# PAMSI 3

Wygenerowano przez Doxygen 1.8.6

Pn, 17 mar 2014 01:23:48

ii SPIS TREŚCI

# Spis treści

1	Stro	na główna	1
2	Inde	eks klas	1
	2.1	Lista klas	1
3	Inde	eks plików	1
	3.1	Lista plików	1
4	Dok	umentacja klas	2
	4.1	Dokumentacja struktury ElementKolejki	2
		4.1.1 Opis szczegółowy	2
	4.2	Dokumentacja struktury ElementStosu	2
		4.2.1 Opis szczegółowy	3
	4.3	Dokumentacja klasy KolejkaLista	3
		4.3.1 Opis szczegółowy	3
		4.3.2 Dokumentacja funkcji składowych	4
	4.4	Dokumentacja klasy KolejkaTab	5
		4.4.1 Opis szczegółowy	6
		4.4.2 Dokumentacja funkcji składowych	6
	4.5	Dokumentacja klasy StosLista	7
		4.5.1 Opis szczegółowy	7
		4.5.2 Dokumentacja funkcji składowych	8
	4.6	Dokumentacja klasy StosTab1	9
		4.6.1 Opis szczegółowy	10
		4.6.2 Dokumentacja funkcji składowych	10
	4.7	Dokumentacja klasy StosTabx2	11
		4.7.1 Opis szczegółowy	11
		4.7.2 Dokumentacja funkcji składowych	12
5	Doki	umentacja plików	13
	5.1		13
			14
	5.2		14
	5.3		14
	5.4		14
	5.5		14
	5.6		14
	5.7		15
	5.8		15
	5.9		15
	0.0	=	

1 Strona główna 1

	5.10	Dokumentacja pliku StosTabx2.h	15
	5.11	Dokumentacja pliku TestStosowlKolejek.cpp	15
		5.11.1 Dokumentacja funkcji	16
6	Spra	wozdanie z wykonania programu	18
Ind	deks		19
1	Str	ona główna	

Laboratorium 3.

Rozne implementacje stosu i kolejki. Pomiar zlozonosci obliczeniowek wstawiania elementow do stosu zaimplentowanego na tablicy.

Autor

Jakub Chmiel 200314

### Indeks klas

### 2.1 Lista klas

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

ElementKolejki Struktura pojedynczego elementu przechowywanego w kolejce	2
ElementStosu Struktura pojedynczego elementu przechowywanego w stosie	2
KolejkaLista Implementacja kolejki na liscie	3
KolejkaTab Implementacja kolejki na tablicy	5
StosLista Implementacja stosu na liscie	7
StosTab1 Implementacja stosu na tablicy. Wersja z powiekszeniem tablicy o 1 element przy przekroczeniu	9
StosTabx2 Implementacja stosu na tablicy. Wersja z powiekszeniem tablicy dwukrotnie przy przekroczeniu	11

# Indeks plików

### Lista plików

Tutaj znajduje się lista wszystkich udokumentowanych plików z ich krótkimi opisami:

### definicje.h

Plik zawiera ogolne instrukcje preprocesora wspolne dla wszystkich plikow zrodlowych 13

KolejkaLista.cpp	14
KolejkaLista.h	14
KolejkaTablica.cpp	14
KolejkaTablica.h	14
StosLista.cpp	14
StosLista.h	15
StosTab1.cpp	15
StosTab1.h	15
StosTabx2.h	15
TestStosowlKolejek.cpp	15

# 4 Dokumentacja klas

### 4.1 Dokumentacja struktury ElementKolejki

Struktura pojedynczego elementu przechowywanego w kolejce.

```
#include <KolejkaLista.h>
```

### Metody publiczne

• ElementKolejki ()

Konstruktor.

•  $\sim$ ElementKolejki ()

Destruktor.

### Atrybuty publiczne

• ElementKolejki \* nastepny

Wskaznik do nastepnego elementu.

• ElementKolejki \* poprzedni

Wskaznik do poprzedniego elementu.

• TYP elem

Wlasciwy zapamietany element.

### 4.1.1 Opis szczegółowy

Struktura pojedynczego elementu przechowywanego w kolejce.

