Benchmark + Mnozenie

0.3

Wygenerowano przez Doxygen 1.8.6

Cz, 12 mar 2015 00:49:53

Contents

1 Indeks hierarchiczny **1**

1.1 Hierarchia klas . . . 1

2 Indeks klas **3**

2.1 Lista klas . . . 3

3 Indeks plików **5**

3.1 Lista plików . . . 5

4 Dokumentacja klas **7**

4.1 Dokumentacja klasy Benchmark . . . 7

4.1.1 Opis szczegółowy . . . 7

4.1.2 Dokumentacja funkcji składowych . . . 7

4.1.2.1 rozpocznij_pomiar . . . 7

4.1.2.2 testuj . . . 8

4.1.2.3 zakoncz_pomiar . . . 9

4.1.3 Dokumentacja atrybutów składowych . . . 9

4.1.3.1 czas_pomiaru . . . 9

4.1.3.2 t1 . . . 9

4.1.3.3 t2 . . . 9

4.2 Dokumentacja klasy Program . . . 10

4.2.1 Opis szczegółowy . . . 11

4.2.2 Dokumentacja konstruktora i destruktora . . . 11

4.2.2.1 Program . . . 11

4.2.2.2 ~Program . . . 11

4.2.3 Dokumentacja funkcji składowych . . . 11

4.2.3.1 getRozmiar_tab . . . 11

4.2.3.2 wczytaj_dane . . . 11

4.2.3.3 wczytaj_dane . . . 12

4.2.3.4 wykonaj_program . . . 13

4.2.3.5 wyswietl_dane . . . 13

4.2.3.6 zapisz_dane . . . 13

4.2.4 Dokumentacja atrybutów składowych . . . 13
4.2.4.1 plik_we . . . 13
4.2.4.2 plik_wy . . . 14
4.2.4.3 rozmiar_tab . . . 14
4.2.4.4 tab . . . 14
4.3 Dokumentacja klasy Tabx2 . . . 14
4.3.1 Opis szczegółowy . . . 15
4.3.2 Dokumentacja funkcji składowych . . . 15
4.3.2.1 wykonaj_program . . . 15

5 Dokumentacja plików **17**
5.1 Dokumentacja pliku benchmark.cpp . . . 17
5.2 benchmark.cpp . . . 17
5.3 Dokumentacja pliku benchmark.hh . . . 18
5.4 benchmark.hh . . . 19
5.5 Dokumentacja pliku main.cpp . . . 19
5.5.1 Dokumentacja funkcji . . . 20
5.5.1.1 main . . . 20
5.6 main.cpp . . . 21
5.7 Dokumentacja pliku program.cpp . . . 21
5.8 program.cpp . . . 21
5.9 Dokumentacja pliku program.hh . . . 22
5.10 program.hh . . . 23
5.11 Dokumentacja pliku tabx2.cpp . . . 24
5.12 tabx2.cpp . . . 24
5.13 Dokumentacja pliku tabx2.hh . . . 24
5.13.1 Opis szczegółowy . . . 25
5.14 tabx2.hh . . . 25

Indeks **27**

Chapter 1

Indeks hierarchiczny

1.1 Hierarchia klas

Ta lista dziedziczenia posortowana jest z grubsza, choć nie całkowicie, alfabetycznie:

Benchmark 7
Program 10
Tabx2 14

Chapter 2

Indeks klas

2.1 Lista klas

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

Benchmark
Klasa Benchmark 7
Program
Modeluje klase Program 10
Tabx2 14

Chapter 3

Indeks plików

3.1 Lista plików

Tutaj znajduje się lista wszystkich plików z ich krótkimi opisami:

benchmark.cpp
: Plik zawiera metody klasy Benchmark . . . 17

benchmark.hh
: Definicja klasy Benchmark . . . 19

main.cpp . . . 21

program.cpp
: Plik zawiera metody klasy Program . . . 21

program.hh
: Definicja klasy Program . . . 23

tabx2.cpp
: Plik zawiera metody klasy Tabx2 . . . 24

tabx2.hh
: Definicja klasy Tabx2 . . . 25

Chapter 4

Dokumentacja klas

4.1 Dokumentacja klasy Benchmark

Klasa Benchmark.

