

## Laboratorium 3

Generated by Doxygen 1.8.1.2

Sun Mar 16 2014 23:33:05



# Contents

<b>1</b>	<b>PAMSI</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	wlaczStoper . . . . .	6
3.1.2.4	wykonajAlgorytm . . . . .	6
3.1.2.5	wylaczStoper . . . . .	6
3.2	kolejka_list Class Reference . . . . .	7
3.2.1	Detailed Description . . . . .	7
3.2.2	Member Function Documentation . . . . .	7
3.2.2.1	dequeue . . . . .	7
3.2.2.2	enqueue . . . . .	7
3.2.2.3	isempty . . . . .	7
3.2.2.4	size . . . . .	8
3.3	kolejka_tab Class Reference . . . . .	8
3.3.1	Detailed Description . . . . .	8
3.3.2	Member Function Documentation . . . . .	8
3.3.2.1	dequeue . . . . .	8
3.3.2.2	enqueue . . . . .	8
3.3.2.3	isempty . . . . .	9
3.3.2.4	size . . . . .	9
3.4	Kontener Class Reference . . . . .	9
3.4.1	Detailed Description . . . . .	10
3.4.2	Member Function Documentation . . . . .	10
3.4.2.1	dodaj_element . . . . .	10

3.4.2.2	dodaj_elementy	10
3.4.2.3	operator+	10
3.4.2.4	operator=	10
3.4.2.5	operator==	11
3.4.2.6	operator[]	11
3.4.2.7	wczytajDane	11
3.4.2.8	wez_dane	11
3.4.2.9	wez_rozmiar	11
3.4.2.10	zamien_elementy	12
3.5	stos_list Class Reference	12
3.5.1	Detailed Description	12
3.5.2	Member Function Documentation	12
3.5.2.1	isempty	12
3.5.2.2	pop	13
3.5.2.3	push	13
3.5.2.4	size	13
3.6	stos_tab Class Reference	13
3.6.1	Detailed Description	14
3.6.2	Member Function Documentation	14
3.6.2.1	isempty	14
3.6.2.2	pop	14
3.6.2.3	push	14
3.6.2.4	size	14
3.7	Tester Class Reference	14
3.7.1	Detailed Description	15
3.7.2	Member Function Documentation	15
3.7.2.1	zamienNazwy	15

## **Chapter 1**

# **PAMSI**



## 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="#">kolejka_list</a>	Klasa modelujaca strukture koleki Kolejka jest zbudowana w oparciu o liste . . . . .	7
<a href="#">kolejka_tab</a>	Klasa modelujaca strukture koleki Kolejka jest zbudowana w oparciu o liste . . . . .	8
<a href="#">Kontener</a>	Klasa Dane . . . . .	9
<a href="#">stos_list</a>	Klasa modelujaca strukture stosu Stos jest zbudowany w oparciu o liste . . . . .	12
<a href="#">stos_tab</a>	Klasa modelujaca strukture stosu Stos jest zbudowany w oparciu o tablice . . . . .	13
<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 . . . . .	14





## Chapter 3

# Class Documentation

### 3.1 Dzialanie Class Reference

Klasa modelujaca gowna czesc programu.

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

#### Public Member Functions

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

#### 3.1.1 Detailed Description

Klasa modelujaca gowna czesc programu.

Klasa modeluje glowna 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 Dzialanie::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::wczytajDaneWejscowe ( string nazwa, stos\_tab stos )**

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 LARGE\_INTEGER Dzialanie::włączStoper ( )**

Metoda uruchamia pomiar czasu.

Czas jest mierzony w milisekundach

**Returns**

czas, w którym został włączony stoper

**3.1.2.4 void Dzialanie::wykonajAlgorytm ( )**

Metoda wykonuje algorytm na danych wejscowych ( tablicy)

Algorytm do wykonania : pomnoz kazdy element razy 2.

**Returns**

void

**3.1.2.5 LARGE\_INTEGER Dzialanie::wylaczStoper ( )**

Metoda konczy pomiar czasu.

Czas jest mierzony w milisekundach

**Returns**

czas, w którym stoper został wylaczony

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

- inc/dzialanie.hh
- src/dzialanie.cpp

## 3.2 kolejka\_list Class Reference

Klasa modelująca strukture koleki Kolejka jest zbudowana w oparciu o liste.

```
#include <kolejka_list.hh>
```

### Public Member Functions

- void **enqueue** (int element)  
*Metoda dodajaca element na stos Metoda dodaje element na koniec listy.*
- void **dequeue** (int \*a)  
*Metoda usuwajaca element ze stos Metoda usuwa element znajdujacy sie na koncu listy. Gdy stos jest pusty, wyswietlony zostaje blad mowiacy ze brak danych do sciagniecia.*
- bool **isempty** ()  
*Metoda sprawdzajaca czy stos jest pusty. Metoda sprawdza czy stos jest pusty.*
- int **size** ()  
*Metoda zwracajaca rozmiar stosu Metoda zwraca rozmiar listy.*
- void **wyswietl** ()

### 3.2.1 Detailed Description

Klasa modelująca strukture koleki Kolejka jest zbudowana w oparciu o liste.