Dokumentacja dla tej struktury została wygenerowana z pliku:

· KolejkaLista.h

### 4.2 Dokumentacja struktury ElementStosu

Struktura pojedynczego elementu przechowywanego w stosie.

```
#include <StosLista.h>
```

#### Metody publiczne

• ElementStosu ()

Konstruktor.

∼ElementStosu ()

Destruktor.

### Atrybuty publiczne

• ElementStosu \* nastepny

Wskaznik do nastepnego elementu.

TYP elem

Wlasciwy zapamietany element.

### 4.2.1 Opis szczegółowy

Struktura pojedynczego elementu przechowywanego w stosie.

Dokumentacja dla tej struktury została wygenerowana z pliku:

• StosLista.h

### 4.3 Dokumentacja klasy KolejkaLista

Implementacja kolejki na liscie.

```
#include <KolejkaLista.h>
```

#### Metody publiczne

• KolejkaLista ()

Konstruktor.

∼KolejkaLista ()

Destruktor.

### Statyczne metody publiczne

• static bool enqueue (KolejkaLista &cel, TYP element)

Wstawia element na poczatek kolejki.

• static bool dequeue (KolejkaLista &cel, TYP &element)

Usuwa element z konca kolejki.

• static bool isempty (KolejkaLista &cel)

Sprawdza czy kolejka nie ma zadnych elementow.

• static int size (KolejkaLista &cel)

Sprawdza ile elementow jest w kolejce.

#### 4.3.1 Opis szczegółowy

Implementacja kolejki na liscie.

#### 4.3.2 Dokumentacja funkcji składowych

### 4.3.2.1 bool KolejkaLista::dequeue ( KolejkaLista & cel, TYP & element ) [static]

Usuwa element z konca kolejki.

#### **Parametry**

cel	docelowa kolejka.
element	sciagniety z kolejki.

#### Zwraca

powodzenie wykonania funkcji.

### 4.3.2.2 bool KolejkaLista::enqueue ( KolejkaLista & cel, TYP element ) [static]

Wstawia element na poczatek kolejki.

#### **Parametry**

cel	docelowa kolejka.
element	do wstawienia na kolejke docelowa.

#### Zwraca

powodzenie wykonania funkcji.

### 4.3.2.3 bool KolejkaLista::isempty ( KolejkaLista & cel ) [static]

Sprawdza czy kolejka nie ma zadnych elementow.

### **Parametry**

cel	docelowkolejka.
-----	-----------------

#### Zwraca

true - jesli pusty. false - jesli nie pusty.

### 4.3.2.4 int KolejkaLista::size ( KolejkaLista & cel ) [static]

Sprawdza ile elementow jest w kolejce.

#### **Parametry**

cel	docelowa kolejka.

### Zwraca

ilosc elementow.

Dokumentacja dla tej klasy została wygenerowana z plików:

- · KolejkaLista.h
- · KolejkaLista.cpp

### 4.4 Dokumentacja klasy KolejkaTab

Implementacja kolejki na tablicy.

#include <KolejkaTablica.h>

### Metody publiczne

· KolejkaTab ()

Konstruktor.

∼KolejkaTab ()

Destruktor.

### Statyczne metody publiczne

static bool enqueue (KolejkaTab &cel, TYP element)

Wstawia element na poczatek kolejki.

• static bool dequeue (KolejkaTab &cel, TYP &element)

Usuwa element z konca kolejki.

static bool isempty (KolejkaTab &cel)

Sprawdza czy kolejka nie ma zadnych elementow.

• static int size (KolejkaTab &cel)

Sprawdza ile elementow jest w kolejce.

#### 4.4.1 Opis szczegółowy

Implementacja kolejki na tablicy.

### 4.4.2 Dokumentacja funkcji składowych

4.4.2.1 bool KolejkaTab::dequeue ( KolejkaTab & cel, TYP & element ) [static]

Usuwa element z konca kolejki.

### **Parametry**

cel	docelowa kolejka.
element	sciagniety z kolejki.

#### Zwraca

powodzenie wykonania funkcji.

### 4.4.2.2 bool KolejkaTab::enqueue ( KolejkaTab & cel, TYP element ) [static]

Wstawia element na poczatek kolejki.

### **Parametry**

cel	docelowa kolejka.
element	do wstawienia na kolejke docelowa.

#### Zwraca

powodzenie wykonania funkcji.

