Drzewo czerwono-czarne

Wygenerowano przez Doxygen 1.8.6

N, 24 kwi 2016 21:39:54

Spis treści

1 Indeks hierarchiczny

1.1 Hierarchia klas

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

IBinaryTree< Object > ??

RBTree< Object > ??

RBTree_Test< Object > ??

IRunnable< Object > ??

RBTree_Test< Object > ??

Stopwatch ??

AdvancedStopwatch ??

TreeNode< Object > ??

2 Indeks klas

2.1 Lista klas

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

AdvancedStopwatch
Klasa implementująca rozbudowany stoper ??

IBinaryTree< Object > ??

IRunnable< Object >
Klasa szablonowa modelująca interfejs "Biegacza" ??

RBTree< Object >
Szablonowa klasa implementująca drzewo binarne czerwono-czarne ??

RBTree_Test< Object >
Szablonowa klasa testowego drzewa czerwono-czarnego ??

Stopwatch
Klasa implementująca podstawowy stoper ??

TreeNode< Object > ??

3 Indeks plików

3.1 Lista plików

Tutaj znajduje się lista wszystkich plików z ich krótkimi opisami:

prj/inc/AdvancedStopwatch.hh ??

prj/inc/IBinaryTree.hh ??

prj/inc/IRunnable.hh ??

prj/inc/RBTree.hh ??

prj/inc/RBTree_Test.hh ??

prj/inc/Stopwatch.hh ??

prj/inc/TreeNode.hh ??

prj/src/AdvancedStopwatch.cpp ??

prj/src/BTree.cpp ??

prj/src/main.cpp ??

prj/src/RBTree.cpp ??

prj/src/RBTree_Test.cpp ??

prj/src/Stopwatch.cpp ??

4 Dokumentacja klas

4.1 Dokumentacja klasy AdvancedStopwatch

Klasa implementująca rozbudowany stoper.

`#include <AdvancedStopwatch.hh>`

Diagram dziedziczenia dla AdvancedStopwatch

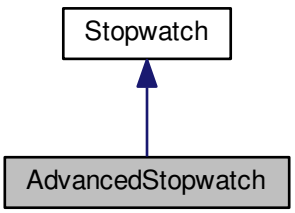
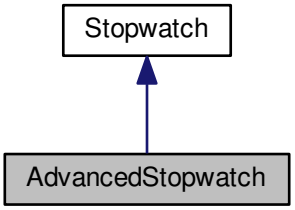


Diagram współpracy dla AdvancedStopwatch:



Metody publiczne

- AdvancedStopwatch ()
- ~AdvancedStopwatch ()
- unsigned & Rozmiar ()
- bool SaveElapsedTime (double rekord)

  Metoda zapisująca wartość pomiaru czasu okrążenia.
- double SeriesAverage ()

  Metoda wyliczająca średni czas okrążenia.
- bool SaveAverageTimeToBuffer (double rekord)

  Metoda zapisująca średni czas okrążenia do bufora plikowego.
- void PrintElapsedTimes ()

  Metoda wypisująca zawartość pamięci stopera.
- void CleanElapsedTimes ()

  Metoda usuwająca zawartość pamięci stopera.
- void CleanFileBuffer ()

  Metoda usuwająca zawartość bufora plikowego stopera.
- bool DumpFileBuffer (string nazwaPliku)

  Metoda zapisująca zawartość bufora plikowego do pliku.
- bool DumpToFile (string nazwaPliku, double rekord)

  Metoda zapisująca pojedynczy rekord bufora plikowego do pliku.

Dodatkowe Dziedziczone Składowe

4.1.1 Opis szczegółowy

Klasa implementująca rozbudowany stoper.

Klasa jest modelelem stopera z funkcją zapisu czasu okrążeń, liczeniem średniego czasu kilku okrążeń, zapisu zmierzonych czasów do pliku.

Definicja w linii 28 pliku AdvancedStopwatch.hh.

4.1.2 Dokumentacja konstruktora i destruktora

4.1.2.1 AdvancedStopwatch::AdvancedStopwatch ( )

Definicja w linii 4 pliku AdvancedStopwatch.cpp.

4.1.2.2 AdvancedStopwatch::∼AdvancedStopwatch ( )

Definicja w linii 15 pliku AdvancedStopwatch.cpp.

4.1.3 Dokumentacja funkcji składowych

4.1.3.1 void AdvancedStopwatch::CleanElapsedTimes ( )

Metoda usuwająca zawartość pamięci stopera.

Definicja w linii 66 pliku AdvancedStopwatch.cpp.

4.1.3.2 void AdvancedStopwatch::CleanFileBuffer ( )

Metoda usuwająca zawartość bufora plikowego stopera.

Definicja w linii 74 pliku AdvancedStopwatch.cpp.

4.1.3.3 bool AdvancedStopwatch::DumpFileBuffer (string *nazwaPliku*)

Metoda zapisująca zawartość bufora plikowego do pliku.

Dokonuje zapisu rekordów w buforze do pliku.

Parametry

>p0.10	>p0.15	p0.678
in *nazwaPliku* - nazwa pliku, do którego mają zostać zapisane czasy

Zwracane wartości

>p0.25	p0.705
true - jeśli udało się zapisać
false - jeśli udało się zapisać

Definicja w linii 80 pliku AdvancedStopwatch.cpp.

4.1.3.4 bool AdvancedStopwatch::DumpToFile (string *nazwaPliku,* double *rekord*)

Metoda zapisująca pojedynczy rekord bufora plikowego do pliku.

Dokonuje zapisu wybranego rekordu w buforze do pliku.

Parametry

>p0.10	>p0.15	p0.678
in *nazwaPliku* - nazwa pliku, do którego ma zostać zapisany czas
in *rekord* - wartość pomiaru czasu, która ma być zapisana

Zwracane wartości

>p0.25	p0.705
true - jeśli udało się zapisać
false - jeśli udało się zapisać

Definicja w linii 98 pliku AdvancedStopwatch.cpp.

4.1.3.5 void AdvancedStopwatch::PrintElapsedTimes ( )

Metoda wypisująca zawartość pamięci stopera.

Definicja w linii 58 pliku AdvancedStopwatch.cpp.

