

Złożoność obliczeniowa

Generated by Doxygen 1.8.1.2

Sun Mar 16 2014 23:43:37

Contents

1	Program wyliczający czas wykonywanego alorytmu	1
2	Class Index	3
2.1	Class List	3
3	File Index	5
3.1	File List	5
4	Class Documentation	7
4.1	dane Class Reference	7
4.1.1	Detailed Description	8
4.1.2	Member Function Documentation	8
4.1.2.1	dodaj_element	8
4.1.2.2	dodaj_elementy	8
4.1.2.3	odwroc_element	8
4.1.2.4	operator+	8
4.1.2.5	operator=	9
4.1.2.6	operator==	9
4.1.2.7	operator[]	9
4.1.2.8	size	9
4.1.2.9	wczytaj	10
4.1.2.10	wnetrze	10
4.1.2.11	wypisz	10
4.1.2.12	zamien_element	10
4.2	kolejkalista Class Reference	11
4.2.1	Detailed Description	11
4.2.2	Member Function Documentation	11
4.2.2.1	dequeue	11
4.2.2.2	enqueue	11
4.2.2.3	isempty	12
4.2.2.4	size	12
4.3	kolejkatab Class Reference	12

4.3.1	Detailed Description	12
4.3.2	Constructor & Destructor Documentation	13
4.3.2.1	kolejkatab	13
4.3.3	Member Function Documentation	13
4.3.3.1	dequeue	13
4.3.3.2	enqueue	13
4.3.3.3	isempty	13
4.3.3.4	size	13
4.4	stos Class Reference	14
4.4.1	Detailed Description	14
4.4.2	Constructor & Destructor Documentation	14
4.4.2.1	stos	14
4.4.2.2	stos	15
4.4.3	Member Function Documentation	15
4.4.3.1	isempty	15
4.4.3.2	pop	15
4.4.3.3	push	15
4.4.3.4	size	15
4.5	stoslista Class Reference	16
4.5.1	Detailed Description	16
4.5.2	Member Function Documentation	16
4.5.2.1	isempty	16
4.5.2.2	pop	16
4.5.2.3	push	17
4.5.2.4	size	17
4.6	zegar Class Reference	17
4.6.1	Detailed Description	18
4.6.2	Constructor & Destructor Documentation	18
4.6.2.1	zegar	18
4.6.3	Member Function Documentation	18
4.6.3.1	algorytm	18
4.6.3.2	porownaj	18
4.6.3.3	wczytaj	19
4.6.3.4	wczytaj_dane_pod	19
4.6.3.5	wczytaj_dane_spr	19
4.6.3.6	wlaczStoper	19
4.6.3.7	wylaczStoper	20
5	File Documentation	21
5.1	C:/Users/Ania/workspace/zadanie/doc/pages/strona.dox File Reference	21

5.2	C:/Users/Ania/workspace/zadanie/inc/dane.hh File Reference	21
5.2.1	Detailed Description	21
5.3	C:/Users/Ania/workspace/zadanie/inc/kolejka.hh File Reference	21
5.3.1	Detailed Description	22
5.4	C:/Users/Ania/workspace/zadanie/inc/kolejka_lista.hh File Reference	22
5.4.1	Detailed Description	22
5.5	C:/Users/Ania/workspace/zadanie/inc/stos.hh File Reference	22
5.5.1	Detailed Description	22
5.6	C:/Users/Ania/workspace/zadanie/inc/stos_lista.hh File Reference	23
5.6.1	Detailed Description	23
5.7	C:/Users/Ania/workspace/zadanie/inc/uruchom.hh File Reference	23
5.7.1	Detailed Description	23
5.8	C:/Users/Ania/workspace/zadanie/src/dane.cpp File Reference	23
5.9	C:/Users/Ania/workspace/zadanie/src/kolejka.cpp File Reference	24
5.10	C:/Users/Ania/workspace/zadanie/src/kolejka_lista.cpp File Reference	24
5.11	C:/Users/Ania/workspace/zadanie/src/main.cpp File Reference	24
5.11.1	Detailed Description	24
5.11.2	Function Documentation	24
5.11.2.1	main	24
5.12	C:/Users/Ania/workspace/zadanie/src/stos.cpp File Reference	24
5.13	C:/Users/Ania/workspace/zadanie/src/stos_lista.cpp File Reference	25
5.14	C:/Users/Ania/workspace/zadanie/src/uruchom.cpp File Reference	25

Chapter 1

Program wyliczający czas wykonywanego algorytmu

Author

Anna Plywaczyk, 200340

Date

02.03.2014

Version

0,1

Aplikacja porozumiewa się z użytkownikiem. Prosi o podanie nazwy pliku, na którym ma być wykonany algorytm. W programie zostanie włączony stoper, którym zmierzymy czas w jakim algorytm zostanie wykonany. Pomiar czasu zostanie wykonany kilkakrotnie i na ekran zostanie wypisana wartość średnia. Aplikacja ponownie poprosi o podanie nazwy pliku, który ma zostać porównany z plikiem pomnożonym przez 2. Jeżeli wektory będą jednakowe program zwróci komunikat o tym iż mnożenie zostało wykonane poprawnie w przeciwnym wypadku zostanie zwrócony komunikat o błędnym wykonaniu algorytmu.