4.4.2.3 bool KolejkaTab::isempty ( KolejkaTab & cel ) [static]

Sprawdza czy kolejka nie ma zadnych elementow.

cel	docelowkolejka.
-----	-----------------

#### Zwraca

```
true - jesli pusty. false - jesli nie pusty.
```

4.4.2.4 int KolejkaTab::size ( KolejkaTab & cel ) [static]

Sprawdza ile elementow jest w kolejce.

**Parametry** 

cel	docelowa kolejka.

#### Zwraca

ilosc elementow.

Dokumentacja dla tej klasy została wygenerowana z plików:

- · KolejkaTablica.h
- KolejkaTablica.cpp

### 4.5 Dokumentacja klasy StosLista

Implementacja stosu na liscie.

```
#include <StosLista.h>
```

### Metody publiczne

• StosLista ()

Konstruktor.

∼StosLista ()

Destruktor.

### Statyczne metody publiczne

• static bool push (StosLista &cel, TYP element)

Wstawia element na szczyt stosu.

• static bool pop (StosLista &cel, TYP &element)

Usuwa element ze szczytu stosu.

• static bool isempty (StosLista &cel)

Sprawdza czy stos nie ma zadnych elementow.

• static int size (StosLista &cel)

Sprawdza ile elementow jest na stosie.

#### 4.5.1 Opis szczegółowy

Implementacja stosu na liscie.

- 4.5.2 Dokumentacja funkcji składowych
- **4.5.2.1** bool StosLista::isempty ( StosLista & cel ) [static]

Sprawdza czy stos nie ma zadnych elementow.

cel	docelowy stos.
-----	----------------

#### Zwraca

true - jesli pusty. false - jesli nie pusty.

4.5.2.2 bool StosLista::pop ( StosLista & cel, TYP & element ) [static]

Usuwa element ze szczytu stosu.

### **Parametry**

cel	docelowey stos.
element	sciagniety ze stosu.

#### Zwraca

powodzenie wykonania funkcji.

4.5.2.3 bool StosLista::push ( StosLista & cel, TYP element ) [static]

Wstawia element na szczyt stosu.

#### **Parametry**

cel	docelowy stos.
element	do wstawienia na stos docelowy.

### Zwraca

powodzenie wykonania funkcji.

4.5.2.4 int StosLista::size ( StosLista & cel ) [static]

Sprawdza ile elementow jest na stosie.

#### **Parametry**

cel	docelowy stos.

#### Zwraca

ilosc elementow.

Dokumentacja dla tej klasy została wygenerowana z plików:

- StosLista.h
- · StosLista.cpp

### 4.6 Dokumentacja klasy StosTab1

Implementacja stosu na tablicy. Wersja z powiekszeniem tablicy o 1 element przy przekroczeniu.

#include <StosTab1.h>

#### Metody publiczne

StosTab1 ()

Konstruktor.

∼StosTab1 ()

Destruktor usuwa zaalakowana dynamicznie tablice.

#### Statyczne metody publiczne

static bool push (StosTab1 &cel, TYP element)

Wstawia element na szczyt stosu.

static bool pop (StosTab1 &cel, TYP &element)

Usuwa element ze szczytu stosu.

static bool isempty (StosTab1 &cel)

Sprawdza czy stos nie ma zadnych elementow.

static int size (StosTab1 &cel)

Sprawdza ile elementow jest na stosie.

#### 4.6.1 Opis szczegółowy

Implementacja stosu na tablicy. Wersja z powiekszeniem tablicy o 1 element przy przekroczeniu.

4.6.2 Dokumentacja funkcji składowych

4.6.2.1 bool StosTab1::isempty ( StosTab1 & cel ) [static]

Sprawdza czy stos nie ma zadnych elementow.

### **Parametry**

cel	docelowy stos.
-----	----------------

### Zwraca

true - jesli pusty. false - jesli nie pusty.

4.6.2.2 bool StosTab1::pop ( StosTab1 & cel, TYP & element ) [static]

Usuwa element ze szczytu stosu.

#### **Parametry**

cel	docelowey stos.
element	sciagniety ze stosu.

#### Zwraca

powodzenie wykonania funkcji.

4.6.2.3 bool StosTab1::push ( StosTab1 & cel, TYP element ) [static]

Wstawia element na szczyt stosu.

cel	docelowy stos.
element	do wstawienia na stos docelowy.

#### Zwraca

powodzenie wykonania funkcji.