### 3.2.2 Member Function Documentation

#### 3.2.2.1 void kolejka\_list::dequeue ( int \* a )

Metoda usuwajaca element ze stos Metoda usuwa element znajdujacy sie na koncu listy. Gdy stos jest pusty, wyswietlony zostaje blad mowiacy ze brak danych do sciagniecia.

##### Parameters

<i>a</i>	- wskaznik do zmiennej, do ktorej sciagamy wartosc.
----------	---

#### 3.2.2.2 void kolejka\_list::enqueue ( int element )

Metoda dodajaca element na stos Metoda dodaje element na koniec listy.

##### Parameters

<i>element</i>	- element, ktory zostanie dodany do stosu
----------------	---

#### 3.2.2.3 bool kolejka\_list::isempty ( )

Metoda sprawdzajaca czy stos jest pusty. Metoda sprawdza czy stos jest pusty.

##### Returns

```
lista.empty();
```

### 3.2.2.4 int kolejka\_list::size ( )

Metoda zwracająca rozmiar stosu Metoda zwraca rozmiar listy.

#### Returns

rozmiar stosu.

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

- inc/kolejka\_list.hh
- src/kolejka\_list.cpp

## 3.3 kolejka\_tab Class Reference

Klasa modelująca strukturę kolejki Kolejka jest zbudowana w oparciu o listę.

```
#include <kolejka_tab.hh>
```

### Public Member Functions

- **kolejka\_tab** (int zwiększ)
- void **enqueue** (int element)  
*Metoda dodająca element na stos Metoda dodaje element na koniec listy.*
- void **dequeue** (int \*a)  
*Metoda usuwająca element ze stosu Metoda usuwa element znajdujący się na końcu listy. Gdy stos jest pusty, wyświetlony zostaje błąd mówiący o braku danych do ściągnięcia.*
- bool **isempty** ()  
*Metoda sprawdzająca czy stos jest pusty. Metoda sprawdza czy stos jest pusty.*
- int **size** ()  
*Metoda zwracająca rozmiar stosu Metoda zwraca rozmiar listy.*
- void **wyswietl** ()

### 3.3.1 Detailed Description

Klasa modelująca strukturę kolejki Kolejka jest zbudowana w oparciu o listę.

### 3.3.2 Member Function Documentation

#### 3.3.2.1 void kolejka\_tab::dequeue ( int \* a )

Metoda usuwająca element ze stosu Metoda usuwa element znajdujący się na końcu listy. Gdy stos jest pusty, wyświetlony zostaje błąd mówiący o braku danych do ściągnięcia.

#### Parameters

<b>a</b>	- wskaźnik do zmiennej, do której ściągamy wartość.
----------	---

#### 3.3.2.2 void kolejka\_tab::enqueue ( int element )

Metoda dodająca element na stos Metoda dodaje element na koniec listy.

## Parameters

<i>element</i>	- element, który zostanie dodany do stosu
----------------	---

## 3.3.2.3 bool kolejka\_tab::isempty ( )

Metoda sprawdzająca czy stos jest pusty. Metoda sprawdza czy stos jest pusty.

## Returns

lista.empty();

## 3.3.2.4 int kolejka\_tab::size ( )

Metoda zwracająca rozmiar stosu Metoda zwraca rozmiar listy.

## Returns

rozmiar stosu.

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

- inc/kolejka\_tab.hh
- src/kolejka\_tab.cpp

## 3.4 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 zawartosci innej tablicy.

- bool `operator==` (`Kontener` tab)

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

## Friends

- ostream & `operator<<` (ostream &out, `Kontener` Tab)

*Operator wypisywania* Metoda pozwala na wypisanie zawartosci tablicy na standardowe wyjście.

### 3.4.1 Detailed Description

Klasa Dane.

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

### 3.4.2 Member Function Documentation

#### 3.4.2.1 void Kontener::dodaj\_element ( int e )

Metoda dodaje element na koniec tablicy.