Chapter 2

Class Index

2.1 Class List

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

dane	Modeluje pojecie danych, uytych w programie, kt moim przypadku sa wektorem, ktory zostaje wczytany z pliku. Pierwsza zmienna wczytana z pliku jest liczba elemntow wystepujacych w tym pliku. Przyjelam, ze zmienne wczytane z pliku sa liczbami calkowitymi (int)	7
kolejkalista	Klasa modeluje pojecie kolejki, bazujacej na liscie. Wykonywane sa operacje dodawnia fo kolejki, zdjecia z kolejki, sprawdzenia czy jest pusta oraz sprawdzenia rozmiaru. Tablice zaalokowalam w sposob dynamiczny	11
kolejkatab	Klasa modeluje pojecie kolejki, bazujacej na tablicy. Wykonywane sa operacje dodawnia do kolejki, zdjecia z kolejki, sprawdzenia czy jest pusta oraz sprawdzenia rozmiaru. Tablice zaalokowalam w sposob dynamiczny. Dodatkowo uyte niektore funkcje pomocnicze jak np. powiekszenie tablicy	12
stos	Klasa modeluje pojecie stosu,bazujacego na tablicy. Na stosie wyonywane sa operacje dodatnia na stos, zdjecia ze stosu, sprawdzenia czy jest pusty oraz sprawdzenia rozmiaru. Tablice zaalokowalam w sposob dynamiczny. Dodatkowo uyte niektore funkcje pomocnicze jak np. powiekszenie stosu	14
stoslista	Klasa modeluje pojecie stosu,bazujacego na liscie. Na stosie wyonywane sa operacje dodatnia na stos, zdjecia ze stosu, sprawdzenia czy jest pusty oraz sprawdzenia rozmiaru	16
zegar	Klasa modeluje uruchomienia gownych wlasciowosci programu. Atrybutem klasy sa stowrzone dwa elemnty klasy dane, na ktorych wykonywane sa dzialania	17

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

C:/Users/Ania/workspace/zadanie/inc/ dane.hh	
Definicja klasy dane	21
C:/Users/Ania/workspace/zadanie/inc/ kolejka.hh	
Definicja klasy kolejkat	21
C:/Users/Ania/workspace/zadanie/inc/ kolejka_lista.hh	
Definicja klasy kolejkalista	22
C:/Users/Ania/workspace/zadanie/inc/ stos.hh	
Definicja klasy stos	22
C:/Users/Ania/workspace/zadanie/inc/ stos_lista.hh	
Definicja klasy stoslista	23
C:/Users/Ania/workspace/zadanie/inc/ uruchom.hh	
Definicja klasy zegar	23
C:/Users/Ania/workspace/zadanie/src/ dane.cpp	23
C:/Users/Ania/workspace/zadanie/src/ kolejka.cpp	24
C:/Users/Ania/workspace/zadanie/src/ kolejka_lista.cpp	24
C:/Users/Ania/workspace/zadanie/src/ main.cpp	
Funkcja glowna ktorej glownym zalozeniem jest wczytanie plikow z rozna wielkoscia elementow znajdujacych siliku, obliczenie sredniej wartosci czasu, w jakim zostaje wykonany algorytm (w naszym przypadku pomnozenie przez 2),nastepnie program porownuje poprawnosc wykoannia mnozenia z plikiem sprawdzajacym. Uzytkownik musi w programie zdefiniowac: liczbe powtorzen (zmienna j), ilosc plikow - do ilu wykonywane jest mnozenie (zmienna i), nazwy plikow (string czesc_1, i, czesc_2 - wszystko opcjonalnie)	24
C:/Users/Ania/workspace/zadanie/src/ stos.cpp	24
C:/Users/Ania/workspace/zadanie/src/ stos_lista.cpp	25
C:/Users/Ania/workspace/zadanie/src/ uruchom.cpp	25

Chapter 4

Class Documentation

4.1 dane Class Reference

Modeluje pojecie danych, uitych w programie, kt moim przypadku sa wektorem, który zostaje wczytany z pliku. Pierwsza zmienna wczytana z pliku jest liczba elemntow wystepujacych w tym pliku. Przyjelam, ze zmienne wczytane z pliku sa liczbami calkowitymi (int).

```
#include <dane.hh>
```

Public Member Functions

- void [wczytaj](#) (string nazwa)
Funkcja wczytująca dane do wektora z pliku. Funkcja otwiera plik zdefiniowany w glownej funkcji przez uytownika, sprawdza czy plik zostal otwarty, jeeli zostal otwarty wczytana jest liczba elementow pliku, nastepnie wczytywane sa wszystkie liczby do wektora.
- void [wypisz](#) ()
Funkcja wypisująca na ekran wszystkie elementy wektora. Funkcja wypisuje na ekran wszystkie elemnty z pliku na ekran w postaci wektora w kolumnie.
- void [zamien_element](#) (int i, int j)
Zamiana elementow Funkcja zamiania dwa elementy wektora, zadane poprzez wywołanie argumentow funkcji.
- void [dodaj_element](#) (int e)
Dodawanie elementu. Funkcja dodaje element na koniec wektora. Funkcja zdefiniowana jest poprzez wywołanie argumentow funkcji.
- void [odwroc_element](#) ()
Odwracanie elementow Funkcja ogawraca wektor, ostatni element wektora staje sierwszy, a pierwszy ostatnim.
- void [dodaj_elementy](#) (dane wektor2)
Dodawanie elementu. Funkcja dodaje element na koniec wektora. Funkcja zdefiniowana jest poprzez wywołanie argumentow funkcji.
- int & [operator\[\]](#) (int indeks)
Uzycie operatora [] Przeciazienie operatora stwoarzone abysmy mogli odwolac sie do konkretnego elementu wektora.
- unsigned int [size](#) ()
Rozmiar wektora.
- [dane](#) & [operator+](#) (dane wektor2)
Uzycie operatora + na wektorze Przeciazienie operatora dodwania, ktory mozemy wykonywac na wektorach.
- [dane](#) & [operator=](#) (dane wektor2)
Uzycie operatora = na wektorze Przeciazienie opertora przypisywania.
- vector< int > & [wnetrze](#) ()
Metoda dajca dostep do zawartosci wektora danych.
- bool [operator==](#) (dane wektor2)
Operator porownania dwoch wektorow Funkcja, ktora jest operatorem porownania dwoch wektorow.

4.1.1 Detailed Description

Modeluje pojecie danych, uitych w programie, kt moim przypadku sa wektorem, ktory zostaje wczytany z pliku. Pierwsza zmienna wczytana z pliku jest liczba elemntow wystepujacych w tym pliku. Przyjelam, ze zmienne wczytane z pliku sa liczbami calkowitymi (int).

Definition at line 31 of file dane.hh.

4.1.2 Member Function Documentation

4.1.2.1 void dane::dodaj_element (int e)

Dodawanie elementu. Funkcja dodaje element na koniec wektora. Funkcja zdefiniowana jest poprzez wywołanie argumentow funkcji.

Parameters

in	e	- liczba, ktora ma zostac dodana na koniec wektora. return (brak)
----	---	---

Definition at line 49 of file dane.cpp.