4.1.3.6 unsigned& AdvancedStopwatch::Rozmiar ( ) [inline]

Definicja w linii 36 pliku AdvancedStopwatch.hh.

4.1.3.7 bool AdvancedStopwatch::SaveAverageTimeToBuffer (double *rekord*)

Metoda zapisująca średni czas okrążenia do bufora plikowego.

Dodaje podany czas do pamięci stopera, z której można dokonać zapisu do pliku.

Parametry

>p0.10	>p0.15	p0.678
`in` *rekord* - wartość pomiaru czasu

Zwracane wartości

>p0.25	p0.705
true - jeśli udało się zapisać
false - jeśli udało się zapisać

Definicja w linii 49 pliku AdvancedStopwatch.cpp.

4.1.3.8 bool AdvancedStopwatch::SaveElapsedTime (double *rekord*)

Metoda zapisująca wartość pomiaru czasu okrążenia.

Dodaje podany czas do tablicy czasów okrążeń.

Parametry

>p0.10	>p0.15	p0.678
`in` *rekord* - wartość pomiaru czasu

Zwracane wartości

>p0.25	p0.705
true - jeśli udało się zapisać
false - jeśli udało się zapisać

Definicja w linii 26 pliku AdvancedStopwatch.cpp.

4.1.3.9 double AdvancedStopwatch::SeriesAverage ( )

Metoda wyliczająca średni czas okrążenia.

Definicja w linii 37 pliku AdvancedStopwatch.cpp.

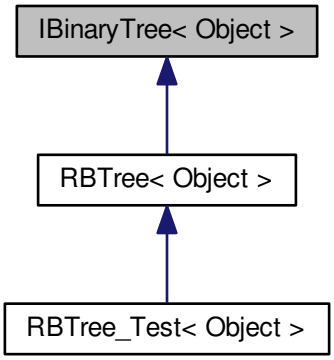
Dokumentacja dla tej klasy została wygenerowana z plików:

- prj/inc/AdvancedStopwatch.hh
- prj/src/AdvancedStopwatch.cpp

4.2 Dokumentacja szablonu klasy IBinaryTree< Object >

`#include <IBinaryTree.hh>`

Diagram dziedziczenia dla IBinaryTree< Object >



Metody publiczne

- virtual void Insert (const Object newItem)=0
- virtual void Remove (Object item)=0
- virtual bool IsEmpty ()=0

4.2.1 Opis szczegółowy

template<typename Object>class IBinaryTree< Object >

Definicja w linii 13 pliku IBinaryTree.hh.

4.2.2 Dokumentacja funkcji składowych

4.2.2.1 template<typename Object > virtual void IBinaryTree< Object >::Insert (const Object *newItem*) [pure virtual]

Implementowany w RBTree< Object >.

4.2.2.2 template<typename Object > virtual bool IBinaryTree< Object >::IsEmpty ( ) [pure virtual]

Implementowany w RBTree< Object >.

4.2.2.3 template<typename Object > virtual void IBinaryTree< Object >::Remove (Object *item*) [pure virtual]

Implementowany w RBTree< Object >.

Dokumentacja dla tej klasy została wygenerowana z pliku:

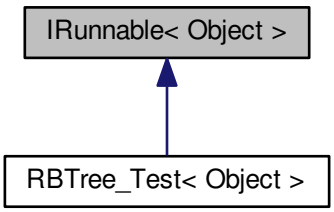
- prj/inc/IBinaryTree.hh

4.3 Dokumentacja szablonu klasy IRunnable< Object >

Klasa szablonowa modelująca interfejs "Biegacza".

`#include <IRunnable.hh>`

Diagram dziedziczenia dla IRunnable< Object >



Metody publiczne

- virtual bool Prepare (Object parametr)=0

  Metoda przygotowująca obiekt do operacji.
- virtual bool Run ()=0

  Metoda uruchamiająca zdefiniowaną operację

4.3.1 Opis szczegółowy

template<typename Object>class IRunnable< Object >

Klasa szablonowa modelująca interfejs "Biegacza".

Klasa jest abstrakcyjnym uogólnieniem obiektu, na którym można wykonać zdefiniowane operacje, którym z kolei można zmierzyć czas wykonywania.

Definicja w linii 22 pliku IRunnable.hh.

4.3.2 Dokumentacja funkcji składowych

4.3.2.1 template<typename Object > virtual bool IRunnable< Object >::Prepare (Object *parametr*) [pure virtual]

Metoda przygotowująca obiekt do operacji.

Parametry

>p0.10	>p0.15	p0.678
`in` *rozmiar* - liczba elementów do przygotowania;

Zwracane wartości

>p0.25	p0.705
true - jeśli przygotowanie się powiodło

false - jeśli wystąpił jakiś błąd

Implementowany w RBTree_Test< Object >.

4.3.2.2 template<typename Object > virtual bool IRunnable< Object >::Run ( ) [pure virtual]

Metoda uruchamiająca zdefiniowaną operację

Parametry

>p0.10	>p0.15	p0.678
in	*track*	- parametr wykonania operacji

Zwracane wartości

>p0.25	p0.705
true	- jeśli operacja zakończyła się pomyślnie
false	- jeśli wystąpił jakiś błąd

Implementowany w RBTree_Test< Object >.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- prj/inc/IRunnable.hh

4.4 Dokumentacja szablonu klasy RBTree< Object >

Szablonowa klasa implementująca drzewo binarne czerwono-czarne.

