Dokumentáció

Készítette Doxygen 1.8.3.1

Sun May 12 2013 15:33:53

Tartalomjegyzék

1. fejezet

NHF 2013 - QuadTree (Négy elágazású duplán láncolt generikus fa)

Feladat kiírás: Készítsen GENERIKUS duplán láncolt 4 elágazású fát (quad-tree)! Valósítsa meg az összes értelmes műveletet operátor átdefiniálással (overload), de nem kell ragaszkodni az összes operátor átdefiniálásához! Amennyiben lehetséges használjon iterátort!

Szerző

Tóth András (O8POUA)

2. fejezet

Osztálymutató

2.1. Osztálylista

Az összes osztály, struktúra, unió és interfész listája rövid leírásokkal:

3. fejezet

Fájlmutató

3.1. Fájllista

Az összes dokumentált fájl listája rövid leírásokkal:

4. fejezet

Osztályok dokumentációja

4.1. QuadTree< T >::iterator osztályreferencia

Iterátor.

`#include <QuadTree.hpp>`

Publikus tagfüggvények

- iterator (QuadTreeNode< T > *node=NULL)
  Konstruktor.
- QuadTreeNode< T > & **operator*** ()
- QuadTreeNode< T > * **operator->** ()
- iterator & operator++ ()
  Prefix operator++.
- iterator operator++ (int)
  Postfix operator++.
- bool operator== (const iterator &iter)
  Egyenlőség operátor.
- bool operator!= (const iterator &iter)

Védett attribútumok

- QuadTreeNode< T > * **node**

4.1.1. Részletes leírás

template<class T>class QuadTree< T >::iterator

Iterátor.

4.1.2. Tagfüggvények dokumentációja

4.1.2.1. template<class T> bool QuadTree< T >::iterator::operator!= (const iterator & *iter*) [inline]

Visszatérési érték

Igaz, ha nem egyezik meg a csomópont.

4.1.2.2. template<class T > QuadTree< T >::iterator & QuadTree< T >::iterator::operator++ ( )

Prefix operator++.

Visszatérési érték

Következő elem.

Ha a gyökér elemnél vagyunk (csak annak nincs szülője), akkor végigértünk az összes elemen.

A szülő hányadig gyerekén állunk?

Ha a következő gyerek levél, akkor ez lesz a következő elem.

Ha a következő gyerek nem levél, akkor a következő elem a legbaloldalibb levél a következő gyerek alatt.

Egyébként a szülő a következő elem.

4.1.2.3. template<class T> bool QuadTree< T >::iterator::operator== (const iterator & *iter*) [inline]

Egyenlőség operátor.

Visszatérési érték

Igaz, ha megegyezik az csomópont.

Ez a dokumentáció az osztályról a következő fájl alapján készült:

- QuadTree.hpp

4.2. Point< T > osztálysablon-referencia

Point (pont) osztály. A négy elágazású generikus fát ilyen pontokkal tölthetjük fel.

`#include <QuadTree.hpp>`

Publikus tagfüggvények

- Point ()

  Default konstruktor.
- Point (T data, double x=0, double y=0)

  Konstruktor adattal.
- bool operator== (Point &point)

  operator==, két pont egyenlősége vizsgálható. Igaz, ha a két pont minden adata megegyezik.
- bool operator!= (Point &point)

  operator!=, két pont egyenlősége vizsgálható. Igaz, ha a két pont valamelyik adata nem egyezik meg.
- T & **getData** ()

Barátok

- class **QuadTree**< **T** >
- class **QuadTreeNode**< **T** >
- std::ostream & **operator** (std::ostream &, const Point &)

4.2.1. Részletes leírás

template<class T>class Point< T >

Point (pont) osztály. A négy elágazású generikus fát ilyen pontokkal tölthetjük fel.

Ez a dokumentáció az osztályról a következő fájl alapján készült:

- QuadTree.hpp

4.3. QuadTree< T > osztálysablon-referencia

QuadTree (generikus négy elágazású fa) osztály. A négy elágazású generikus fa QuadTreeNode-okból épül fel.

`#include <QuadTree.hpp>`

Osztályok

- class iterator

  Iterátor.