4.1.2.2 void dane::dodaj_elementy (dane wektor2)

Dodawanie elementu. Funkcja dodaje element na koniec wektora. Funkcja zdefiniowana jest poprzez wywołanie argumentow funkcji.

Parameters

in	wektor2	- wektor, ktory ma zostac dodany na koniec wektora, z dwoch zostaje stworzony jeden. return (brak)
----	---------	--

Definition at line 66 of file dane.cpp.

4.1.2.3 void dane::odwroc_element ()

Odwracanie elementow Funkcja ogawraca wektor, ostatni element wektora staje sierwszy, a pierwszy ostatnim.

Returns

(brak)

Definition at line 55 of file dane.cpp.

4.1.2.4 dane & dane::operator+ (dane wektor2)

Uzycie operatora + na wektorze Przeciazienie operatora dodwania, ktory mozemy wykonywac na wektorach.

Parameters

in	wektor2	- wektor danych ktory ma zostac dodany do wektora glownego
----	---------	--

Returns

Zwraca wektor, ktory jest suma dwoch innych

Definition at line 79 of file dane.cpp.

4.1.2.5 `dane & dane::operator= (dane wektor2)`

Uzycie operatora = na wektorze Przeciazanie opertora przypisywania.

Parameters

<code>in</code>	<code>wektor2</code>	- wektor danych ktory ma zostac przypisany do wektora glownego
-----------------	----------------------	--

Returns

Zwraca wektor, do ktorego zostal przypisany inny wektor

Definition at line 85 of file dane.cpp.

4.1.2.6 `bool dane::operator== (dane wektor2)`

Operator porownania dwuch wektorow Funkcja, ktora jest operatorem porownania dwuch wektorow.

Parameters

<code>in</code>	<code>wektor2</code>	- wektor danych ktory zostaje porownany z danymi glownymi
-----------------	----------------------	---

Returns

True gdy wektory danych sa jednakowe, natomiast jesli nawet jeden element sie rozni od wektora porownywanego zwraca false.

Definition at line 93 of file dane.cpp.

4.1.2.7 `int & dane::operator[] (int indeks)`

Uzycie operatora [] Przeciazanie operatora stwoarzone abysmy mogli odwolac sie do konkretnego elementu wektora.

Parameters

<code>in</code>	<code>indeks</code>	- zmienna calkowita, poperzez ktora mozemy dostac do konkretnego elementu wektora.
-----------------	---------------------	--

Returns

Zwraca wartosc jaka znajduje siadanym elemencie wektora.

Definition at line 74 of file dane.cpp.

4.1.2.8 `unsigned int dane::size () [inline]`

Rozmiar wektora.

Returns

Funkcja zwaraca liczbe elementow wektora.

Definition at line 102 of file dane.hh.

4.1.2.9 void dane::wczytaj (string nazwa)

Funkcja wczytująca dane do wektora z pliku. Funkcja otwiera plik zdefiniowany w głównej funkcji przez użytkownika, sprawdza czy plik został otwarty, jeżeli został otwarty wczytana jest liczba elementów pliku, następnie wczytywane są wszystkie liczby do wektora.

Parameters

in	<i>nazwa</i>	- zmienna, która jest wprowadzona przez użytkownika do programu
----	--------------	---

Returns

(brak)

Definition at line 12 of file dane.cpp.

4.1.2.10 vector<int>& dane::wnetrze () [inline]

Metoda dająca dostęp do zawartości wektora danych.

Returns

Wektor danych.

Definition at line 123 of file dane.hh.

4.1.2.11 void dane::wypisz ()

Funkcja wypisująca na ekran wszystkie elementy wektora. Funkcja wypisuje na ekran wszystkie elementy z pliku na ekran w postaci wektora w kolumnie.

Returns

(brak)

Definition at line 29 of file dane.cpp.

4.1.2.12 void dane::zamien_element (int i, int j)

Zamiana elementów Funkcja zamienia dwa elementy wektora, zadane poprzez wywołanie argumentów funkcji.

Parameters

in	<i>i</i>	- pierwszy numer elementu, który ma zostać zamieniony.
in	<i>j</i>	- drugi numer elementu, który ma zostać zamieniony.

Returns

(brak)

Definition at line 37 of file dane.cpp.

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

- C:/Users/Ania/workspace/zadanie/inc/[dane.hh](#)
- C:/Users/Ania/workspace/zadanie/src/[dane.cpp](#)

4.2 kolejkalista Class Reference

Klasa modeluje pojecie kolejki, bazujacej na liscie. Wykonywane sa operacje dodawania do kolejki, zdjecia z kolejki, sprawdzenia czy jest pusta oraz sprawdzenia rozmiaru. Tablice zaalokowalam w sposob dynamiczny.

```
#include <kolejka_lista.hh>
```

Public Member Functions

- void [enqueue](#) (int el_dodawany)
Dodanie elementu z kolejki Funkcja dodaje element na poczatek kolejki.
- void [dequeue](#) (int *a)
Zdejmowanie elementu z kolejki. Funkcja zdejmuje element z konca kolejki.
- bool [isempty](#) ()
Sprawdzanie pojemnosci w kolejce. Funkcja sprawdza czy jest pusta poprzez porownanie liczby_elementow do 0.
- int [size](#) ()
Zwrocenie rozmiaru funkcji. Funkcja sprawdza ile elementow jest w kolejce.

4.2.1 Detailed Description

Klasa modeluje pojecie kolejki, bazujacej na liscie. Wykonywane sa operacje dodawania do kolejki, zdjecia z kolejki, sprawdzenia czy jest pusta oraz sprawdzenia rozmiaru. Tablice zaalokowalam w sposob dynamiczny.

Definition at line 29 of file kolejka_lista.hh.

4.2.2 Member Function Documentation

4.2.2.1 void kolejkalista::dequeue (int * a)

Zdejmowanie elementu z kolejki. Funkcja zdejmuje element z konca kolejki.

Parameters

in	a	- wskaznik do ktorego przypisywany jest element zdejmowany, aby mogl zostal uzyty w przyszlosci
----	---	---

Returns

(brak)

Definition at line 18 of file kolejka_lista.cpp.

4.2.2.2 void kolejkalista::enqueue (int el_dodawany)

Dodanie elementu z kolejki Funkcja dodaje element na poczatek kolejki.