`#include <benchmark.hh>`

Metody publiczne

- void rozpocznij_pomiar ()

  Procedura rozpocznij_pomiar.
- void zakoncz_pomiar ()

  Procedura zakoncz_pomiar.
- double testuj (Program &program, char *dane, int ilosc_danych, int ilosc_testow)

  Metoda testuj.

Atrybuty prywatne

- timeval t1

  Zmienne t1, t2.
- timeval t2
- double czas_pomiaru

  Zmienna czas_pomiaru.

4.1.1 Opis szczegółowy

Jest to klasa służąca do testowania programów.

Definicja w linii 23 pliku benchmark.hh.

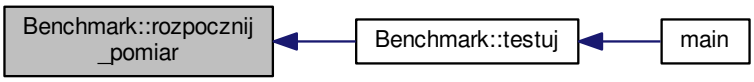
4.1.2 Dokumentacja funkcji składowych

4.1.2.1 void Benchmark::rozpocznij_pomiar ( )

Rozpoczyna pomiar czasu.

Definicja w linii 7 pliku benchmark.cpp.

Oto graf wywoływań tej funkcji:



4.1.2.2 double Benchmark::testuj (Program & *program*, char * *dane*, int *ilosc_danych*, int *ilosc_testow*)

Dokonuje testow wybranego programu.

Parametry

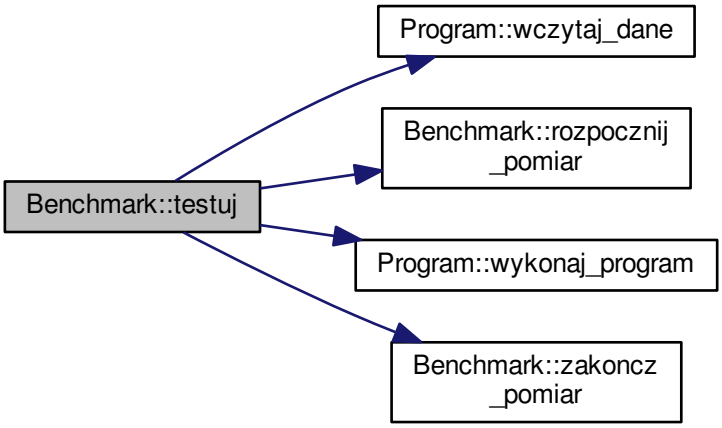
in	*program*	Program wybrany do testowania.
in	*dane*	Wskaznik na nazwe pliku z danymi.
in	*ilosc_danych*	Ilosc danych, ktore chcemy pobrac do testu.
in	*ilosc_testow*	Ilosc testow, jakie chcemy przeprowadzic.

Zwraca

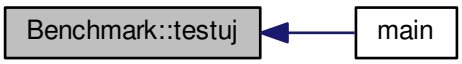
Metoda zwraca sredni czas wykonania programu dla podanych parametrow.

Definicja w linii 17 pliku benchmark.cpp.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:

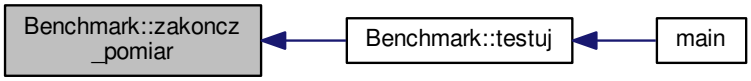


4.1.2.3 void Benchmark::zakoncz_pomiar ( )

Konczy pomiar czasu i zapisuje wartosc zmierzona w zmiennej czas_pomiaru.

Definicja w linii 11 pliku benchmark.cpp.

Oto graf wywoływań tej funkcji:



4.1.3 Dokumentacja atrybutów składowych

4.1.3.1 double Benchmark::czas_pomiaru [private]

Przechowuje obliczony czas pojedynczego pomiaru (w ms)

Definicja w linii 37 pliku benchmark.hh.

4.1.3.2 timeval Benchmark::t1 [private]

Zmienne przechowujace momenty rozpaczecia i zakonczenia pomiaru czasu.

Definicja w linii 30 pliku benchmark.hh.

4.1.3.3 timeval Benchmark::t2 [private]

Definicja w linii 30 pliku benchmark.hh.

Dokumentacja dla tej klasy została wygenerowana z plików:

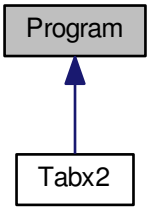
- benchmark.hh
- benchmark.cpp

4.2 Dokumentacja klasy Program

Modeluje klase Program.