`#include <RBTree.hh>`

Diagram dziedziczenia dla RBTree< Object >

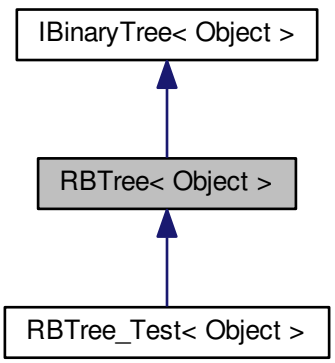
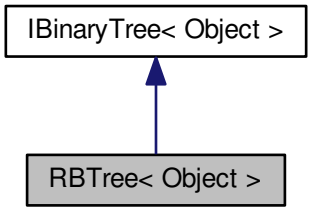


Diagram współpracy dla RBTree< Object >:



Metody publiczne

- RBTree ()

  Konstruktor listy jednokierunkowaj.
- ~RBTree ()

  Destruktor listy jednokierunkowaj.
- virtual bool IsEmpty ()

  Metoda sprawdzająca, czy drzewo jest puste.
- void Clear (TreeNode< Object > *v)

  Metoda usuwająca rekurencyjnie podany węzeł
- void PrintAll (TreeNode< Object > *v)

  Metoda drukująca drzewo na standardowe wyjście.
- TreeNode< Object > & SGuard ()

  Metoda zwracająca referencję do wartownika.
- TreeNode< Object > *& Root ()

  Metoda zwracająca korzeń
- TreeNode< Object > * Find (Object item)
- TreeNode< Object > * Max (TreeNode< Object > *v)

  Metoda znajdująca największy następnik podanego elementu.
- TreeNode< Object > * Min (TreeNode< Object > *v)

  Metoda znajdująca najmniejszy następnik podanego elementu.
- TreeNode< Object > * Successor (TreeNode< Object > *v)

  Metoda znajdująca następnik podanego węzła.
- void RotateRight (TreeNode< Object > *v)

  Wykonuje rotację w prawo.
- void RotateLeft (TreeNode< Object > *v)

  Wykonuje rotację w lewo.
- virtual void Insert (const Object newItem)

  Dodaje nowy element do drzewa.
- virtual void Remove (Object item)

  Usuwa podany element z drzewa.

Atrybuty chronione

- TreeNode< Object > S
- TreeNode< Object > * root

4.4.1 Opis szczegółowy

template<typename Object>class RBTree< Object >

Szablonowa klasa implementująca drzewo binarne czerwono-czarne.

Drzewo jest zbudowana w oparciu o węzły TreeNode. Może przechowywać dowolny typ danych dzięki zastosowaniu szablonu.

Definicja w linii 24 pliku RBTree.hh.

4.4.2 Dokumentacja konstruktora i destruktora

4.4.2.1 template<typename Object > RBTree< Object >::RBTree ( )

Konstruktor listy jednokierunkowaj.

Ustawia kolor wartownika na czarny i przypisuje wartownika do roota.

Definicja w linii 142 pliku RBTree.hh.

4.4.2.2 template<typename Object > RBTree< Object >::~RBTree ( )

Destruktor listy jednokierunkowaj.

Usuwa drzewo poprzez rekurencyjne wywołanie metody Clear().

Definicja w linii 150 pliku RBTree.hh.

4.4.3 Dokumentacja funkcji składowych

4.4.3.1 template<typename Object > void RBTree< Object >::Clear (TreeNode< Object > * *v*)

Metoda usuwająca rekurencyjnie podany węzeł

Usuwa synów (jeśli istnieją) oraz sam podany węzeł.

Parametry

>p0.10	>p0.15	p0.678
`in` *v*	- węzeł do usunięcia	

Definicja w linii 162 pliku RBTree.hh.

4.4.3.2 template<typename Object > TreeNode< Object > * RBTree< Object >::Find (Object *item*)

Definicja w linii 197 pliku RBTree.hh.

4.4.3.3 template<typename Object > void RBTree< Object >::Insert (const Object *newItem*) [virtual]

Dodaje nowy element do drzewa.

Tworzy nowy węzeł o podanej wartości, szuka miejsca do dodania. W razie potrzeby wykonuje balansowanie drzewa metodologią czerwono-czarną.

Parametry

>p0.10	>p0.15	p0.678
`in` *newItem*	- wartość nowego węzła do dodania	

Implementuje IBinaryTree< Object >.

Definicja w linii 307 pliku RBTree.hh.

4.4.3.4 template<typename Object > bool RBTree< Object >::IsEmpty ( ) [virtual]

Metoda sprawdzająca, czy drzewo jest puste.

Sprawdza, czy root wskazuje na NULL.

Zwracane wartości

>p0.25	p0.705
true - jeśli jest puste
false - jeśli nie jest puste;

Implementuje IBinaryTree< Object >.

Definicja w linii 156 pliku RBTree.hh.

4.4.3.5 template<typename Object > TreeNode< Object > * RBTree< Object >::Max (TreeNode< Object > * *v*)

Metoda znajdująca największy następnik podanego elementu.

Parametry

>p0.10	>p0.15	p0.678
in *v* - węzeł, od którego ma zacząć

Zwraca

Wskaźnik do elementu największego.

Definicja w linii 213 pliku RBTree.hh.

4.4.3.6 template<typename Object > TreeNode< Object > * RBTree< Object >::Min (TreeNode< Object > * *v*)

Metoda znajdująca najmniejszy następnik podanego elementu.

Parametry

>p0.10	>p0.15	p0.678
in *v* - węzeł, od którego ma zacząć

Zwraca

Wskaźnik do elementu najmniejszego.

Definicja w linii 222 pliku RBTree.hh.

4.4.3.7 template<typename Object > void RBTree< Object >::PrintAll (TreeNode< Object > * *v*)