Parameters

in	el_dodawany	- zmienna stala, ktora ma zostac dodana poczatek kolejki
----	-------------	--

Returns

(brak)

Definition at line 14 of file kolejka_lista.cpp.

4.2.2.3 bool kolejalista::isempty ()

Sprawdzanie pojemności w kolejce. Funkcja sprawdza czy jest pusta poprzez porównanie liczby_elementów do 0.

Returns

true jeżeli funkcja jest pusta, w przeciwnym wypadku false.

Definition at line 30 of file kolejka_lista.cpp.

4.2.2.4 int kolejalista::size ()

Zwrócenie rozmiaru funkcji. Funkcja sprawdza ile elementów jest w kolejce.

Returns

Zwraca liczbę elementów.

Definition at line 34 of file kolejka_lista.cpp.

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

- C:/Users/Ania/workspace/zadanie/inc/kolejka_lista.hh
- C:/Users/Ania/workspace/zadanie/src/kolejka_lista.cpp

4.3 kolejkatab Class Reference

Klasa modeluje pojęcie kolejki, bazującej na tablicy. Wykonywane są operacje dodawania do kolejki, zdjęcia z kolejki, sprawdzenia czy jest pusta oraz sprawdzenia rozmiaru. Tablice zaalokowałam w sposób dynamiczny. Dodatkowo użyte niektóre funkcje pomocnicze jak np. powiększenie tablicy.

```
#include <kolejka.hh>
```

Public Member Functions

- [kolejkatab](#) ()
Konstruktor bezargumentowy. Przy tworzeniu tablica jest pusta, w konstruktorze przydzielana jest dynamicznie, na początku tablica ma możliwość przypisania tablicy o rozmiarze 1. Jeżeli nie uda się utworzyć tablicy wyrzucany jest błąd. Przy wywołaniu konstruktora bezparametrycznego powiększająca tablica będzie dwa razy większa.
- void [enqueue](#) (int el_dodawany)
Dodanie elementu z kolejki Funkcja dodaje element na początek kolejki, jeżeli brakuje miejsca podwaja pamięć.
- void [dequeue](#) (int *a)
Zdejmowanie elementu z kolejki. Funkcja zdejmuje element z końca kolejki, jeżeli liczba elementów będzie mniejsza bądź równa 1/4 możliwych elementów w tablicy, zostaje ona pomniejszana.
- bool [isempty](#) ()
Sprawdzanie pojemności w kolejce. Funkcja sprawdza czy jest pusta poprzez porównanie liczby_elementów do 0.
- int [size](#) ()
Sprawdzenie rozmiaru funkcji Funkcja sprawdza ile elementów jest w kolejce.

4.3.1 Detailed Description

Klasa modeluje pojęcie kolejki, bazującej na tablicy. Wykonywane są operacje dodawania do kolejki, zdjęcia z kolejki, sprawdzenia czy jest pusta oraz sprawdzenia rozmiaru. Tablice zaalokowałam w sposób dynamiczny. Dodatkowo użyte niektóre funkcje pomocnicze jak np. powiększenie tablicy.

Definition at line 29 of file kolejka.hh.

4.3.2 Constructor & Destructor Documentation

4.3.2.1 kolejkatb::kolejkatab () [inline]

Konstruktor bezargumentowy. Przy tworzeniu tablica jest pusta, w konstruktorze przydzielana jest dynamicznie, na początku tablica ma mozliwosc przypisania tablicy o rozmiarze 1. Jezeli nie uda sie utworzyc tablicy wyrzucany jest blad. Przy wywoaniu konstrktora bezparametrycznego powiekszajaca tablica bedzie dwa razy wieksza.

Definition at line 79 of file kolejka.hh.

4.3.3 Member Function Documentation

4.3.3.1 void kolejkatb::dequeue (int * a)

Zdejmowanie elementu z kolejki. Funkcja zdejmuje element z konca kolejki, jezeli liczba elementow bedzie mniejsza badz rowna 1/4 mozliwych elementow w tablicy, zostaje ona pomniejszana.

Parameters

in	a	- wskaznik do ktorego przypisywany jest element zdejmowany, aby mogl zostal uzyty w przyszlosci
----	---	---

Returns

(brak)

Definition at line 58 of file kolejka.cpp.

4.3.3.2 void kolejkatb::enqueue (int el_dodawany)

Dodanie elementu z kolejki Funkcja dodaje element na poczatek kolejki, jezeli brakuje miejsca podwaja pamiec.

Parameters

in	el_dodawany	- zmienna stała, ktora ma zostac dodana na koniec stosu
----	-------------	---

Returns

(brak)

Definition at line 39 of file kolejka.cpp.

4.3.3.3 bool kolejkatb::isempty ()

Sprawdzanie pojemnosci w kolejce. Funkcja sprawdza czy jest pusta poprzez porownanie liczby_elementow do 0.

Returns

true jezeli funkcja jest pusta, w przeciwnym wypadku false.

Definition at line 73 of file kolejka.cpp.

4.3.3.4 int kolejkatb::size ()

Sprawdzenie rozmiaru funkcji Funkcja sprawdza ile elementow jest w kolejce.

Returns

(brak)

Definition at line 77 of file kolejka.cpp.

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

- C:/Users/Ania/workspace/zadanie/inc/kolejka.hh
- C:/Users/Ania/workspace/zadanie/src/kolejka.cpp

4.4 stos Class Reference

Klasa modeluje pojecie stosu,bazujacego na tablicy. Na stosie wyonywane sa operacje dodatnia na stos, zdjecia ze stosu, sprawdzenia czy jest pusty oraz sprawdzenia rozmiaru. Tablice zaalokowalam w sposob dynamiczny. Dodatkowo uyte niektore funkcje pomocnicze jak np. powiekszenie stosu.

```
#include <stos.hh>
```