`#include <program.hh>`

Diagram dziedziczenia dla Program



Metody publiczne

- int getRozmiar_tab ()

  Akcesor getRozmiar_tab.
- Program ()

  Konstruktor bezparametryczny.
- ∼Program ()

  Destruktor.
- bool wczytaj_dane (char ∗nazwa_pliku)

  Metoda wczytaj_dane.
- bool wczytaj_dane (char ∗nazwa_pliku, int ile_danych)

  Metoda wczytaj_dane.
- bool zapisz_dane (char ∗nazwa_pliku)
- void wyswietl_dane ()

  Procedura wyswietl_dane.
- virtual bool wykonaj_program ()

  Wirtualna metoda wykonaj_program.

Atrybuty chronione

- int rozmiar_tab

  Zmiena rozmiar_tab.
- int ∗ tab

  Zmienna tablica.
- ifstream plik_we

  Zmienna plik_we.
- ofstream plik_wy

  Zmienna plik_wy.

4.2.1 Opis szczegółowy

Klasa Program zawiera zmienne oraz metody wspolne dla wszystkich programow. Sa one zwiazane z przechowywaniem i obsluga danych.

Definicja w linii 22 pliku program.hh.

4.2.2 Dokumentacja konstruktora i destruktora

4.2.2.1 Program::Program ( ) [inline]

Przypisuje domyslna wartosc 0 dla rozmiaru tablicy danych oraz NULL dla wskaznika.

Definicja w linii 71 pliku program.hh.

4.2.2.2 Program::~Program ( ) [inline]

Usuwa dynamicznie utworzona tablice danych oraz przypisuje wskaznikowi wartosc NULL.

Definicja w linii 79 pliku program.hh.

4.2.3 Dokumentacja funkcji składowych

4.2.3.1 int Program::getRozmiar_tab ( ) [inline]

Metoda dajaca mozliwosc odczytu rozmiaru tablicy.

Definicja w linii 63 pliku program.hh.

4.2.3.2 bool Program::wczytaj_dane (char * *nazwa_pliku*)

Wczytuje dane z pliku. W pierwszej linii pliku musi znajdowac sie informacja o ilosci wczytywanych danych, dane w kolejnych liniach: ilosc_danych dana1 dana2 ...

Parametry

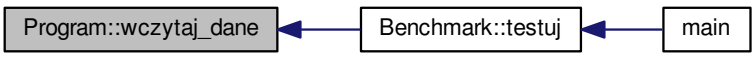
in	*nazwa_pliku*	Wskaznik do nazwy pliku do wczytania.

Zwracane wartości

TRUE	Poprawnie wczytano plik.
FALSE	Blad podczas wczytywania pliku.

Definicja w linii 8 pliku program.cpp.

Oto graf wywoływań tej funkcji:



4.2.3.3 bool Program::wczytaj_dane (char * *nazwa_pliku,* int *ile_danych*)

Wczytuje okreslona liczbe danych z pliku. W pierwszej linii pliku musi znajdowac sie informacja o ilosci wczytywanych danych, dane w kolejnych liniach: ilosc_danych dana1 dana2 ...

Parametry

in	*nazwa_pliku*	Wskaznik do nazwy pliku do wczytania.
in	*ile_danych*	Ilosc danych, jakie chcemy wczytac.

Zwracane wartości

TRUE	Poprawnie wczytano plik.
FALSE	Blad podczas wczytywania pliku.

Definicja w linii 26 pliku program.cpp.

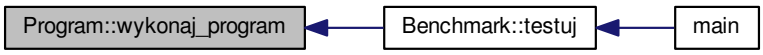
4.2.3.4 bool Program::wykonaj_program ( ) [virtual]

Wykonuje program na zadanej liczbie danych.

Reimplementowana w Tabx2.

Definicja w linii 67 pliku program.cpp.

Oto graf wywoływań tej funkcji:



4.2.3.5 void Program::wyswietl_dane ( )

Wypisuje wczytane dane jedna pod druga na standardowy strumien wyjscia.

Definicja w linii 62 pliku program.cpp.