##### Parameters

e	- wartosc elementu
---	--------------------

#### 3.4.2.2 void Kontener::dodaj\_elementy ( Kontener tab )

Metoda dodaje na koniec tablicy zawartosc innej tablicy.

##### Parameters

tablica,ktora	bedzie dodana na koniec
---------------	-------------------------

#### 3.4.2.3 Kontener & Kontener::operator+ ( Kontener tab )

Operator dodawania tablic Operator pozwala na dodanie 2 tablic.

`\param tablica do dodania`

##### Returns

dwie polaczone tablice

#### 3.4.2.4 Kontener & Kontener::operator= ( Kontener tab )

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

##### Parameters

tablica,ktora	przypisujemy
---------------	--------------

#### 3.4.2.5 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

#### 3.4.2.6 int &Kontener::operator[]( int *index* )

Operator indeksujacy tablice.

##### Parameters

<i>index</i>	- indeks, ktorego referencja zostanie zwrocona
--------------	--

##### Returns

referencja do zadanego indeksu

#### 3.4.2.7 void Kontener::wczytajDane ( string *nazwaPliku* )

metoda wczytuje dane do tablicy z pliku

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

#### 3.4.2.8 vector<int>&Kontener::wez\_dane ( ) [inline]

Metoda zwracajaca referencje do tablicy danych Metoda pozwala na dostep do tablicy i jej modyfikacje.

##### Returns

referencja do tablicy

#### 3.4.2.9 unsigned int Kontener::wez\_rozmiar ( ) [inline]

Metoda zwracajaca rozmiar tablicy.

#### Returns

rozmiar tablicy

#### 3.4.2.10 void Kontener::zamien\_elementy ( unsigned int *i*, unsigned int *j* )

Metoda zamienia ze soba dwa elementy tablicy.

#### Parameters

<i>i,j</i>	- indeksy, ktore zostana zamienione
------------	-------------------------------------

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

- inc/kontener.hh
- src/kontener.cpp

## 3.5 stos\_list Class Reference

Klasa modelujaca strukture stosu Stos jest zbudowany w oparciu o liste.

```
#include <stos_list.hh>
```

### Public Member Functions

- void **push** (int element)  
*Metoda dodajaca element na stos Metoda dodaje element na koniec listy.*
- void **pop** (int \*a)  
*Metoda usuwajaca element ze stos Metoda usuwa element znajdujacy sie na koncu listy. Gdy stos jest pusty, wyswietlony zostaje blad mowiacy ze brak danych do sciagniecia.*
- bool **isempty** ()  
*Metoda sprawdzajaca czy stos jest pusty. Metoda sprawdza czy stos jest pusty.*
- int **size** ()  
*Metoda zwracajaca rozmiar stosu Metoda zwraca rozmiar listy.*
- void **wyswietl** ()

#### 3.5.1 Detailed Description

Klasa modelujaca strukture stosu Stos jest zbudowany w oparciu o liste.

#### 3.5.2 Member Function Documentation

##### 3.5.2.1 bool stos\_list::isempty ( )

Metoda sprawdzajaca czy stos jest pusty. Metoda sprawdza czy stos jest pusty.

#### Returns

lista.empty();



## 3.5.2.2 void stos\_list::pop ( int \* a )

Metoda usuwajaca element ze stos Metoda usuwa element znajdujacy sie na koncu listy. Gdy stos jest pusty, wyswietlony zostaje blad mowiacy ze brak danych do sciagniecia.

## Parameters

<i>a</i>	- wskaznik do zmiennej, do ktorej sciagamy wartosc.
----------	---

## 3.5.2.3 void stos\_list::push ( int element )

Metoda dodajaca element na stos Metoda dodaje element na koniec listy.

## Parameters

<i>element</i>	- element, ktory zostanie dodany do stosu
----------------	---

## 3.5.2.4 int stos\_list::size ( )

Metoda zwracajaca rozmiar stosu Metoda zwraca rozmiar listy.

## Returns

rozmiar stosu.

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

- inc/stos\_list.hh
- src/stos\_list.cpp

## 3.6 stos\_tab Class Reference

Klasa modelujaca strukture stosu Stos jest zbudowany w oparciu o tablice.

```
#include <stos_tab.hh>
```

## Public Member Functions

- **stos\_tab** (int zwieksz)
- void **push** (int element)  
*Metoda dodajaca element na stos Metoda dodaje element na koniec tablicy. W przypadku gdy tablica jest pelna zostaje ona powiekszona o 1 element.*
- void **pop** (int \*a)  
*Metoda usuwajaca element ze stos Metoda usuwa element znajdujacy sie na koncu tablicy. Z kazdorazowym pobraniem danych rozmiar tablicy jest zmniejszany o 1.*
- bool **isempty** ()  
*Metoda sprawdzajaca czy stos jest pusty. Metoda sprawdza czy stos jest pusty. Sprawdzenie polega na odczytaniu zmiennej rozmiar, ktora pamietala ile jest elementow na stosie.*
- int **size** ()  
*Metoda zwracajaca rozmiar stosu Metoda zwraca rozmiar stosu. Rozmiar stosu przechowywany jest w zmiennej rozmiar.*
- void **wyswietl** ()

### 3.6.1 Detailed Description

Klasa modelujaca strukture stosu Stos jest zbudowany w oparciu o tablice.