Public Member Functions

- [stos \(\)](#)
Konstruktor bezargumentowy. Przy tworzeniu tablica jest pusta, w konstruktorze przydzielana jest dynamicznie, na poczatku tablica ma mozliwosc przypisania tablicy o rozmiarze 1. Jezeli nie uda sie utworzyc tablicy wyrzucany jest blad. Przy wywoaniu konstrktora bezparametrycznego powiekszajaca tablica bedzie dwa razy wieksza.
- [stos \(int cos\)](#)
Konstruktor parametryczny. Przy tworzeniu tablica jest pusta, w konstruktorze przydzielana jest dynamicznie, na poczatku tablica ma mozliwosc przypisania tablicy o rozmiarze 1. Jezeli nie uda sie utworzyc tablicy wyrzucany jest blad. Przy wywoaniu konstrktora parametrycznego tablica zostanie powiekszona o 1 element.
- void [push](#) (int el_dodawany)
Dodanie elementu na stos. Funkcja dodaje element na koniec stosu, jezeli brakuje miejsca powieksza pamiec.
- void [pop](#) (int *a)
Zdejmowanie elementu ze stosu. Funkcja zdejmuje element z konca stosu, jezeli liczba elementow bedzie mniejsza badz rowna 1/4 mozliwych elementow w tablicy, zostaje ona pomniejszana.
- bool [isempty](#) ()
Zwrocenie rozmiaru stosu Funkcja sprawdza czy stos jest pusty poprzez porownanie liczby_elementow do 0.
- int [size](#) ()
Sprawdzenie rozmiaru funkcji Funkcja sprawdza ile elementow jest na stosie.

4.4.1 Detailed Description

Klasa modeluje pojecie stosu,bazujacego na tablicy. Na stosie wyonywane sa operacje dodatnia na stos, zdjecia ze stosu, sprawdzenia czy jest pusty oraz sprawdzenia rozmiaru. Tablice zaalokowalam w sposob dynamiczny. Dodatkowo uyte niektore funkcje pomocnicze jak np. powiekszenie stosu.

Definition at line 29 of file stos.hh.

4.4.2 Constructor & Destructor Documentation

4.4.2.1 `stos::stos () [inline]`

Konstruktor bezargumentowy. Przy tworzeniu tablica jest pusta, w konstruktorze przydzielana jest dynamicznie, na poczatku tablica ma mozliwosc przypisania tablicy o rozmiarze 1. Jezeli nie uda sie utworzyc tablicy wyrzucany jest blad. Przy wywoaniu konstrktora bezparametrycznego powiekszajaca tablica bedzie dwa razy wieksza.

Definition at line 79 of file stos.hh.

4.4.2.2 `stos::stos (int cos) [inline]`

Konstruktor parametryczny. Przy tworzeniu tablica jest pusta, w konstruktorze przydzielana jest dynamicznie, na początku tablica ma mozliwosc przypisania tablicy o rozmiarze 1. Jezeli nie uda sie utworzyc tablicy wyrzucany jest blad. Przy wywoaniu konstrktora parametrycznego tablica zostanie powiekszona o 1 element.

Definition at line 96 of file `stos.hh`.

4.4.3 Member Function Documentation

4.4.3.1 `bool stos::isempty ()`

Zwrocenie rozmiaru stosu Funkcja sprawdza czy stos jest pusty poprzez porownanie `liczby_elementow` do 0.

Returns

true jezeli funkcja jest pusta, w przeciwnym wypadku false.

Definition at line 65 of file `stos.cpp`.

4.4.3.2 `void stos::pop (int * a)`

Zdejmowanie elementu ze stosu. Funkcja zdejmuje element z konca stosu, jezeli liczba elementow bedzie mniejsza badz rowna 1/4 mozliwych elementow w tablicy, zostaje ona pomniejszana.

Parameters

<code>in</code>	<code>a</code>	- wskaznik do ktorego przypisywany jest element zdejmowany, aby mogl zostal uzyty w przyszlosci
-----------------	----------------	---

Returns

(brak)

Definition at line 50 of file `stos.cpp`.

4.4.3.3 `void stos::push (int el_dodawany)`

Dodanie elementu na stos. Funkcja dodaje element na koniec stosu, jezeli brakuje miejsca powieksza pamiec.

Parameters

<code>in</code>	<code>el_dodawany</code>	- zmienna stała, ktora ma zostac dodana na koniec stosu
-----------------	--------------------------	---

Returns

(brak)

Definition at line 41 of file `stos.cpp`.

4.4.3.4 `int stos::size ()`

Sprawdzenie rozmiaru funkcji Funkcja sprawdza ile elementow jest na stosie.

Returns

Zwraca liczbe elementow znajdujacych sie na stosie

Definition at line 69 of file stos.cpp.

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

- C:/Users/Ania/workspace/zadanie/inc/[stos.hh](#)
- C:/Users/Ania/workspace/zadanie/src/[stos.cpp](#)

4.5 stoslista Class Reference

Klasa modeluje pojecie stosu,bazujacego na liscie. Na stosie wyonywane sa operacje dodatnia na stos, zdjecia ze stosu, sprawdzenia czy jest pusty oraz sprawdzenia rozmiaru.

```
#include <stos_lista.hh>
```

Public Member Functions

- void [push](#) (int el_dodawany)
Dodanie elementu na stos. Funkcja dodaje element na koniec stosu.
- void [pop](#) (int *a)
Zdejmowanie elementu ze stosu. Funkcja zdejmuje element z konca stosu.
- bool [isempty](#) ()
Sprawdzenie czy funkcja jest pusta Funkcja sprawdza czy stos jest pusty poprzez uzycie funkcji size ktora dziala na liscie.
- int [size](#) ()
Zwrocenie rozmiaru funkcji Funkcja sprawdza ile elementow jest na stosie.

4.5.1 Detailed Description

Klasa modeluje pojecie stosu,bazujacego na liscie. Na stosie wyonywane sa operacje dodatnia na stos, zdjecia ze stosu, sprawdzenia czy jest pusty oraz sprawdzenia rozmiaru.

Definition at line 25 of file stos_lista.hh.

4.5.2 Member Function Documentation

4.5.2.1 bool stoslista::isempty ()

Sprawdzenie czy funkcja jest pusta Funkcja sprawdza czy stos jest pusty poprzez uzycie funkcji size ktora dziala na liscie.

Returns

true jezeli funkcja jest pusta, w przeciwnym wypadku false.

Definition at line 29 of file stos_lista.cpp.

4.5.2.2 void stoslista::pop (int * a)

Zdejmowanie elementu ze stosu. Funkcja zdejmuje element z konca stosu.

