector), -rozmiar tablicy.

### 3.2.2 Member Function Documentation

#### 3.2.2.1 void Kontener::dodaj\_element ( int *e* )

Metoda dodaje element na koniec tablicy.

##### Parameters

<i>e</i>	- wartosc elementu
----------	--------------------

#### 3.2.2.2 void Kontener::dodaj\_elementy ( Kontener *tab* )

Metoda dodaje na koniec tablicy zawartosc innej tablicy.

##### Parameters

<i>tablica, ktora</i>	bedzie dodana na koniec
-----------------------	-------------------------

#### 3.2.2.3 Kontener & Kontener::operator+ ( Kontener *tab* )

Operator dodawania tablic Operator pozwala na dodanie 2 tablic.

`\param tablica do dodania`

##### Returns

dwie polaczone tablice

#### 3.2.2.4 Kontener & Kontener::operator= ( Kontener *tab* )

Operator przypisania Operator pozwala na przypisanie do tablicy zawartosci innej tablicy.

##### Parameters

<i>tablica, ktora</i>	przypisujemy
-----------------------	--------------

#### 3.2.2.5 bool Kontener::operator== ( Kontener *tab* )

Operator porownania 2 tablic Operator pozwala na porownanie 2 tablic. Sprawdza on czy sa takie same pod wzgledem zawartosci.

##### Parameters

<i>tablica, z</i>	ktora bedziemy porownywac
-------------------	---------------------------

##### Returns

true - tablice sa identyczne  
false - tablice sa rozne

### 3.2.2.6 `int & Kontener::operator[] ( int index )`

Operator indeksujący tablice.

#### Parameters

<i>index</i>	- indeks, którego referencja zostanie zwrócona
--------------	--

#### Returns

referencja do zadanego indeksu

### 3.2.2.7 `void Kontener::wczytajDane ( string nazwaPliku )`

metoda wczytuje dane do tablicy z pliku

Format danych w pliku jest następujący: pierwszy wiersz - ilość elementów, a następnie w kolumnie kolejne wartości tablicy.

#### Parameters

<i>nazwaPliku</i>	- nazwa pliku do otwarcia
-------------------	---------------------------

#### Returns

void

### 3.2.2.8 `vector<int>& Kontener::wez_dane ( ) [inline]`

Metoda zwracająca referencje do tablicy danych Metoda pozwala na dostęp do tablicy i jej modyfikacje.

#### Returns

referencja do tablicy

### 3.2.2.9 `unsigned int Kontener::wez_rozmiar ( ) [inline]`

Metoda zwracająca rozmiar tablicy.

#### Returns

rozmiar tablicy

### 3.2.2.10 `void Kontener::zamien_elementy ( unsigned int i, unsigned int j )`

Metoda zamienia ze sobą dwa elementy tablicy.

#### Parameters

<i>i, j</i>	- indeksy, które zostaną zamienione
-------------	-------------------------------------

The documentation for this class was generated from the following files:

- inc/kontener.hh

- scr/kontener.cpp

### 3.3 Tester Class Reference

Klasa [Tester](#) Klasa modeluje narzedzie do benchmarkowania algorytmu. Poprzez wielokrotne wynonywanie algorytmu mozemy wyciagnac sredni czas dzialania, co jest dokladniejszym pomiarem.

```
#include <tester.hh>
```

#### Public Member Functions

- [Tester](#) ()  
*konstruktor klasy Konstruktor inicjuje wartosci: -powtorzenia (ile razy wykonywac algorytm) -ilosc (ile mamy zestawow dancyh) -wejscie (nazwa pliku z danymi wejscowymi) -wynik (nazwa pliku z poprawnym wynikiem algorytmu)*
- void [otworzPlik](#) ()  
*Metoda otwierajaca referencje do pliku CSV.*
- void [zamknijPlik](#) ()  
*Metoda otwierajaca referencje do pliku CSV.*
- void [symulacja](#) ()  
*Metoda symulujaca badanie algorytmu Metoda wykonuje symulacje dzialania algorytmu. Wykonuje algorytm zadana liczbe razy dla zadanych zestawow danych. Wyniki zapisuje do pliku CSV o nazwie rezultat.csv Format zapisu: rozmiar\_problemu,ilosc\_powtorzen,sredni\_czas.*
- void [zamienNazwy](#) (int numer)  
*Metoda aktualizuje nazwe plikow wejscowych Format plikow wejscowych jest scisle okreslony. Dane wejscowe: wejscie%.txt, gdzie % to numer zestawu Dane wynikowe: wynik%.txt, gdzie % to numer zestawu Zestawy nalazy numerowac od 0.*

#### 3.3.1 Detailed Description

Klasa [Tester](#) Klasa modeluje narzedzie do benchmarkowania algorytmu. Poprzez wielokrotne wynonywanie algorytmu mozemy wyciagnac sredni czas dzialania, co jest dokladniejszym pomiarem.

#### 3.3.2 Member Function Documentation

##### 3.3.2.1 void [Tester::zamienNazwy](#) ( int *numer* )

Metoda aktualizuje nazwe plikow wejscowych Format plikow wejscowych jest scisle okreslony. Dane wejscowe: wejscie%.txt, gdzie % to numer zestawu Dane wynikowe: wynik%.txt, gdzie % to numer zestawu Zestawy nalazy numerowac od 0.

##### Parameters

<i>numer</i>	- % zostaje zamieniony na wartosc 'numer'
--------------	---

The documentation for this class was generated from the following files:

- inc/tester.hh
- scr/tester.cpp

# Index

- dodaj\_element
  - Kontener, [8](#)
- dodaj\_elementy
  - Kontener, [8](#)
- Dzialanie, [5](#)
  - sprawdz, [5](#)
  - wczytajDaneWejscowe, [6](#)
  - wez\_rozmiar, [6](#)
  - wlaczStoper, [6](#)
  - wykonajAlgorytm, [6](#)
  - wylaczStoper, [6](#)
- Kontener, [7](#)
  - dodaj\_element, [8](#)
  - dodaj\_elementy, [8](#)
  - operator+, [8](#)
  - operator=, [8](#)
  - operator==, [8](#)
  - wczytajDane, [9](#)
  - wez\_dane, [9](#)
  - wez\_rozmiar, [9](#)
  - zamien\_elementy, [9](#)
- operator+
  - Kontener, [8](#)
- operator=
  - Kontener, [8](#)
- operator==
  - Kontener, [8](#)
- sprawdz
  - Dzialanie, [5](#)
- Tester, [10](#)
  - zamienNazwy, [10](#)
- wczytajDane
  - Kontener, [9](#)
- wczytajDaneWejscowe
  - Dzialanie, [6](#)
- wez\_dane
  - Kontener, [9](#)
- wez\_rozmiar
  - Dzialanie, [6](#)
  - Kontener, [9](#)
- wlaczStoper
  - Dzialanie, [6](#)
- wykonajAlgorytm
  - Dzialanie, [6](#)
- wylaczStoper
  - Dzialanie, [6](#)
- zamien\_elementy
  - Kontener, [9](#)
- zamienNazwy
  - Tester, [10](#)