4.2.3.6 bool Program::zapisz_dane (char * *nazwa_pliku*)

Metoda zapisz_dane

Zapisuje przetworzone dane do pliku. W pierwszej linijce zamieszcza informacje o ilosci danych, w kolejnych liniach pojedyncze dane: ilosc_danych dana1 dana2 ...

Parametry

in	*nazwa_pliku*	Wskaznik do nazwy pliku do zapisu.

Zwracane wartości

TRUE	Poprawnie zapisano plik.
FALSE	Blad podczas zapisu pliku.

Definicja w linii 47 pliku program.cpp.

4.2.4 Dokumentacja atrybutów składowych

4.2.4.1 ifstream Program::plik_we [protected]

Zmienna przechowujaca strumien wejsciowy do otwartego pliku z wczytywanymi danymi.

Definicja w linii 47 pliku program.hh.

4.2.4.2 ofstream Program::plik_wy [protected]

Zmienna przechowujaca strumien wyjsciowy do tworzonego pliku z danymi po przetworzeniu.

Definicja w linii 55 pliku program.hh.

4.2.4.3 int Program::rozmiar_tab [protected]

Zmienna przechowujaca informacje o ilosci wczytanych danych, ktora rowna jest dlugosci utworzonej tablicy dynamicznej (wskazywanej wskaznikiem tab).

Definicja w linii 31 pliku program.hh.

4.2.4.4 int* Program::tab [protected]

Zamienna wskaznikowa wskazujaca na dynamicznie tworzona tablice z danymi.

Definicja w linii 39 pliku program.hh.

Dokumentacja dla tej klasy została wygenerowana z plików:

- program.hh
- program.cpp

4.3 Dokumentacja klasy Tabx2

`#include <tabx2.hh>`

Diagram dziedziczenia dla Tabx2

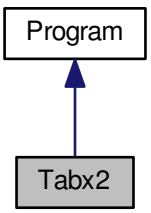
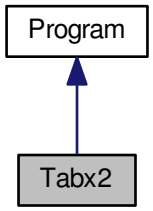


Diagram współpracy dla Tabx2:



Metody publiczne

- virtual bool wykonaj_program ()

  Metoda wirtualna wykonaj_program.

Dodatkowe Dziedziczone Składowe

4.3.1 Opis szczegółowy

Definicja w linii 18 pliku tabx2.hh.

4.3.2 Dokumentacja funkcji składowych

4.3.2.1 bool Tabx2::wykonaj_program ( ) [virtual]

Dokonuje przemnozenia przez 2 wszystkich danych znajdujacych sie w tablicy wskazywanej przez tab.

Zwracane wartości

TRUE	Poprawnie dokonano mnozenia wszystkich liczb
FALSE	Rozmiar tablicy danych wynosi 0

Reimplementowana z Program.

Definicja w linii 7 pliku tabx2.cpp.

Dokumentacja dla tej klasy została wygenerowana z plików:

- tabx2.hh
- tabx2.cpp

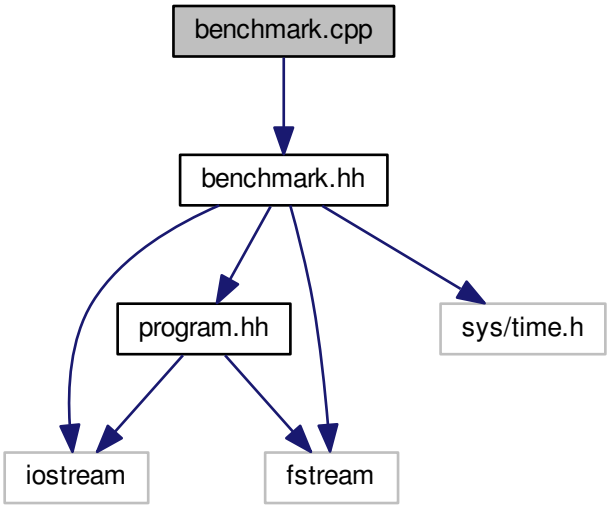
Chapter 5

Dokumentacja plików

5.1 Dokumentacja pliku benchmark.cpp

Plik zawiera metody klasy Benchmark.