Parameters

<code>in</code>	<code>a</code>	- wskaźnik do którego przypisywany jest element zdejmowany, aby mógł zostać użyty w przyszłości
-----------------	----------------	---

Returns

(brak)

Definition at line 17 of file `stos_lista.cpp`.4.5.2.3 `void stoslista::push (int el_dodawany)`

Dodanie elementu na stos. Funkcja dodaje element na koniec stosu.

Parameters

<code>in</code>	<code>el_dodawany</code>	- zmienna stała, która ma zostać dodana na koniec stosu
-----------------	--------------------------	---

Returns

(brak)

Definition at line 12 of file `stos_lista.cpp`.4.5.2.4 `int stoslista::size ()`

Zwrócenie rozmiaru funkcji Funkcja sprawdza ile elementów jest na stosie.

Returns

Zwraca liczbę elementów znajdujących się na stosie.

Definition at line 33 of file `stos_lista.cpp`.

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

- `C:/Users/Ania/workspace/zadanie/inc/stos_lista.hh`
- `C:/Users/Ania/workspace/zadanie/src/stos_lista.cpp`

4.6 zegar Class Reference

Klasa modeluje uruchomienia głównych właściwości programu. Atrybutem klasy są stworzone dwa elementy klasy dane, na których wykonywane są działania.

```
#include <uruchom.hh>
```

Public Member Functions

- `zegar ()`
- `void wczytaj_dane_pod (string nazwa_pliku_pod)`
Wczytanie danych podstawowych Funkcja wczytuje dane, na których wykonywany jest algorytm. Dane te są główną funkcją programu.
- `void wczytaj_dane_spr (string nazwa_pliku_spr)`
Wczytanie danych sprawdzających Funkcja wczytuje dane, na które zostają porównane z danymi na których został wykonany algorytm.

- void `algorytm` ()
Funkcja wykonujca zadany algorytm na wektorze. Funkcja wykonuje zadany algorytm na wektorze wejsciowym. W naszym przypadku wektor pomnoony jest przez sta liczbe
- bool `porownaj` ()
Funkcja porjca dwa wektory. Funkcja porje dwa wektory, sprawdza czy wszystkie elemnty s ze sob r.
- LARGE_INTEGER `wlaczStoper` ()
*Funkcja zapamietujca czas poczatkowy. Funkcja naleca do biblioteki "windows.h", stoper zostaje wczony. Funkcja naleca do funkcji bool `QueryPerformanceCounter(_out LARGE_INTEGER *lpPerformanceCount)`, funkcja ta zwraca wartosc niezerowa jeeli waczenie zako sikcesem, natomiast w przeciwnym wypadku zostanie wyrzucony bd i zwrartosc 0. Dla komputerltiprocessorowych nie ma znaczenia, ktest uywany, mog jedynie rnimalnie czasy.*
- LARGE_INTEGER `wylaczStoper` ()
*Funkcja zapamietujca czas koy. Funkcja naleca do biblioteki "windows.h", stoper zostaje wyczony, aby zost zmieryzony czas wykonania algorytmu w programie, poprzez odje czasu pocztkowego od czasu koego. Funkcja naleca do funkcji bool `QueryPerformanceCounter(_out LARGE_INTEGER *lpPerformanceCount)`, funkcja ta zwraca wartosc niezerowa jeeli wyaczenie zako sikcesem, natomiast w przeciwnym wypadku zostanie wyrzucony bd i zwrartosc 0. Dla komputerltiprocessorowych nie ma znaczenia, ktest uywany, mog jedynie rnimalnie czasy.*
- void `wczytaj` (string nazwa)
Funkcja wczytujca dane do stosu. Funkcja otwiera plik zdefiniowany w glownej funkcji przez uytkownika, sprawdza czy plik zost otwarty, jeeli zost otwarty wczytana jest liczba elementow pliku, nastepnie wczytywane sa wszystkie liczby do tablicy.

4.6.1 Detailed Description

Klasa modeluje uruchomienia gwnych wlasciowosci programu. Atrybutem klasy sa stowrzone dwa elemnty klasy dane, na ktorych wykonywane sa dzialania.

Definition at line 28 of file `uruchom.hh`.

4.6.2 Constructor & Destructor Documentation

4.6.2.1 `zegar::zegar ()` [`inline`]

Definition at line 45 of file `uruchom.hh`.

4.6.3 Member Function Documentation

4.6.3.1 `void zegar::algorytm ()`

Funkcja wykonujca zadany algorytm na wektorze. Funkcja wykonuje zadany algorytm na wektorze wejsciowym. W naszym przypadku wektor pomnoony jest przez sta liczbe

Returns

(brak)

Definition at line 14 of file `uruchom.cpp`.

4.6.3.2 `bool zegar::porownaj ()` [`inline`]

Funkcja porjca dwa wektory. Funkcja porje dwa wektory, sprawdza czy wszystkie elemnty s ze sob r.

Returns

Funkcja zwraca true jezeli wektory sa jednakowe w przeciwnym wypadku zostaje wzrocony false.

Definition at line 83 of file `uruchom.hh`.

4.6.3.3 void zegar::wczytaj (string nazwa)

Funkcja wczytująca dane do stosu. Funkcja otwiera plik zdefiniowany w głównej funkcji przez użytkownika, sprawdza czy plik został otwarty, jeżeli został otwarty wczytana jest liczba elementów pliku, następnie wczytywane są wszystkie liczby do tablicy.

Parameters

in	<i>nazwa</i>	- zmienna, która jest wprowadzona przez użytkownika do programu
----	--------------	---

Returns

(brak)

Definition at line 38 of file uruchom.cpp.

4.6.3.4 void zegar::wczytaj_dane_pod (string nazwa_pliku_pod) [inline]

Wczytanie danych podstawowych Funkcja wczytuje dane, na których wykonywany jest algorytm. Dane te są główną funkcją programu.

Parameters

in	<i>nazwa_pliku_pod</i>	- jest to zmienna która jest ciągiem znaków (nazwa pliku), który ma zostać otwarty.
----	------------------------	---

Returns

(brak)

Definition at line 57 of file uruchom.hh.

4.6.3.5 void zegar::wczytaj_dane_spr (string nazwa_pliku_spr) [inline]

Wczytanie danych sprawdzających Funkcja wczytuje dane, na które zostają porównane z danymi na których został wykonany algorytm.

