PAMSI LAB I

0.3

Wygenerowano przez Doxygen 1.8.6

Śr, 11 mar 2015 22:05:45

Spis treści

Rozdział 1

Strona główna

Autor

Bartłomiej Ankowski

Data

09.03.2015

Wersja

0.3

1.1 Programu

Program jest przykładowym narzędziem do sprawdzania złożoności obliczeniowej i wpływu zwiekszania ilości przetwarzanych danych na czas wykonywania operacji. W swojej funkcjonalnosci oferuje odczytywanie i zapisywanie danych przechowywanych w wektorze do pliku zewnetrzego o nazwie podanej przez uzytkownika.

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 Interfejs

Modeluje Interfejs.

`#include <Interfejs.h>`

Diagram dziedziczenia dla Interfejs

Metody publiczne

- virtual void zapisz (string PlikOut)=0

  Metoda zapisujaca dane do pliku zewnetrznego.
- virtual void odczyt (string plikIn)=0

  Metoda odczytujaca dane z pliku zewnetrzengo.
- virtual void dzialanie ()=0

  Metoda wykonujaca pewna operacje algebraiczna.

5.1.1 Opis szczegółowy

Modeluje Interfejs.

Jest klasa czysta abstrakcyjna,bazowa dla Wektora

Definicja w linii 24 pliku Interfejs.h.

5.1.2 Dokumentacja funkcji składowych

5.1.2.1 virtual void Interfejs::dzialanie ( ) [pure virtual]

Metoda wykonujaca pewna operacje algebraiczna.

Metoda ma za zadanie wykonac na obiekcie danej klasy zaimplementowana w niej operacje algebraiczna

Implementowany w Wektor.

5.1.2.2 virtual void Interfejs::odczyt (string *plikIn*) [pure virtual]

Metoda odczytujaca dane z pliku zewnetrzengo.

Metoda ma za zadanie odczytac dane z pliku zewnetrzego i umiescic dane w obiekcie danej klasy

Parametry

in	*PlikIn*	- Nazwa pliku wejsciowego

Implementowany w Wektor.

5.1.2.3 virtual void Interfejs::zapisz (string *PlikOut*) [pure virtual]

Metoda zapisujaca dane do pliku zewnetrznego.

Metoda ma za zadanie przekazac zawartosc danych do pliku zewnetrzego

Parametry

in	*PlikOut*	- Nazwa pliku zewnetrzenego

Implementowany w Wektor.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- Interfejs.h

5.2 Dokumentacja klasy Wektor

Modeluje pojecie Wektora.

`#include <Wektor.h>`

Diagram dziedziczenia dla Wektor

Diagram współpracy dla Wektor:

Metody publiczne

- Wektor (int Rozmiar)

  Konstruktor.
- Wektor (Wektor &W)

  Konstruktor kopiujący Wektora.
- virtual void zapisz (string PlikOut)

  Metoda odczytujaca dane z pliku zewnetrzengo.
- virtual void odczyt (string PlikIn)

  Metoda odczytujaca dane z pliku zewnetrzengo.
- bool operator= (const Wektor &W)

  Operator przypisania.
- virtual void dzialanie ()

  Metoda wykonujaca mnozenie.
- ~Wektor ()

  Destruktor.

Atrybuty prywatne

- int * _W

  Pola Wektora.
- int _Rozmiar

  Pola Wektora.

Przyjaciele

- ostream & operator<< (ostream &StrmWyj, const Wektor &W)

  Opeartor przesuniecia bitowego w lewo.
- istream & operator>> (istream &StrmWej, Wektor &W)

  Operatro przesuniecia bitowego w prawo.

5.2.1 Opis szczegółowy

Modeluje pojecie Wektora.

Klasa modeluje pojecie wektora jako dynamiczna tablice jednowymiarowa zawieracja dane typu calkowitego,jest klasa dziedziczaca od klasy Interfejs

Definicja w linii 21 pliku Wektor.h.

5.2.2 Dokumentacja konstruktora i destruktora

5.2.2.1 Wektor::Wektor (int *Rozmiar*)

Konstruktor.

Konstruktor ma za zadanie pobrać rozmiar oraz zaalokować odpowiednią ilość pamięci dla wektora.

Parametry

in	*Rozmiar*	- Wielkość odpowiadająca podanemu przez użytkownuka rozmiarowi układu. Jest używana do zaalokowania odpowiedniej ilości pamięci oraz sterowania pętlami

Definicja w linii 9 pliku Wektor.cpp.

5.2.2.2 Wektor::Wektor (Wektor & *W*)

Konstruktor kopiujący Wektora.

Konstruktor kopiujący jest niezbędny w przypadku gdy w klasach występują pola wskaźnikowe, gdyż w przypadku gdy posiadamy jedynie konstruktor kopiujacy automatyczny każdy nowo powstały obiekt jest dokładną kopią obiektu wzorcowego, gdyż wskaźniki pokazują w to samo miejsce i to one są kopiowane

Parametry

in	*W*	- Wektor wzorcowy

Definicja w linii 16 pliku Wektor.cpp.

5.2.2.3 Wektor::~Wektor ( ) [inline]

Destruktor.

Destruktor Wektora zwalniający zaalokowaną pamięć

Definicja w linii 113 pliku Wektor.h.

5.2.3 Dokumentacja funkcji składowych

5.2.3.1 void Wektor::dzialanie ( ) [virtual]

Metoda wykonujaca mnozenie.

Metoda ma zadanie wykonac mnozenie przez 2 odpowiednia ilosc danych oraz zmierzyc czas wykonywania tych operacji dla 10 powtorzen dla kazdej ilosci paczki danych

Implementuje Interfejs.