`#include "benchmark.hh"`
Wykres zależności załączania dla benchmark.cpp:



5.2 benchmark.cpp

```
00001 #include "benchmark.hh"
00002
00007 void Benchmark::rozpocznij_pomiar(){
00008   gettimeofday(&t1, NULL);
00009 }
00010
00011 void Benchmark::zakoncz_pomiar(){
00012   gettimeofday(&t2, NULL);
00013   czas_pomiaru = (t2.tv_sec - t1.tv_sec) * 1000.0;      // sec to ms
00014   czas_pomiaru += (t2.tv_usec - t1.tv_usec) / 1000.0;   // us to ms
```

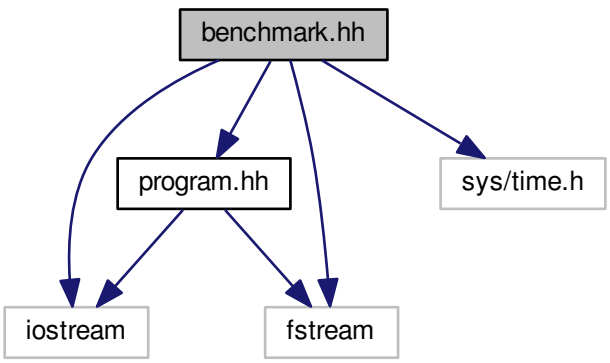
```
00015 }
00016
00017 double Benchmark::testuj(Program &program, char* dane, int ilosc_danych, int
      ilosc_testow){
00018   double suma=0;
00019   double srednia=0;
00020   ofstream wyniki;
00021   wyniki.open("wyniki.csv",ios::app);
00022
00023   if(program.wczytaj_dane(dane,ilosc_danych)==false){
00024       cerr<<"Niewystarczajaca ilosc danych!"<<endl;
00025       return 0;
00026     }
00027   //char* dane_wy = (char*)"dane_wy.dat"; //do zapisu do pliku
00028   rozpocznij_pomiar();
00029   program.wykonaj_program();
00030   //program.zapisz_dane(dane_wy);//zapisywanie wynikow do pliku
00031   zakoncz_pomiar();
00032   suma+=czas_pomiaru;
00033   for(int i=1;i<ilosc_testow;i++){
00034     //program.wczytaj_dane(dane,ilosc_danych); //zawsze dane od poczatku
00035     rozpocznij_pomiar();
00036     program.wykonaj_program();
00037     zakoncz_pomiar();
00038     suma+=czas_pomiaru;
00039   }
00040   srednia=suma/(ilosc_testow);
00041   wyniki<<endl<<ilosc_danych<<","<<srednia;
00042   wyniki.close();
00043   return srednia;
00044 }
```

5.3 Dokumentacja pliku benchmark.hh

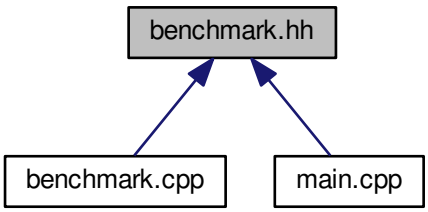
Definicja klasy Benchmark.

```
#include <iostream>
#include "program.hh"
#include <sys/time.h>
#include <fstream>
```

Wykres zależności załączania dla benchmark.hh:



Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



Komponenty

- class Benchmark

  Klasa Benchmark.