Parameters

in	<i>nazwa_pliku_spr</i>	- jest to zmienna która jest ciągiem znaków (nazwa pliku), który ma zostać otwarty, w celu porównania.
----	------------------------	--

Returns

(brak)

Definition at line 68 of file uruchom.hh.

4.6.3.6 LARGE_INTEGER zegar::włączStoper ()

Funkcja zapamiętująca czas początkowy. Funkcja należy do biblioteki "windows.h", stoper zostaje włączony. Funkcja należy do funkcji bool QueryPerformanceCounter(_out LARGE_INTEGER *lpPerformanceCount), funkcja ta zwraca wartość niezerową jeżeli włączenie zakończy się sukcesem, natomiast w przeciwnym wypadku zostanie wyrzucony błąd i zwróci 0. Dla komputerów 32-bitowych nie ma znaczenia, ktest używany, mogą jedynie zmierzyć czas.

Definition at line 20 of file uruchom.cpp.

4.6.3.7 LARGE_INTEGER zegar::wylaczStoper ()

Funkcja zapamiętująca czas kł. Funkcja należy do biblioteki "windows.h", stoper zostaje wyczony, aby został zmierzony czas wykonania algorytmu w programie, poprzez odjęcie czasu początkowego od czasu końcowego. Funkcja należy do funkcji bool QueryPerformanceCounter(_out LARGE_INTEGER *lpPerformanceCount), funkcja ta zwraca wartość niezerową jeżeli wywołanie zakończy się sukcesem, natomiast w przeciwnym wypadku zostanie wyrzucony błąd i zwróci wartość 0. Dla komputerów wieloprotocowych nie ma znaczenia, który jest używany, mogą jedynie różnić się czasy.

Definition at line 29 of file uruchom.cpp.

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

- [C:/Users/Ania/workspace/zadanie/inc/uruchom.hh](#)
- [C:/Users/Ania/workspace/zadanie/src/uruchom.cpp](#)

Chapter 5

File Documentation

5.1 C:/Users/Ania/workspace/zadanie/doc/pages/strona.dox File Reference

5.2 C:/Users/Ania/workspace/zadanie/inc/dane.hh File Reference

Definicja klasy dane.

```
#include <iostream>
#include <string>
#include <vector>
```

Classes

- class [dane](#)

Modeluje pojecie danych, uitych w programie, kt moim przypadku sa wektorem, który zostaje wczytany z pliku. Pierwsza zmienna wczytana z pliku jest liczba elemntow wystepujacych w tym pliku. Przyjelam, ze zmienne wczytane z pliku sa liczbami calkowitymi (int).

5.2.1 Detailed Description

Definicja klasy dane. Plik zawiera definicje klasy dane, która jest klasa podstawowa programu

Definition in file [dane.hh](#).

5.3 C:/Users/Ania/workspace/zadanie/inc/kolejka.hh File Reference

Definicja klasy kolejkatab.

```
#include "uruchom.hh"
```

Classes

- class [kolejkatab](#)

Klasa modeluje pojecie kolejki, bazujacej na tablicy. Wykonywane sa operacje dodawnia do kolejki, zdjecia z kolejki, sprawdzenia czy jest pusta oraz sprawdzenia rozmiaru. Tablice zaalokowalam w sposob dynamiczny. Dodatkowo uyte niektore funkcje pomocnicze jak np. powiekszenie tablicy.

5.3.1 Detailed Description

Definicja klasy kolejkat. Plik zawiera definicję klasy kolejkat, w której wykorzystana jest tablica do zapisania elementów w kolejce.

Definition in file [kolejka.hh](#).

5.4 C:/Users/Ania/workspace/zadanie/inc/kolejka_lista.hh File Reference

Definicja klasy kolejkalista.

```
#include <list>
#include <iostream>
#include <cstdlib>
```

Classes

- class [kolejkalista](#)

Klasa modeluje pojecie kolejki, bazujacej na liscie. Wykonywane sa operacje dodawania do kolejki, zdjecia z kolejki, sprawdzenia czy jest pusta oraz sprawdzenia rozmiaru. Tablice zaalokowalam w sposob dynamiczny.

5.4.1 Detailed Description

Definicja klasy kolejkalista. Plik zawiera definicję klasy kolejkalista, w której wykorzystana jest lista do zapisania elementów w kolejce.

Definition in file [kolejka_lista.hh](#).

5.5 C:/Users/Ania/workspace/zadanie/inc/stos.hh File Reference

Definicja klasy stos.

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

Classes

- class [stos](#)

Klasa modeluje pojecie stosu, bazujacego na tablicy. Na stosie wyonywane sa operacje dodatnia na stos, zdjecia ze stosu, sprawdzenia czy jest pusty oraz sprawdzenia rozmiaru. Tablice zaalokowalam w sposob dynamiczny. Dodatkowo uyte niektore funkcje pomocnicze jak np. powiekszenie stosu.

5.5.1 Detailed Description

Definicja klasy stos. Plik zawiera definicję klasy stos, która do zapisu i zapamiętania liczb używa tablice.

Definition in file [stos.hh](#).

5.6 C:/Users/Ania/workspace/zadanie/inc/stos_lista.hh File Reference

Definicja klasy stoslisa.

```
#include <list>
#include <iostream>
#include <cstdlib>
```

Classes

- class [stoslisa](#)

Klasa modeluje pojecie stosu, bazujacego na liscie. Na stosie wyonywane sa operacje dodatnia na stos, zdjecia ze stosu, sprawdzenia czy jest pusty oraz sprawdzenia rozmiaru.

5.6.1 Detailed Description

Definicja klasy stoslisa. Plik zawiera definicje klasy stoslisa, ktora do zapisu i zapamietania liczb uzywa listy.

Definition in file [stos_lista.hh](#).

5.7 C:/Users/Ania/workspace/zadanie/inc/uruchom.hh File Reference

Definicja klasy zegar.

```
#include "dane.hh"
#include <windows.h>
#include "stos.hh"
```

Classes

- class [zegar](#)

Klasa modeluje uruchomienia gownych wlasciowosci programu. Atrybutem klasy sa stowrzone dwa elemnty klasy dane, na ktorych wykonywane sa dzialania.