Publikus tagfüggvények

- QuadTree (double width=100, double height=100)

  Konstruktor.
- ~QuadTree ()

  Destruktor.
- void insert (Point< T > data)

  Új elem beszúrása.
- unsigned depth () const

  Fa mélységének visszaadása.
- unsigned countNodes () const

  Fában lévő elemek megszámolása.
- Point< T > & find (Point< T > point)

  Adott pont megkeresése. Lehetőség van az adott pontban lévő adat megváltoztatására.
- Point< T > & find (T data)

  Adott adatot tároló pont megkeresése. Lehetőség van az adott pontban lévő adat megváltoztatására.
- iterator begin () const
- iterator end () const

Barátok

- class **QuadTreeNode< T >**
- std::ostream & **operator** (std::ostream &, const QuadTree< T > &)
- std::istream & operator>> (std::istream &is, QuadTree< T > &quadtree)

  Beolvasás. Az beolvasandó adatokat "(x;y): adat" alakban kapjuk.

4.3.1. Részletes leírás

template<class T>class QuadTree< T >

QuadTree (generikus négy elágazású fa) osztály. A négy elágazású generikus fa QuadTreeNode-okból épül fel.

4.3.2. Tagfüggvények dokumentációja

4.3.2.1. template<class T> iterator QuadTree< T >::begin ( ) const [inline]

Visszatérési érték

- Legbaloldalibb levél.

4.3.2.2. template<class T> unsigned QuadTree< T >::countNodes ( ) const [inline]

Fában lévő elemek megszámolása.

Visszatérési érték

Fa elemszáma.

4.3.2.3. template<class T > unsigned QuadTree< T >::depth ( ) const

Fa mélységének visszaadása.

Visszatérési érték

Fa mélysége.

4.3.2.4. template<class T> iterator QuadTree< T >::end ( ) const [inline]

Visszatérési érték

Az utolsó elem után mutat.

4.3.2.5. template<class T > Point< T > & QuadTree< T >::find (Point< T > *point*)

Adott pont megkeresése. Lehetőség van az adott pontban lévő adat megváltoztatására.

Paraméterek

point	- Pont, melyet keresünk.

Visszatérési érték

A keresett pont.

4.3.2.6. template<class T > Point< T > & QuadTree< T >::find (T *data*)

Adott adatot tároló pont megkeresése. Lehetőség van az adott pontban lévő adat megváltoztatására.

Paraméterek

data	- Adat, melyet keresünk.

Visszatérési érték

Az első olyan pont, amely ezt az adatot tartalmazza.

4.3.2.7. template<class T > void QuadTree< T >::insert (Point< T > *point*)

Új elem beszúrása.

Visszatérési érték

- Gyökérre mutató pointer.

Paraméterek

data	- Új pont / adat.
point	- Új pont / adat.

Ha a pont kívül esik a területről, akkor kivételt dob.

Egyébként keressük meg azt a levelet ahova be kéne szúrni az új elemet.

Szúrjuk be az elemet.

Ez a dokumentáció az osztályról a következő fájl alapján készült:

- QuadTree.hpp

4.4. QuadTreeNode< T > osztálysablon-referencia

QuadTreeNode (négy elágazású generikus fa csomópontja) osztály. A négy elágazású generikus fa ilyen csomópontokból épül fel.

`#include <QuadTree.hpp>`

Barátok

- class **QuadTree**< **T** >
- std::ostream & **operator** (std::ostream &, const QuadTreeNode &)

4.4.1. Részletes leírás

template<class T>class QuadTreeNode< T >

QuadTreeNode (négy elágazású generikus fa csomópontja) osztály. A négy elágazású generikus fa ilyen csomópontokból épül fel.

Ez a dokumentáció az osztályról a következő fájl alapján készült:

- QuadTree.hpp

4.5. String osztályreferencia

String osztály. A 'pData'-ban vannak a karakterek (a lezáró nullával együtt), 'len' a hossza. A hosszba nem számít bele a lezáró nulla.

`#include <String.h>`

Publikus tagfüggvények

- String (char ch)

  Konstruktor: egy karakterből.
- String (const char *p="")

  Konstruktor: egy egy nullával lezárt char sorozatból, ez a default konstruktor is.
- const char * c_str () const

  C stringet ad vissza.
- String (const String &s1)

  Másoló konstruktor.
- ~String ()

  Destruktor.
- String & operator= (const String &rhs_s)
- String operator+ (const String &rhs_s) const

  Két Stringet összefűz.
- char & operator[] (unsigned int idx)

  A string egy megadott indexű elemének referenciájával tér vissza.
- const char & operator[] (unsigned int idx) const

  A string egy megadott indexű elemének referenciájával tér vissza.

Barátok

- std::ostream & operator<< (std::ostream &os, const String &s0)

  Kiíró operátor.
- std::istream & operator>> (std::istream &is, String &s0)

  Beolvas az istream-ről egy szót egy String-be.