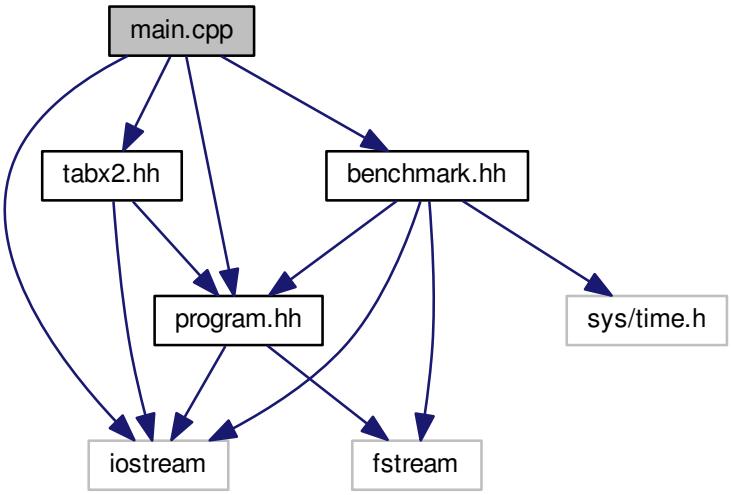
5.4 benchmark.hh

```
00001 //benchmark.hh
00002
00003 #ifndef BENCHMARK_HH
00004 #define BENCHMARK_HH
00005
00006 #include <iostream>
00007 #include "program.hh"
00008 #include <sys/time.h>
00009 #include <fstream>
00010
00016 using namespace std;
00017
00023 class Benchmark{
00024 private:
00030   timeval t1, t2;
00031
00037   double czas_pomiaru;
00038
00039 public:
00045   void rozpocznij_pomiar();
00046
00052   void zakoncz_pomiar();
00053
00066   double testuj(Program &program,char* dane, int ilosc_danych, int ilosc_testow);
00067 };
00068
00069 #endif
```

5.5 Dokumentacja pliku main.cpp

```
#include <iostream>
#include "program.hh"
#include "tabx2.hh"
#include "benchmark.hh"
```

Wykres zależności załączania dla main.cpp:



Funkcje

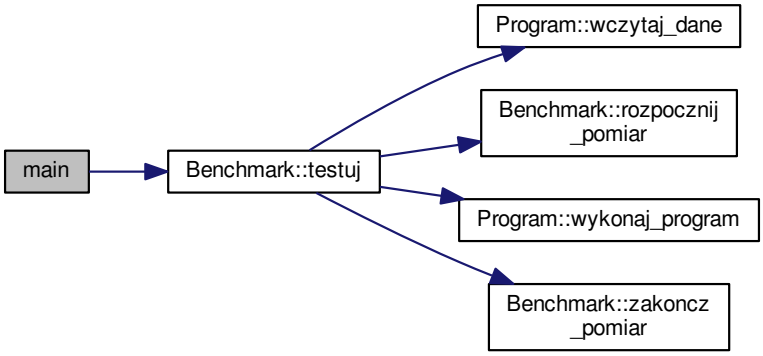
- int main ()

5.5.1 Dokumentacja funkcji

5.5.1.1 int main ( )

Definicja w linii 9 pliku main.cpp.

Oto graf wywołań dla tej funkcji:



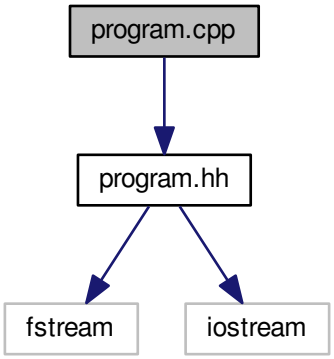
5.6 main.cpp

```
00001 //main.cpp
00002 #include <iostream>
00003 #include "program.hh"
00004 #include "tabx2.hh"
00005 #include "benchmark.hh"
00006
00007 using namespace std;
00008
00009 int main(){
00010   Tabx2 a;
00011   Benchmark b;
00012   char* dane = (char*)"dane.dat";
00013   int ilosc_testow = 20;
00014   for(int ilosc_danych=50; ilosc_danych<1000000;ilosc_danych*=1.5){
00015     cout << b.testuj(a,dane,ilosc_danych,ilosc_testow) << endl;
00016   }
00017   return 0;
00018 }
```

5.7 Dokumentacja pliku program.cpp

Plik zawiera metody klasy Program.

`#include "program.hh"`
Wykres zależności załączania dla program.cpp:



5.8 program.cpp

```
00001 //program.cpp
00002 #include "program.hh"
00003
00004 using namespace std;
00008 bool Program::wczytaj_dane(char* nazwa_pliku){
00009   if(plik_we.good()==true)
00010     plik_we.close();
00011   plik_we.open(nazwa_pliku);
00012   if(plik_we.good()==false){
00013     cerr<<"Blad odczytu pliku!";
00014     return false;
00015   }
00016   plik_we >> rozmiar_tab;
00017   tab=new int [rozmiar_tab];
00018   int i=0;
00019   while(plik_we >> tab[i]){
00020     i++;
```