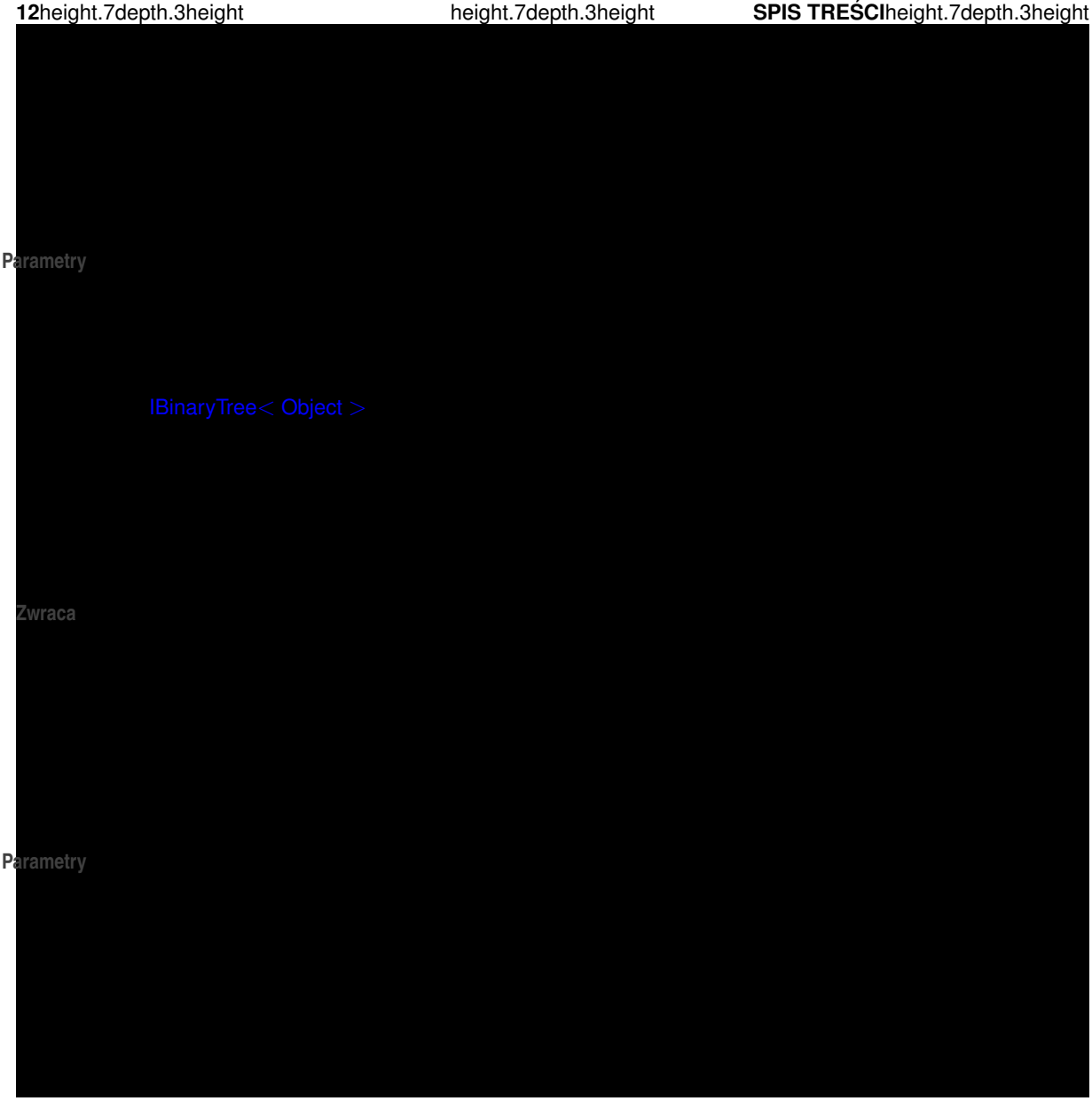
Metoda drukująca drzewo na standardowe wyjście.

Dane węzłów wyświetlane są w porządku in order.

Parametry

>p0.10	>p0.15	p0.678
in *v* - węzeł, od którego ma zacząć

Definicja w linii 173 pliku RBTree.hh.



Parametry

IBinaryTree< Object >

Zwraca

Parametry

Wykonuje rotację w prawo.

Parametry

|>p0.10|>p0.15|p0.678|

`in` ***v*** - węzeł, od którego ma zacząć

Definicja w linii 251 pliku RBTree.hh.

4.4.3.12 template<typename Object > TreeNode< Object > & RBTree< Object >::SGuard ( )

Metoda zwracająca referencję do wartownika.

Zwraca

Referencja do wartownika.

Definicja w linii 185 pliku RBTree.hh.

4.4.3.13 template<typename Object > TreeNode< Object > * RBTree< Object >::Successor (TreeNode< Object > * *v*)

Metoda znajdująca następnik podanego węzła.

Parametry

>p0.10	>p0.15	p0.678
`in` *v* - węzeł, od którego ma zacząć

Zwraca

Wskaźnik do następnika.

Definicja w linii 231 pliku RBTree.hh.

4.4.4 Dokumentacja atrybutów składowych

4.4.4.1 template<typename Object > TreeNode<Object>* RBTree< Object >::root [protected]

wskaźnik na korzeń

Definicja w linii 28 pliku RBTree.hh.

4.4.4.2 template<typename Object > TreeNode<Object> RBTree< Object >::S [protected]

wartownik

Definicja w linii 27 pliku RBTree.hh.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- prj/inc/RBTree.hh

4.5 Dokumentacja szablonu klasy RBTree_Test< Object >

Szablonowa klasa testowego drzewa czerwono-czarnego.

`#include <RBTree_Test.hh>`

Diagram dziedziczenia dla RBTree_Test< Object >

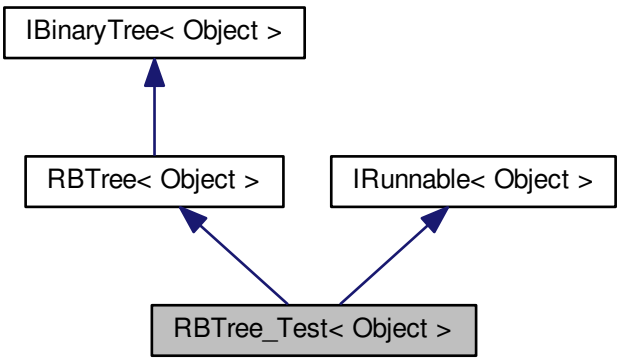
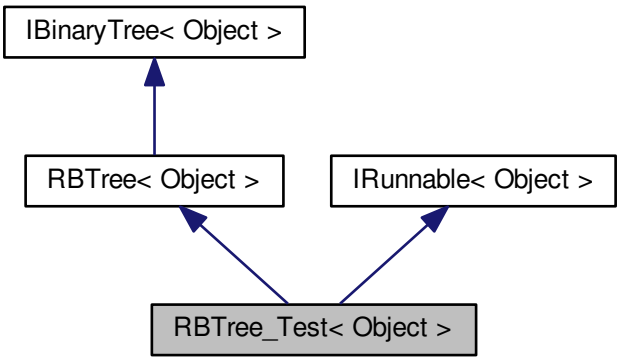


Diagram współpracy dla RBTree_Test< Object >:



Metody publiczne

- virtual bool Prepare (Object parametr)

  Metoda przygotowująca obiekt do operacji.
- virtual bool Run ()

  Metoda uruchamiająca proces odczytu.