4.5.1. Részletes leírás

String osztály. A 'pData'-ban vannak a karakterek (a lezáró nullával együtt), 'len' a hossza. A hosszba nem számít bele a lezáró nulla.

String osztály deklarációja.

Szerző

Tóth András (O8POUA)

4.5.2. Konstruktorok és destruktorok dokumentációja

4.5.2.1. String::String (char *ch*)

Konstruktor: egy karakterből.

Paraméterek

Ch	egy karakter.

4.5.2.2. String::String (const char * *p* = " ")

Konstruktor: egy egy nullával lezárt char sorozatból, ez a default konstruktor is.

Paraméterek

p	Pointer a C stringre.

4.5.2.3. String::String (const String & *s1*)

Másoló konstruktor.

Paraméterek

s1	String, amiből létrehozzuk az új String-et.

4.5.3. Tagfüggvények dokumentációja

4.5.3.1. const char* String::c_str ( ) const [inline]

C stringet ad vissza.

Visszatérési érték

Nullával lezárt karaktersorozatra mutató pointer.

4.5.3.2. String String::operator+ (const String & *rhs_s*) const

Két Stringet összefűz.

Paraméterek

rhs_s	jobboldali String.

Visszatérési érték

Új String, ami tartalmazza a két stringet egymás után.

4.5.3.3. String & String::operator= (const String & *rhs_s*)

Paraméterek

rhs_s	jobboldali String.

Visszatérési érték

Baoldali string.

4.5.3.4. char & String::operator[] (unsigned int *idx*)

A string egy megadott indexű elemének referenciájával tér vissza.

Paraméterek

idx	Karakter indexe.

Visszatérési érték

Karakter referenciája. Indexelési hiba esetén const char* kivételt dob.

4.5.3.5. const char & String::operator[] (unsigned int *idx*) const

A string egy megadott indexű elemének referenciájával tér vissza.

Paraméterek

idx	Karakter indexe.

Visszatérési érték

Karakter referenciája. Indexelési hiba esetén const char* kivételt dob.

4.5.4. Barát és kapcsolódó függvények dokumentációja

4.5.4.1. std::ostream& operator<< (std::ostream & *os,* const String & *s0*) [friend]

Kiíró operátor.

Paraméterek

os	Ostream típusú objektum.
s0	String, amit kiírunk.

Visszatérési érték

os

4.5.4.2. std::istream& operator>> (std::istream & *is,* String & *s0*) [friend]

Beolvas az istream-ről egy szót egy String-be.

Paraméterek

is	Istream típusú objektum.
s0	String, amibe beolvas.

Visszatérési érték

is

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

- String.h
- String.cpp

5. fejezet

Fájlok dokumentációja

5.1. QuadTree.hpp fájlreferencia

Osztályok

- class Point< T >

  Point (pont) osztály. A négy elágazású generikus fát ilyen pontokkal tölthetjük fel.
- class QuadTree< T >

  QuadTree (generikus négy elágazású fa) osztály. A négy elágazású generikus fa QuadTreeNode-okból épül fel.
- class QuadTreeNode< T >

  QuadTreeNode (négy elágazású generikus fa csomópontja) osztály. A négy elágazású generikus fa ilyen csomópontokból épül fel.
- class Point< T >

  Point (pont) osztály. A négy elágazású generikus fát ilyen pontokkal tölthetjük fel.
- class QuadTreeNode< T >

  QuadTreeNode (négy elágazású generikus fa csomópontja) osztály. A négy elágazású generikus fa ilyen csomópontokból épül fel.
- class QuadTree< T >

  QuadTree (generikus négy elágazású fa) osztály. A négy elágazású generikus fa QuadTreeNode-okból épül fel.
- class QuadTree< T >::iterator

  Iterátor.

Függvények

- template<class T >
  std::ostream & operator<< (std::ostream &os, const Point< T > &point)

  Point (pont) kiírása "(x;y): adat" alakban.
- template<class T >
  std::ostream & operator<< (std::ostream &os, const QuadTree< T > &quadtree)

  Fa kiírása iterátor használatával.
- template<class T >
  std::ostream & operator<< (std::ostream &os, const QuadTreeNode< T > &node)

  Csomópont kiírása.

5.1.1. Függvények dokumentációja

5.1.1.1. template<class T > std::ostream & operator<< (std::ostream & *os,* const QuadTreeNode< T > & *node*)

Csomópont kiírása.

Ha van adat, akkor ki kell írni.

Tárgymutató