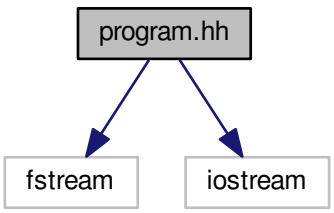
```
00021   }
00022   plik_we.close();
00023   return true;
00024 }
00025
00026 bool Program::wczytaj_dane(char* nazwa_pliku, int ile_danych){
00027   if(plik_we.good()==true)
00028     plik_we.close();
00029   plik_we.open(nazwa_pliku);
00030   if(plik_we.good()==false){
00031     cerr<<"Blad odczytu pliku!"<<endl;
00032     return false;
00033   }
00034   plik_we >> rozmiar_tab;
00035   if(ile_danych>rozmiar_tab){
00036     plik_we.close();
00037     return false;
00038   }
00039   rozmiar_tab=ile_danych;
00040   tab=new int [ile_danych];
00041   for(int i=0;i<ile_danych;i++)
00042     plik_we>>tab[i];
00043   plik_we.close();
00044   return true;
00045 }
00046
00047 bool Program::zapisz_dane(char* nazwa_pliku){
00048   if(plik_wy.good()==true)
00049     plik_wy.close();
00050   plik_wy.open(nazwa_pliku);
00051   if(plik_wy.good()==false){
00052     cerr<<"Blad odczytu pliku!";
00053     return false;
00054   }
00055   plik_wy << rozmiar_tab << endl;
00056   for(int i=0;i<rozmiar_tab;i++)
00057     plik_wy << tab[i] << endl;
00058   plik_wy.close();
00059   return true;
00060 }
00061
00062 void Program::wyswietl_dane(){
00063   for(int i=0;i<rozmiar_tab;i++)
00064     cout<<tab[i]<<endl;
00065 }
00066
00067 bool Program::wykonaj_program(){
00068   cerr<<"Nie wybrano programu do wykonania!"<<endl;
00069   return false;
00070 }
```

5.9 Dokumentacja pliku program.hh

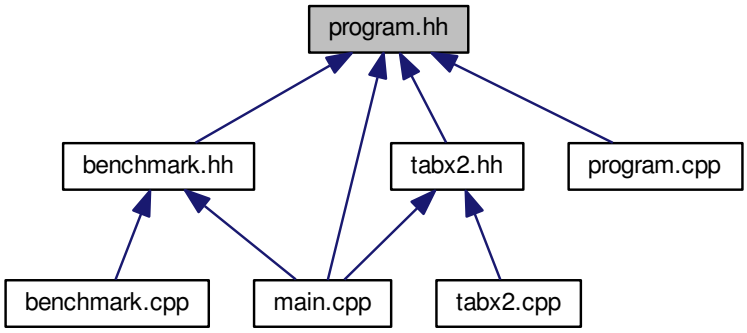
Definicja klasy Program.

```
#include <fstream>
#include <iostream>
```

Wykres zależności załączania dla program.hh:



Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



Komponenty

- class Program

  Modeluje klase Program.

5.10 program.hh

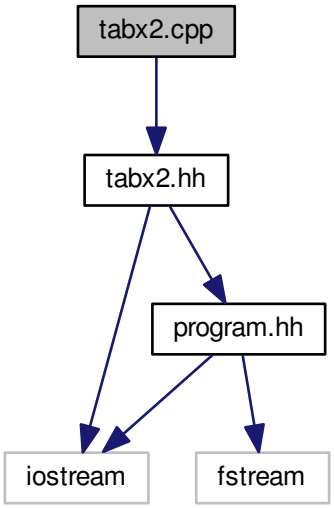
```
00001 //program.hh
00002
00003 #ifndef PROGRAM_HH
00004 #define PROGRAM_HH
00005
00006 #include <fstream>
00007 #include <iostream>
00008
00009 using namespace std;
00010
00022 class Program{
00023 protected:
00031   int rozmiar_tab;
00032
00039   int *tab;
00040
00047   ifstream plik_we;
00048
00055   ofstream plik_wy;
00056
00057 public:
00063   int getRozmiar_tab(){return rozmiar_tab;}
00064
00071   Program(){rozmiar_tab=0;tab=NULL;}
00072
00079   ~Program(){delete[] tab; tab=NULL;}
00080
00096   bool wczytaj_dane(char* nazwa_pliku);
00097
00114   bool wczytaj_dane(char* nazwa_pliku, int ile_danych);
00115
00131  bool zapisz_dane(char* nazwa_pliku);
00132
00138   void wyswietl_dane();
00139
00145   virtual bool wykonaj_program();
00146 };
00147
00148 #endif
```