Definicja w linii 66 pliku Wektor.cpp.

Oto graf wywoływań tej funkcji:

5.2.3.2 void Wektor::odczyt (string *PlikIn*) [virtual]

Metoda odczytujaca dane z pliku zewnetrzengo.

Metoda ma za zadanie odczytac dane z pliku zewnetrzego i umiescic dane w wektorze

Parametry

in	*PlikIn*	- Nazwa pliku wejsciowego

Implementuje Interfejs.

Definicja w linii 47 pliku Wektor.cpp.

Oto graf wywoływań tej funkcji:

5.2.3.3 bool Wektor::operator= (const Wektor & *W*)

Operator przypisania.

Przeciazenie operatora podstawienia umozliwia przypisanie jednego wektora do drugiego bez koniecznosci operowania na poszczegolnych polach

Parametry

in	*W*	- Wektor ktory chcemy przypisac

Zwraca

Jesli operacja przypisania sie powiedzi zwrocona zostaenie wartosc true

Definicja w linii 24 pliku Wektor.cpp.

5.2.3.4 void Wektor::zapisz (string *PlikOut*) [virtual]

Metoda odczytujaca dane z pliku zewnetrzengo.

Metoda ma za zadanie zapisac dane przechowywane w dynamicznej tablicy do pliku zewnetrznego o nazwie podanej jako argument wywolania metody

Parametry

in	*PlikIn*	- Nazwa pliku wejsciowego

Implementuje Interfejs.

Definicja w linii 31 pliku Wektor.cpp.

5.2.4 Dokumentacja przyjaciół i funkcji związanych

5.2.4.1 ostream& operator<< (ostream & *StrmWyj,* const Wektor & *W*) [friend]

Opeartor przesuniecia bitowego w lewo.

Funkcja umozliwa wyswietlenie wczytanego wyrazenia na standardowym wyjsciu

Parametry

in	*StrmWyj-*	referencja strumienia wyjsciowego na ktorym ma byc wyswietlony symbol
in	*W*	- obiekt ktory ma zostac wyswietlony na strumieniu

Zwraca

funkcja zwraca referencje do strumienia, na ktorym ma zostac wyswietlony obiekt

Definicja w linii 132 pliku Wektor.h.

5.2.4.2 istream& operator>> (istream & *StrmWej,* Wektor & *W*) [friend]

Operatro przesuniecia bitowego w prawo.

Funckja umozliwia wczytanie Wektora ze standardowego wejscia

Parametry

in	*StrmWej*	- referencja strumienia wejsciowego z ktorego ma byc wczytany symbol
in	*W*	- referencja Wektora do ktorego wczytywane sa jego poszczegolne elementy jesli sa z odpowiedniego zakresu

Zwraca

funckja zwraca referencje do strumienia

Definicja w linii 156 pliku Wektor.h.

5.2.5 Dokumentacja atrybutów składowych

5.2.5.1 int Wektor::_Rozmiar [private]

Pola Wektora.

pole przechowujące rozmiar Wektora

Definicja w linii 36 pliku Wektor.h.

5.2.5.2 int* Wektor::_W [private]

Pola Wektora.

Pole zawiera wskaźnik na int służacy do pokazania na zaalokowana dynamicznie pamieć

Definicja w linii 30 pliku Wektor.h.

Dokumentacja dla tej klasy została wygenerowana z plików:

- Wektor.h
- Wektor.cpp

Rozdział 6

Dokumentacja plików

6.1 Dokumentacja pliku Interfejs.h

Definicja klasy Interfejs.

```
#include <iostream>
#include <cstdlib>
#include <sstream>
#include <cstring>
#include <fstream>
#include <time.h>
```

Wykres zależności załączania dla Interfejs.h: Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:

Komponenty

- class Interfejs

  Modeluje Interfejs.

6.1.1 Opis szczegółowy

Definicja klasy Interfejs. Plik zawiera definicje klasy Interfejs wraz z deklaracjami metod wirtualnych

Definicja w pliku Interfejs.h.

6.2 Dokumentacja pliku main.cpp

Plik glowny programu.

```
#include "Wektor.h"
```

Wykres zależności załączania dla main.cpp:

Funkcje

- int main ()

6.2.1 Opis szczegółowy

Plik glowny programu.

Definicja w pliku main.cpp.

6.2.2 Dokumentacja funkcji

6.2.2.1 int main ( )

Definicja w linii 8 pliku main.cpp.

Oto graf wywołań dla tej funkcji:

6.3 Dokumentacja pliku strona-glowna.dox

6.4 Dokumentacja pliku Wektor.cpp

Definicje metod klasy Wektor.

```
#include "Wektor.h"
```

Wykres zależności załączania dla Wektor.cpp:

6.4.1 Opis szczegółowy

Definicje metod klasy Wektor.

Definicja w pliku Wektor.cpp.

6.5 Dokumentacja pliku Wektor.h

Definicja klasy Wektor.

```
#include "Interfejs.h"
```

Wykres zależności załączania dla Wektor.h: Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:

Komponenty

- class Wektor

  Modeluje pojecie Wektora.

Definicje

- #define ILOSC 5
- #define ILOSC_POW 10

6.5.1 Opis szczegółowy

Definicja klasy Wektor. Klasa modeluje pojecie wektora, bedacego pojemnikiem na dane

Definicja w pliku Wektor.h.

6.5.2 Dokumentacja definicji

6.5.2.1 #define ILOSC 5

Definicja w linii 3 pliku Wektor.h.

6.5.2.2 #define ILOSC_POW 10

Definicja w linii 4 pliku Wektor.h.

Skorowidz