5.7.1 Detailed Description

Definicja klasy zegar. Plik zawiera definicje klasy zegar, ktora jest klasa glowna programu. Klasa ta jest pochodna i specjalizacja klasy dane.

Definition in file [uruchom.hh](#).

5.8 C:/Users/Ania/workspace/zadanie/src/dane.cpp File Reference

```
#include "dane.hh"
#include <fstream>
```

5.9 C:/Users/Ania/workspace/zadanie/src/kolejka.cpp File Reference

```
#include "kolejka.hh"
```

5.10 C:/Users/Ania/workspace/zadanie/src/kolejka_lista.cpp File Reference

```
#include "kolejka_lista.hh"
```

5.11 C:/Users/Ania/workspace/zadanie/src/main.cpp File Reference

Funkcja glowna ktorej glownym zalozeniem jest wczytanie plikow z rozna wielkoscia elementow znajdujacych siliku, obliczenie sredniej wartosci czasu, w jakim zostaje wykonany algorytm (w naszym przypadku pomnozenie przez 2), nastepnie program porownuje poprawnosc wykoannia mnozenia z plikiem sprawdzajacym. Uzytkownik musi w programie zdefiniowac: liczbe powtorzen (zmienna j), ilosc plikow - do ilu wykonywane jest mnozenie (zmienna i), nazwy plikow (string czesc_1, i, czesc_2 - wszystko opcjonalnie).

```
#include "uruchom.hh"  
#include <sstream>  
#include <fstream>
```

Functions

- int [main](#) ()

5.11.1 Detailed Description

Funkcja glowna ktorej glownym zalozeniem jest wczytanie plikow z rozna wielkoscia elementow znajdujacych siliku, obliczenie sredniej wartosci czasu, w jakim zostaje wykonany algorytm (w naszym przypadku pomnozenie przez 2), nastepnie program porownuje poprawnosc wykoannia mnozenia z plikiem sprawdzajacym. Uzytkownik musi w programie zdefiniowac: liczbe powtorzen (zmienna j), ilosc plikow - do ilu wykonywane jest mnozenie (zmienna i), nazwy plikow (string czesc_1, i, czesc_2 - wszystko opcjonalnie).

Returns

(brak)

Definition in file [main.cpp](#).

5.11.2 Function Documentation

5.11.2.1 int main ()

Definition at line 19 of file main.cpp.

5.12 C:/Users/Ania/workspace/zadanie/src/stos.cpp File Reference

```
#include "stos.hh"
```

5.13 C:/Users/Ania/workspace/zadanie/src/stos_lista.cpp File Reference

```
#include "stos_lista.hh"
```

5.14 C:/Users/Ania/workspace/zadanie/src/uruchom.cpp File Reference

```
#include "uruchom.hh"  
#include <fstream>
```

Index

algorytm
zegar, 18

C:/Users/Ania/workspace/zadanie/doc/pages/strona.-
dox, 21

C:/Users/Ania/workspace/zadanie/inc/dane.hh, 21

C:/Users/Ania/workspace/zadanie/inc/kolejka.hh, 21

C:/Users/Ania/workspace/zadanie/inc/kolejka_lista.hh,
22

C:/Users/Ania/workspace/zadanie/inc/stos.hh, 22

C:/Users/Ania/workspace/zadanie/inc/stos_lista.hh, 23

C:/Users/Ania/workspace/zadanie/inc/uruchom.hh, 23

C:/Users/Ania/workspace/zadanie/src/dane.cpp, 23

C:/Users/Ania/workspace/zadanie/src/kolejka.cpp, 24

C:/Users/Ania/workspace/zadanie/src/kolejka_lista.cpp,
24

C:/Users/Ania/workspace/zadanie/src/main.cpp, 24

C:/Users/Ania/workspace/zadanie/src/stos.cpp, 24

C:/Users/Ania/workspace/zadanie/src/stos_lista.cpp, 25

C:/Users/Ania/workspace/zadanie/src/uruchom.cpp, 25

dane, 7

- dodaj_element, 8
- dodaj_elementy, 8
- odwroc_element, 8
- operator+, 8
- operator=, 8
- operator==, 9
- size, 9
- wczytaj, 9
- wnetrze, 10
- wypisz, 10
- zamien_element, 10

dequeue

- kolejkalista, 11
- kolejkatab, 13

dodaj_element

- dane, 8

dodaj_elementy

- dane, 8

enqueue

- kolejkalista, 11
- kolejkatab, 13

isempty

- kolejkalista, 11
- kolejkatab, 13
- stos, 15
- stoslista, 16

kolejkalista, 11

- dequeue, 11
- enqueue, 11
- isempty, 11
- size, 12

kolejkatab, 12

- dequeue, 13
- enqueue, 13
- isempty, 13
- kolejkatab, 13
- size, 13

main

- main.cpp, 24

main.cpp

- main, 24

odwroc_element

- dane, 8

operator+

- dane, 8

operator=

- dane, 8

operator==

- dane, 9

pop

- stos, 15
- stoslista, 16

porownaj

- zegar, 18

push

- stos, 15
- stoslista, 17

size

- dane, 9
- kolejkalista, 12
- kolejkatab, 13
- stos, 15
- stoslista, 17

stos, 14

- isempty, 15
- pop, 15
- push, 15
- size, 15
- stos, 14

stoslista, 16

- isempty, 16
- pop, 16

- push, [17](#)
- size, [17](#)
- wczytaj
 - dane, [9](#)
 - zegar, [18](#)
- wczytaj_dane_pod
 - zegar, [19](#)
- wczytaj_dane_spr
 - zegar, [19](#)
- wlaczStoper
 - zegar, [19](#)
- wnetrze
 - dane, [10](#)
- wylaczStoper
 - zegar, [19](#)
- wypisz
 - dane, [10](#)
- zamien_element
 - dane, [10](#)
- zegar, [17](#)
 - algorytm, [18](#)
 - porownaj, [18](#)
 - wczytaj, [18](#)
 - wczytaj_dane_pod, [19](#)
 - wczytaj_dane_spr, [19](#)
 - wlaczStoper, [19](#)
 - wylaczStoper, [19](#)
 - zegar, [18](#)