Dodatkowe Dziedziczone Składowe

4.5.1 Opis szczegółowy

template<typename Object>class RBTree_Test< Object >

Szablonowa klasa testowego drzewa czerwono-czarnego.

Definicja w linii 20 pliku RBTree_Test.hh.

4.5.2 Dokumentacja funkcji składowych

4.5.2.1 template<typename Object > bool RBTree_Test< Object >::Prepare (Object *parametr*) [virtual]

Metoda przygotowująca obiekt do operacji.

Metoda wczytuje podaną liczbę losowych elementów do drzewa. Jeśli drzewo jest niepuste, wszystkie elementy są usuwane.

Parametry

>p0.10	>p0.15	p0.678
in *parametr* - liczba elementów do przygotowania;		

Zwracane wartości

>p0.25	p0.705

true - jeśli przygotowanie się powiodło

false - jeśli wystąpił jakiś błąd

Implementuje IRunnable< Object >.

Definicja w linii 50 pliku RBTree_Test.hh.

4.5.2.2 template<typename Object > bool RBTree_Test< Object >::Run ( ) [virtual]

Metoda uruchamiająca proces odczytu.

Wywołuje metodę Find() dla drzewa dla losowo generowanego elementu.

Zwracane wartości

|>p0.25|p0.705|

true - jeśli przygotowanie się powiodło

false - jeśli wystąpił jakiś błąd

Implementuje IRunnable< Object >.

Definicja w linii 67 pliku RBTree_Test.hh.

Dokumentacja dla tej klasy została wygenerowana z pliku:

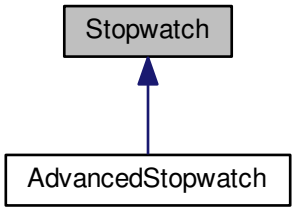
- prj/inc/RBTree_Test.hh

4.6 Dokumentacja klasy Stopwatch

Klasa implementująca podstawowy stoper.

`#include <Stopwatch.hh>`

Diagram dziedziczenia dla Stopwatch



Metody publiczne

- virtual void Start ()
  Rozpoczyna pomiar czasu.
- virtual void Stop ()
  Kończy pomiar czasu.
- virtual double GetElapsedTime ()
  Oblicza czas na podstawie pól klasy.

Atrybuty chronione

- timeval start
- timeval stop

4.6.1 Opis szczegółowy

Klasa implementująca podstawowy stoper.

Klasa jest modelelem stopera z funkcjami start, stop i oblicz czas.

Definicja w linii 20 pliku Stopwatch.hh.

4.6.2 Dokumentacja funkcji składowych

4.6.2.1 double Stopwatch::GetElapsedTime ( ) [virtual]

Oblicza czas na podstawie pól klasy.

Odejmuje wartości zapisane w polach stop i start. Daje wynik w mikrosekundach.

Definicja w linii 12 pliku Stopwatch.cpp.

4.6.2.2 void Stopwatch::Start ( ) [virtual]

Rozpoczyna pomiar czasu.

Przypisuje wynik metody gettimeofday() do pola start.

Definicja w linii 4 pliku Stopwatch.cpp.

4.6.2.3 void Stopwatch::Stop ( ) [virtual]

Kończy pomiar czasu.

Przypisuje wynik metody gettimeofday() do pola stop.

Definicja w linii 8 pliku Stopwatch.cpp.

4.6.3 Dokumentacja atrybutów składowych

4.6.3.1 timeval Stopwatch::start [protected]

Definicja w linii 23 pliku Stopwatch.hh.

4.6.3.2 timeval Stopwatch::stop [protected]

struktury timeval start i stop

Definicja w linii 23 pliku Stopwatch.hh.

Dokumentacja dla tej klasy została wygenerowana z plików:

- prj/inc/Stopwatch.hh
- prj/src/Stopwatch.cpp

4.7 Dokumentacja szablonu klasy TreeNode< Object >

```
#include <TreeNode.hh>
```

Metody publiczne

- TreeNode ()
- ~TreeNode ()
- TreeNode (Object newItem)
- Object & Element ()
- char & Color ()
- TreeNode< Object > *& Parent ()
- TreeNode< Object > *& Left ()
- TreeNode< Object > *& Right ()

4.7.1 Opis szczegółowy

template<typename Object>class TreeNode< Object >

Definicja w linii 8 pliku TreeNode.hh.

4.7.2 Dokumentacja konstruktora i destruktora

4.7.2.1 template<typename Object> TreeNode< Object >::TreeNode ( ) [inline]

Definicja w linii 18 pliku TreeNode.hh.

4.7.2.2 template<typename Object> TreeNode< Object >::~TreeNode ( ) [inline]

Definicja w linii 19 pliku TreeNode.hh.

4.7.2.3 template<typename Object> TreeNode< Object >::TreeNode (Object *newItem*) [inline]

Definicja w linii 20 pliku TreeNode.hh.

4.7.3 Dokumentacja funkcji składowych

4.7.3.1 template<typename Object> char& TreeNode< Object >::Color ( ) [inline]

Definicja w linii 23 pliku TreeNode.hh.

4.7.3.2 template<typename Object> Object& TreeNode< Object >::Element ( ) [inline]

Definicja w linii 21 pliku TreeNode.hh.

4.7.3.3 template<typename Object> TreeNode<Object>* & TreeNode< Object >::Left ( ) [inline]

Definicja w linii 25 pliku TreeNode.hh.

4.7.3.4 template<typename Object> TreeNode<Object>* & TreeNode< Object >::Parent ( ) [inline]

Definicja w linii 24 pliku TreeNode.hh.

4.7.3.5 template<typename Object> TreeNode<Object>* & TreeNode< Object >::Right ( ) [inline]

Definicja w linii 26 pliku TreeNode.hh.