### 3.6.2 Member Function Documentation

#### 3.6.2.1 `bool stos_tab::isempty ( )`

Metoda sprawdzajaca czy stos jest pusty. Metoda sprawdza czy stos jest pusty. Sprawdzenie polega na odczytaniu zmiennej rozmiar,

ktora pamieta ile jest elementow na stosie.

##### Returns

true - rozmiar == 0  
false - rozmiar > 0.

#### 3.6.2.2 `void stos_tab::pop ( int * a )`

Metoda usuwajaca element ze stosu Metoda usuwa element znajdujacy sie na koncu tablicy. Z kazdorazowym pobraniem danych rozmiar tablicy jest zmniejszany o 1.

##### Parameters

<i>a</i>	- wskaznik do zmiennej, do ktorej sciagamy wartosc.
----------	---

#### 3.6.2.3 `void stos_tab::push ( int element )`

Metoda dodajaca element na stos Metoda dodaje element na koniec tablicy. W przypadku gdy tablica jest pelna zostaje ona powiekszona o 1 element.

##### Parameters

<i>element</i>	- element, ktory zostanie dodany do stosu
----------------	---

#### 3.6.2.4 `int stos_tab::size ( )`

Metoda zwracajaca rozmiar stosu Metoda zwraca rozmiar stosu. Rozmiar stosu przechowywany jest w zmiennej rozmiar.

##### Returns

rozmiar stosu.

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

- inc/stos\_tab.hh
- src/stos\_tab.cpp

## 3.7 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.*

### 3.7.1 Detailed Description

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

### 3.7.2 Member Function Documentation

#### 3.7.2.1 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
- src/tester.cpp

# Index

- dequeue
  - kolejka\_list, 7
  - kolejka\_tab, 8
- dodaj\_element
  - Kontener, 10
- dodaj\_elementy
  - Kontener, 10
- Dzialanie, 5
  - sprawdz, 5
  - wczytajDaneWejscowe, 6
  - wlaczStoper, 6
  - wykonajAlgorytm, 6
  - wylaczStoper, 6
- enqueue
  - kolejka\_list, 7
  - kolejka\_tab, 8
- isempty
  - kolejka\_list, 7
  - kolejka\_tab, 9
  - stos\_list, 12
  - stos\_tab, 14
- kolejka\_list, 7
  - dequeue, 7
  - enqueue, 7
  - isempty, 7
  - size, 7
- kolejka\_tab, 8
  - dequeue, 8
  - enqueue, 8
  - isempty, 9
  - size, 9
- Kontener, 9
  - dodaj\_element, 10
  - dodaj\_elementy, 10
  - operator+, 10
  - operator=, 10
  - operator==, 10
  - wczytajDane, 11
  - wez\_dane, 11
  - wez\_rozmiar, 11
  - zamien\_elementy, 12
- operator+
  - Kontener, 10
- operator=
  - Kontener, 10
- operator==
  - Kontener, 10
- pop
  - stos\_list, 12
  - stos\_tab, 14
- push
  - stos\_list, 13
  - stos\_tab, 14
- size
  - kolejka\_list, 7
  - kolejka\_tab, 9
  - stos\_list, 13
  - stos\_tab, 14
- sprawdz
  - Dzialanie, 5
- stos\_list, 12
  - isempty, 12
  - pop, 12
  - push, 13
  - size, 13
- stos\_tab, 13
  - isempty, 14
  - pop, 14
  - push, 14
  - size, 14
- Tester, 14
  - zamienNazwy, 15
- wczytajDane
  - Kontener, 11
- wczytajDaneWejscowe
  - Dzialanie, 6
- wez\_dane
  - Kontener, 11
- wez\_rozmiar
  - Kontener, 11
- wlaczStoper
  - Dzialanie, 6
- wykonajAlgorytm
  - Dzialanie, 6
- wylaczStoper
  - Dzialanie, 6
- zamien\_elementy
  - Kontener, 12
- zamienNazwy
  - Tester, 15