5.11 Dokumentacja pliku tabx2.cpp

Plik zawiera metody klasy Tabx2.

```
#include "tabx2.hh"
```

Wykres zależności załączania dla tabx2.cpp:



5.12 tabx2.cpp

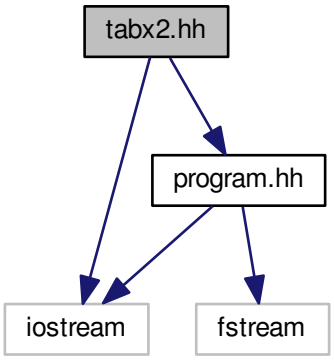
```
00001 #include "tabx2.hh"
00002
00003 using namespace std;
00007 bool Tabx2::wykonaj_program(){
00008   if(rozmiar_tab==0){
00009     cerr<<"Brak danych wejsciowych!"<<endl;
00010     return false;
00011   }
00012   for(int i=0;i<rozmiar_tab;i++){
00013     tab[i]*=2;
00014   }
00015   return true;
00016 }
```

5.13 Dokumentacja pliku tabx2.hh

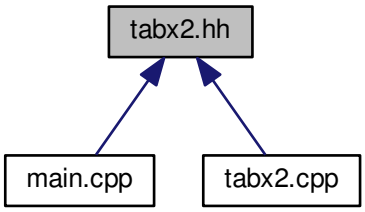
Definicja klasy Tabx2.

```
#include <iostream>
#include "program.hh"
```

Wykres zależności załączania dla tabx2.hh:



Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



Komponenty

- class Tabx2

5.13.1 Opis szczegółowy

Klasa Tabx2 jest klasa pochodna od klasy Program. Definiuje metode podawajajaca kazda liczbe znajdujaca sie w tablicy danych wskazywanej przez zmienna tab klasy Program.

Definicja w pliku tabx2.hh.

5.14 tabx2.hh

```
00001 //mnozenie_tablicy.hh
00002
00003 #ifndef TABX2_HH
00004 #define TABX2_HH
00005
```

```
00015 #include <iostream>
00016 #include "program.hh"
00017
00018 class Tabx2: public Program{
00019
00020 public:
00030   virtual bool wykonaj_program();
00031 };
00032
00033 #endif
```

Index

~Program
    Program, 11

Benchmark, 7
    czas_pomiaru, 9
    rozpocznij_pomiar, 7
    t1, 9
    t2, 9
    testuj, 8
    zakoncz_pomiar, 9
benchmark.cpp, 17
benchmark.hh, 18, 19

czas_pomiaru
    Benchmark, 9

getRozmiar_tab
    Program, 11

main
    main.cpp, 20
main.cpp, 19, 21
    main, 20

plik_we
    Program, 13
plik_wy
    Program, 14
Program, 10
    ~Program, 11
    getRozmiar_tab, 11
    plik_we, 13
    plik_wy, 14
    Program, 11
    rozmiar_tab, 14
    tab, 14
    wczytaj_dane, 11
    wykonaj_program, 13
    wyswietl_dane, 13
    zapisz_dane, 13
program.cpp, 21
program.hh, 22, 23

rozmiar_tab
    Program, 14
rozpocznij_pomiar
    Benchmark, 7

t1
    Benchmark, 9
t2
    Benchmark, 9
tab
    Program, 14
Tabx2, 14
    wykonaj_program, 15
tabx2.cpp, 24
tabx2.hh, 24, 25
testuj
    Benchmark, 8

wczytaj_dane
    Program, 11
wykonaj_program
    Program, 13
    Tabx2, 15
wyswietl_dane
    Program, 13

zakoncz_pomiar
    Benchmark, 9
zapisz_dane
    Program, 13