4.6.2.4 int StosTab1::size ( StosTab1 & cel ) [static]

Sprawdza ile elementow jest na stosie.

**Parametry** 

cel	docelowy stos.

#### Zwraca

ilosc elementow.

Dokumentacja dla tej klasy została wygenerowana z plików:

- StosTab1.h
- StosTab1.cpp

### 4.7 Dokumentacja klasy StosTabx2

Implementacja stosu na tablicy. Wersja z powiekszeniem tablicy dwukrotnie przy przekroczeniu.

```
#include <StosTabx2.h>
```

### Metody publiczne

· StosTabx2 ()

Konstruktor.

∼StosTabx2 ()

Destruktor usuwa zaalakowana dynamicznie tablice.

### Statyczne metody publiczne

• static bool push (StosTabx2 &cel, TYP element)

Wstawia element na szczyt stosu.

static bool pop (StosTabx2 &cel, TYP &element)

Usuwa element ze szczytu stosu.

static bool isempty (StosTabx2 &cel)

Sprawdza czy stos nie ma zadnych elementow.

• static int size (StosTabx2 &cel)

Sprawdza ile elementow jest na stosie.

#### 4.7.1 Opis szczegółowy

Implementacja stosu na tablicy. Wersja z powiekszeniem tablicy dwukrotnie przy przekroczeniu.

- 4.7.2 Dokumentacja funkcji składowych
- **4.7.2.1** bool StosTabx2::isempty ( StosTabx2 & cel ) [static]

Sprawdza czy stos nie ma zadnych elementow.

001	docalowy stos
Cei	docelowy sios.

#### Zwraca

true - jesli pusty. false - jesli nie pusty.

4.7.2.2 bool StosTabx2::pop ( StosTabx2 & cel, TYP & element ) [static]

Usuwa element ze szczytu stosu.

#### **Parametry**

cel	docelowey stos.
element	sciagniety ze stosu.

#### Zwraca

powodzenie wykonania funkcji.

4.7.2.3 bool StosTabx2::push ( StosTabx2 & cel, TYP element ) [static]

Wstawia element na szczyt stosu.

### **Parametry**

cel	docelowy stos.
element	do wstawienia na stos docelowy.

#### Zwraca

powodzenie wykonania funkcji.

4.7.2.4 int StosTabx2::size ( StosTabx2 & cel ) [static]

Sprawdza ile elementow jest na stosie.

#### **Parametry**

cel	docelowy stos.

### Zwraca

ilosc elementow.

Dokumentacja dla tej klasy została wygenerowana z plików:

- · StosTabx2.h
- StosTabx2.cpp

## 5 Dokumentacja plików

### 5.1 Dokumentacja pliku definicje.h

Plik zawiera ogolne instrukcje preprocesora wspolne dla wszystkich plikow zrodlowych.

### Definicje

• #define TYP int

### 5.1.1 Opis szczegółowy

Plik zawiera ogolne instrukcje preprocesora wspolne dla wszystkich plikow zrodlowych.

### 5.2 Dokumentacja pliku KolejkaLista.cpp

```
#include "KolejkaLista.h"
```

### 5.3 Dokumentacja pliku KolejkaLista.h

```
#include "definicje.h"
#include <stdlib.h>
```

### Komponenty

• struct ElementKolejki

Struktura pojedynczego elementu przechowywanego w kolejce.

class KolejkaLista

Implementacja kolejki na liscie.

### 5.4 Dokumentacja pliku Kolejka Tablica.cpp

```
#include "KolejkaTablica.h"
```

### 5.5 Dokumentacja pliku Kolejka Tablica.h

```
#include "definicje.h"
#include <stdlib.h>
```

#### Komponenty

· class KolejkaTab

Implementacja kolejki na tablicy.

### 5.6 Dokumentacja pliku StosLista.cpp

```
#include "StosLista.h"
```

### 5.7 Dokumentacja pliku StosLista.h

```
#include "definicje.h"
#include <stdlib.h>
```

#### Komponenty

• struct ElementStosu

Struktura pojedynczego elementu przechowywanego w stosie.

· class StosLista

Implementacja stosu na liscie.