Dokumentacja dla tej klasy została wygenerowana z pliku:

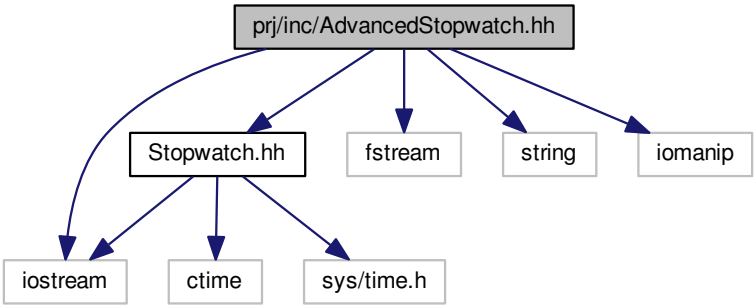
- prj/inc/TreeNode.hh

5 Dokumentacja plików

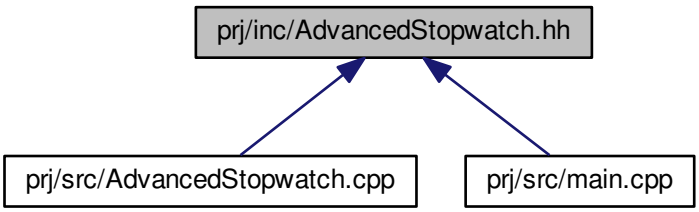
5.1 Dokumentacja pliku prj/inc/AdvancedStopwatch.hh

```
#include "Stopwatch.hh"
#include <iostream>
#include <fstream>
#include <string>
#include <iomanip>
```

Wykres zależności załączania dla AdvancedStopwatch.hh:



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



Komponenty

- class AdvancedStopwatch

  Klasa implementująca rozbudowany stoper.

Definicje

- #define MAX_LAPS 100
- #define BUFOR 10
- #define DOKLADNOSC 8

5.1.1 Opis szczegółowy

Plik zawiera implementację rozbudowanego stopera.

Definicja w pliku AdvancedStopwatch.hh.

5.1.2 Dokumentacja definicji

5.1.2.1 #define BUFOR 10

Definicja w linii 12 pliku AdvancedStopwatch.hh.

5.1.2.2 #define DOKLADNOSC 8

Definicja w linii 13 pliku AdvancedStopwatch.hh.

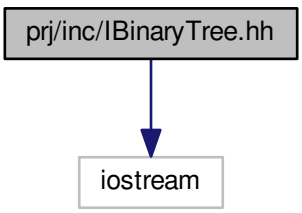
5.1.2.3 #define MAX_LAPS 100

Definicja w linii 11 pliku AdvancedStopwatch.hh.

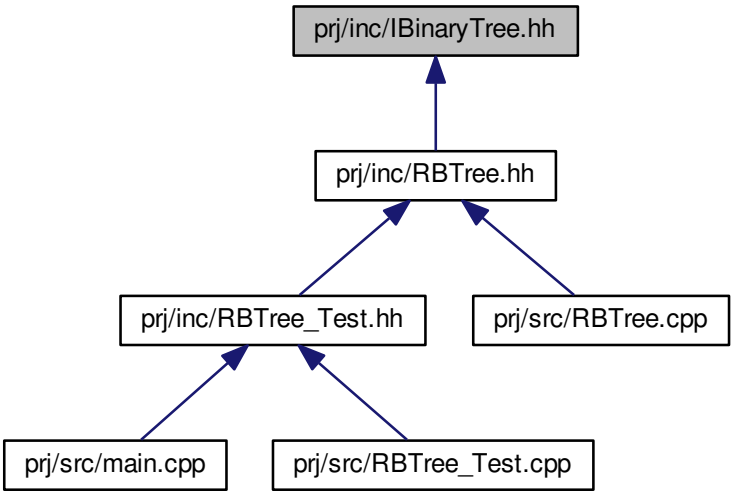
5.2 Dokumentacja pliku prj/inc/IBinaryTree.hh

```
#include <iostream>
```

Wykres zależności załączania dla IBinaryTree.hh:



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



Komponenty

- class IBinaryTree< Object >

5.2.1 Opis szczegółowy

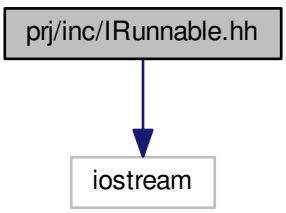
Plik zawiera interfejs drzewa binarnego.

Definicja w pliku IBinaryTree.hh.

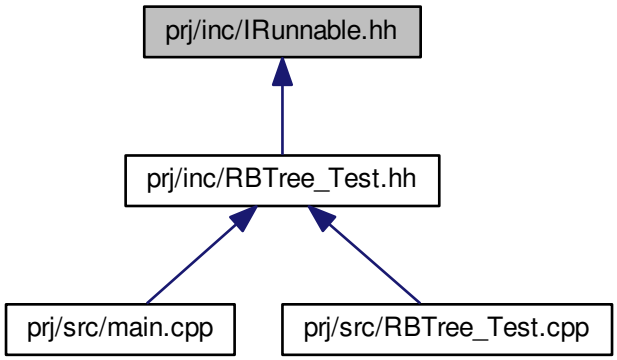
5.3 Dokumentacja pliku prj/inc/IRunnable.hh

```
#include <iostream>
```

Wykres zależności załączania dla IRunnable.hh:



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



Komponenty

- class IRunnable< Object >

  Klasa szablonowa modelująca interfejs "Biegacza".

5.3.1 Opis szczegółowy

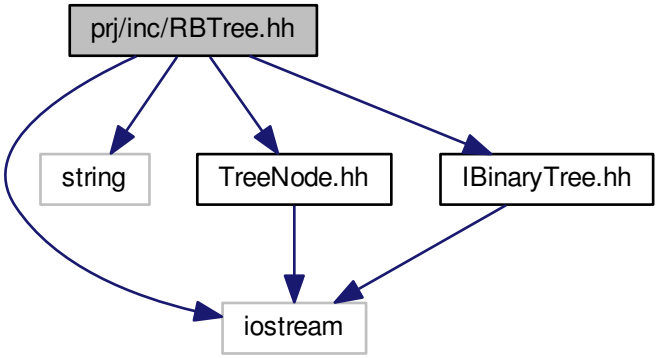
Plik zawiera interfejs obiektu, który można poddawać pomiarom czasu działania.

Definicja w pliku IRunnable.hh.