### 5.8 Dokumentacja pliku StosTab1.cpp

```
#include "StosTab1.h"
```

### 5.9 Dokumentacja pliku StosTab1.h

```
#include <stdlib.h>
#include "definicje.h"
```

### Komponenty

• class StosTab1

Implementacja stosu na tablicy. Wersja z powiekszeniem tablicy o 1 element przy przekroczeniu.

### 5.10 Dokumentacja pliku StosTabx2.h

```
#include <stdlib.h>
#include "definicje.h"
```

### Komponenty

class StosTabx2

Implementacja stosu na tablicy. Wersja z powiekszeniem tablicy dwukrotnie przy przekroczeniu.

### 5.11 Dokumentacja pliku TestStosowlKolejek.cpp

```
#include <iostream>
#include <fstream>
#include <time.h>
#include <stdlib.h>
#include "definicje.h"
#include "StosTab1.h"
#include "StosTabx2.h"
#include "StosLista.h"
#include "KolejkaLista.h"
#include "KolejkaTablica.h"
```

#### Definicje

#define PLIK DANYCH "dane3.txt"

nazwa pliku z danymi wejsciowymi, takze dla wyjscia generowanych liczb.

• #define LICZBA POWTORZEN 10

ilosc powtorzen pomiaru czasu dla kazdego rozmiaru problemu.

#define LICZBA WIELKOSCI 9

ilosc roznych rozmiarow problemu.

#define WIELKOSCI PROBLEMU {10, 100, 1000, 2000, 3000, 4000, 6000, 8000, 10000}

tablica zawierajaca wszystkie mierzone rozmiary problemu.

#define WIELKOSC\_GENEROWANYCH\_DANYCH 10000

ilosc danych do wygenerowania.

#### **Funkcje**

• double algorytm1 (TYP \*zrodlo, int rozmiar)

Testowany algorytm dla stosu tablicowego z powiekszeniem 1.

double algorytm2 (TYP \*zrodlo, int rozmiar)

Testowany algorytm dla stosu tablicowego z powiekszeniem x2.

• bool wczytaj\_dane (const char \*nazwa\_pliku, TYP \*&tablica, int &rozmiar)

wczytuje dane z pliku

- bool zapisz\_dane (const char \*nazwa\_pliku, int \*col\_rozmiar\_problemu, double \*col\_czas, int rozmiar)
   zapisuje dane do pliku .csv dane zawieraja: 1 kolumna: wielkosc problemu 2 kolumna: czas potrzebny do zrealizowanego danego problemu
- void testuj\_algorytm (double(\*algorytm)(TYP \*zrodlo, int rozmiar), TYP \*tablica, int rozmiar, const char \*plik-\_wyjsciowy)

wykonuje testy czasu algorytmu dla przygotowanych parametrow zmierzone czasu zapisuje do pliku

• void generuj\_dane ()

generuje dane jesli ich nie ma

void sprawdz\_poprawnosc ()

funkcja sprawdza poprawnosc dzialania struktur danych.

• int main ()

funkcja main

### 5.11.1 Dokumentacja funkcji

5.11.1.1 double algorytm1 ( TYP \* zrodlo, int rozmiar )

Testowany algorytm dla stosu tablicowego z powiekszeniem 1.

#### **Parametry**

zrodlo	tablica danych do wrzucenia na stos
rozmiar	ilosc danych

#### Zwraca

czas wykonywania

5.11.1.2 double algorytm2 ( TYP \* zrodlo, int rozmiar )

Testowany algorytm dla stosu tablicowego z powiekszeniem x2.

zrodlo	tablica danych do wrzucenia na stos
rozmiar	ilosc danych

#### Zwraca

czas wykonywania

5.11.1.3 void testuj\_algorytm ( double(\*)(TYP \*zrodlo, int rozmiar) algorytm, TYP \* tablica, int rozmiar, const char \* plik\_wyjsciowy )

wykonuje testy czasu algorytmu dla przygotowanych parametrow zmierzone czasu zapisuje do pliku

\*\*

### **Parametry**

		funkcja z algorytmem do testowania
	double(*algorytm)	
Ī	*tablica	dane dla algorytmu
Ī	rozmiar	rozmiar tablicy
	*plik_wyjsciowy	nazwa pliku do zapisu zmierzonych czasow

5.11.1.4 bool wczytaj\_dane ( const char \* nazwa\_pliku, TYP \*& tablica, int & rozmiar )

wczytuje dane z pliku

Format:

liczba\_danych

dana1

dana2

.

### **Parametry**

*nazwa_pliku	nazwa pliku z danymi
*&tablica	tablica docelowa (usuwana w przypadku !=NULL)
&rozmiar	rozmiar tablicy docelowej

5.11.1.5 bool zapisz\_dane ( const char \* nazwa\_pliku, int \* col\_rozmiar\_problemu, double \* col\_czas, int rozmiar )

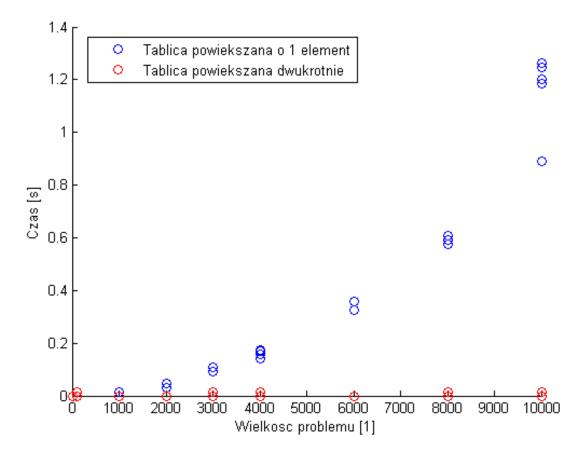
zapisuje dane do pliku .csv dane zawieraja: 1 kolumna: wielkosc problemu 2 kolumna: czas potrzebny do zrealizowanego danego problemu

#### **Parametry**

*nazwa_pliku	nazwa pliku do zapisu
*col_rozmiar	tablica z 1 kolumna
problemu	
*col_czas	druga kolumna
rozmiar	rozmiar obu tablic

### Zwraca

- · true sukces
- · false blad



Rysunek 1: Wykres pomiaru czasu.

# 6 Sprawozdanie z wykonania programu

Pomiar czasu dla dwóch różnych implementacji stosu na tablicy przedstawiono na wykresie 1. Z wykresu wynika, że implementacja korzystająca z powiększania tablicy zawsze o jeden element ma dużo większą złożoność obliczeniową (prawdopodobnie kwadratową) niż implementacja polegająca na powiększaniu tablicy dwukrotnie.

# Skorowidz

algorytm1	size, 9
TestStosowlKolejek.cpp, 16	StosLista.cpp, 14
algorytm2	StosLista.h, 15
TestStosowlKolejek.cpp, 16	StosTab1, 9
	isempty, 10
definicje.h, 13	pop, 10
dequeue	push, 10
KolejkaLista, 4	size, 11
KolejkaTab, 6	StosTab1.cpp, 15
Element/cloiki 2	StosTab1.h, 15
ElementKolejki, 2	StosTabx2, 11
ElementStosu, 2	isempty, 12
enqueue Koloikal ista E	pop, 13
KolejkaLista, 5	push, 13
KolejkaTab, 6	size, 13
isempty	StosTabx2.h, 15
KolejkaLista, 5	T 101 H4 1 1 1 45
KolejkaTab, 6	TestStosowlKolejek.cpp, 15
StosLista, 8	algorytm1, 16
StosTab1, 10	algorytm2, 16
StosTabx2, 12	testuj_algorytm, 17
,	wczytaj_dane, 17
KolejkaLista, 3	zapisz_dane, 17
dequeue, 4	testuj_algorytm
enqueue, 5	TestStosowlKolejek.cpp, 17
isempty, 5	
size, 5	wczytaj_dane
KolejkaLista.cpp, 14	TestStosowlKolejek.cpp, 17
KolejkaLista.h, 14	zaniaz dana
KolejkaTab, 5	zapisz_dane TootStangy/Koloick.opp. 17
dequeue, 6	TestStosowlKolejek.cpp, 17
enqueue, 6	
isempty, 6	
size, 7	
KolejkaTablica.cpp, 14	
KolejkaTablica.h, 14	
pop	
StosLista, 9	
StosTab1, 10	
StosTabx2, 13	
push	
StosLista, 9	
StosTab1, 10	
StosTabx2, 13	
size	
KolejkaLista, 5	
KolejkaTab, 7	
StosLista, 9	
StosTab1, 11	
StosTabx2, 13	
StosLista, 7	
isempty, 8	
pop, 9	
push, 9	