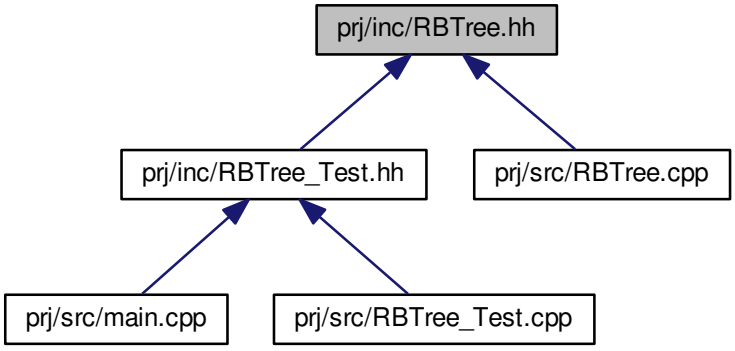
5.4 Dokumentacja pliku prj/inc/RBTree.hh

```
#include <iostream>
#include <string>
#include "TreeNode.hh"
#include "IBinaryTree.hh"
```

Wykres zależności załączania dla RBTree.hh:



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



Komponenty

- class RBTree< Object >

  Szablonowa klasa implementująca drzewo binarne czerwono-czarne.

5.4.1 Opis szczegółowy

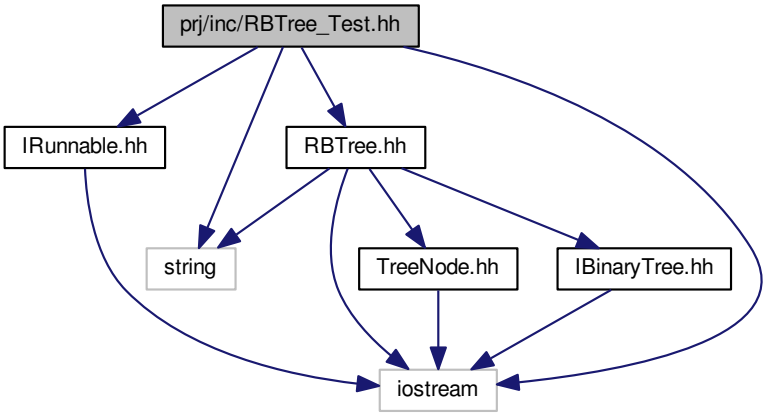
Plik zawiera definicję klasy implementującej drzewo binarne.

Definicja w pliku RBTree.hh.

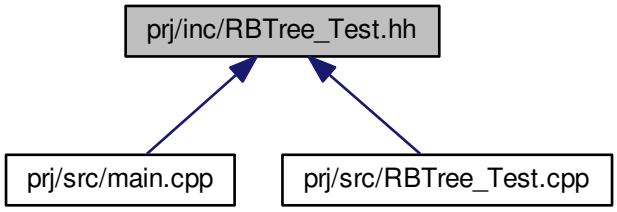
5.5 Dokumentacja pliku prj/inc/RBTree_Test.hh

```
#include <iostream>
#include <string>
#include "IRunnable.hh"
#include "RBTree.hh"
```

Wykres zależności załączania dla RBTree_Test.hh:



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



Komponenty

- class RBTree_Test< Object >

  Szablonowa klasa testowego drzewa czerwono-czarnego.

5.5.1 Opis szczegółowy

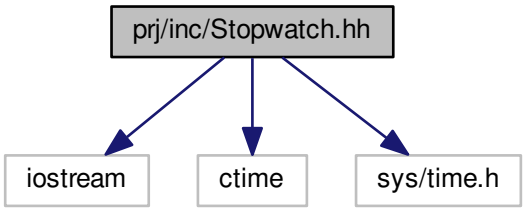
Plik zawiera implementację IRunnable w klasie drzewa czerwono-czarnego.

Definicja w pliku RBTree_Test.hh.

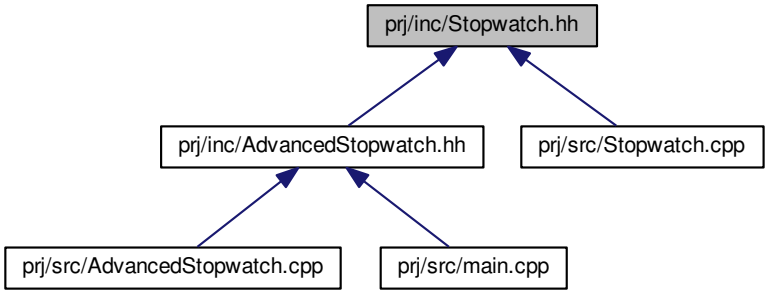
5.6 Dokumentacja pliku prj/inc/Stopwatch.hh

```
#include <iostream>
#include <ctime>
#include <sys/time.h>
```

Wykres zależności załączania dla Stopwatch.hh:



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



Komponenty

- class Stopwatch

  Klasa implementująca podstawowy stoper.

5.6.1 Opis szczegółowy

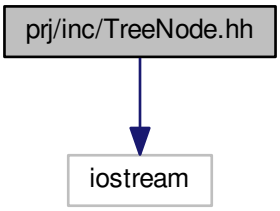
Plik zawiera implementację podstawowego stopera.

Definicja w pliku Stopwatch.hh.

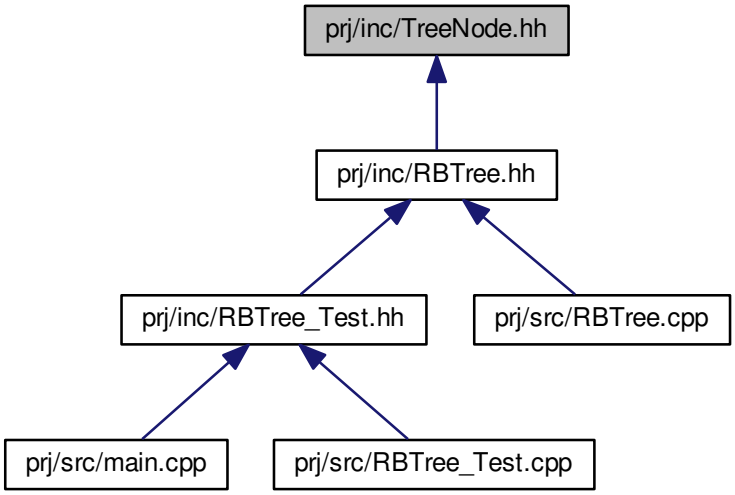
5.7 Dokumentacja pliku prj/inc/TreeNode.hh

```
#include <iostream>
```

Wykres zależności załączania dla TreeNode.hh:



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



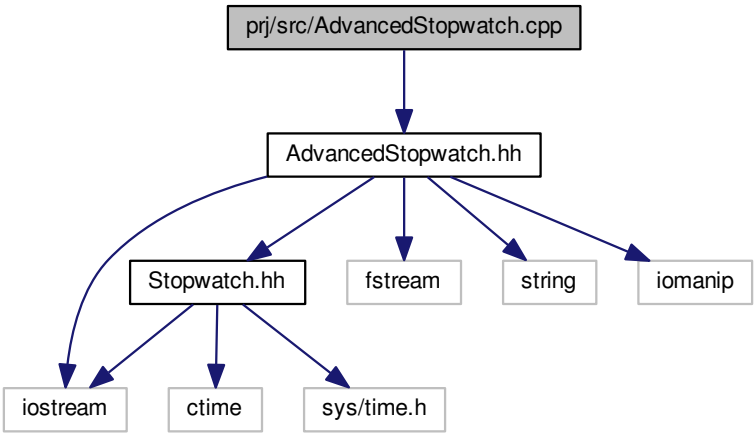
Komponenty

- class TreeNode< Object >

5.8 Dokumentacja pliku prj/src/AdvancedStopwatch.cpp

```
#include "AdvancedStopwatch.hh"
```

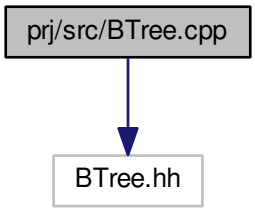
Wykres zależności załączania dla AdvancedStopwatch.cpp:



5.9 Dokumentacja pliku prj/src/BTree.cpp

```
#include "BTree.hh"
```

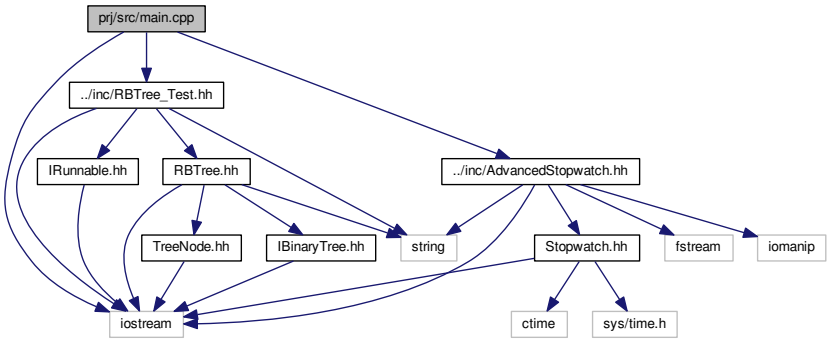
Wykres zależności załączania dla BTree.cpp:



5.10 Dokumentacja pliku prj/src/main.cpp

```
#include <iostream>
#include "../inc/RBTree_Test.hh"
#include "../inc/AdvancedStopwatch.hh"
```

Wykres zależności załączania dla main.cpp:



Funkcje

- int main ()

5.10.1 Dokumentacja funkcji

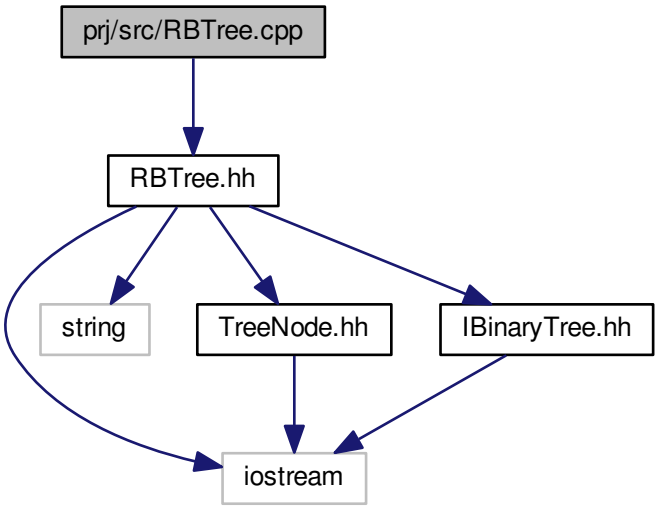
5.10.1.1 int main ( )

Definicja w linii 6 pliku main.cpp.

5.11 Dokumentacja pliku prj/src/RBTree.cpp

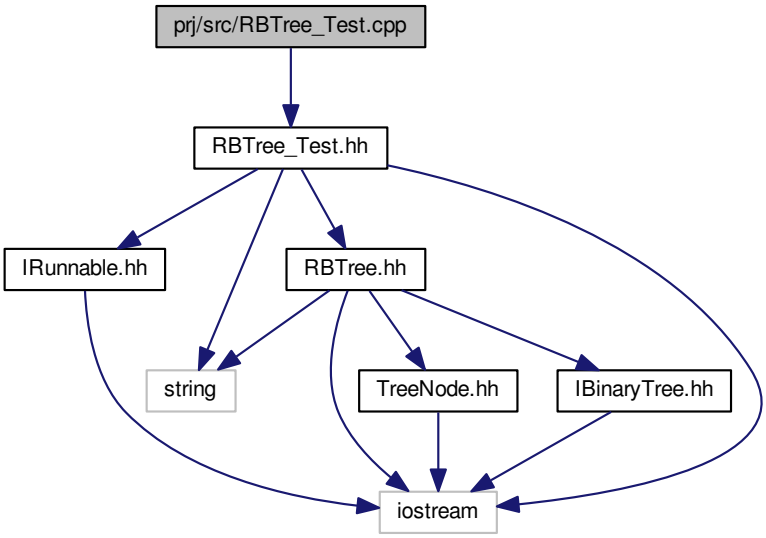
```
#include "RBTree.hh"
```

Wykres zależności załączania dla RBTree.cpp:



5.12 Dokumentacja pliku prj/src/RBTree_Test.cpp

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



5.13 Dokumentacja pliku prj/src/Stopwatch.cpp

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