

HÁZI FELADAT

Programozás alapjai 2.

Pontosított feladatspecifikáció

Borbola Martin

AC380P

2016. április 17.

TARTALOM

1. Feladat.....	1
2. Pontosított feladatspecifikáció	2
3. Terv	2
4. Objektum terv	2
5. Algoritmusok	3
6. Tesztprogram algoritmusok	3

1. Feladat

Programozás alapjai II. házi feladat
Borbola Martin (AC380P) részére:

Készítsen GENERIKUS bináris fát! A kulcsok közötti rendezettséget a szokásos relációs operátorokkal vizsgálja, amit szükség esetén specializál!

Valósítsa meg az összes értelmes műveletet operátor átdefiniálással (overload), de nem kell ragaszkodni az összes operátor átdefiniálásához! Amennyiben lehetséges használjon iterátort!

Demonstrálja a működést külön modulként fordított tesztprogrammal! A megoldáshoz NE használjon STL tárolót vagy algoritmust!

A tesztprogramot úgy specifikálja, hogy az parancssoros batch alkalmazásként (is) működjön, azaz a szabványos bemenetről olvasson, és a szabványos kimenetre, és/vagy a hibakimenetre írjon! Amennyiben a feladat teszteléséhez fájlból, vagy fájlokból kell input adatot olvasnia, úgy a fájl neve *.dat alakú legyen!

2. Pontosított feladatspecifikáció

A feladat egy generikus bináris fa elkészítése. A fa típusát sablon paraméterként lehet megadni.

Az automatikusan létrejövő tagfüggvények mellett (másolás, értékadás, létrehozás, megszüntetés) keresést, beillesztést és a törlés műveletet valósítom meg, lehetőleg operátor átdefiniálással. Ezenkívül lehet fájlba menteni és onnan betölteni.

A teszteléséhez egy olyan programot készítek, ami különböző adattípusokkal létrehozott tömbökkel a standard inputról beolvasott adatok alapján műveleteket végez.

3. Terv

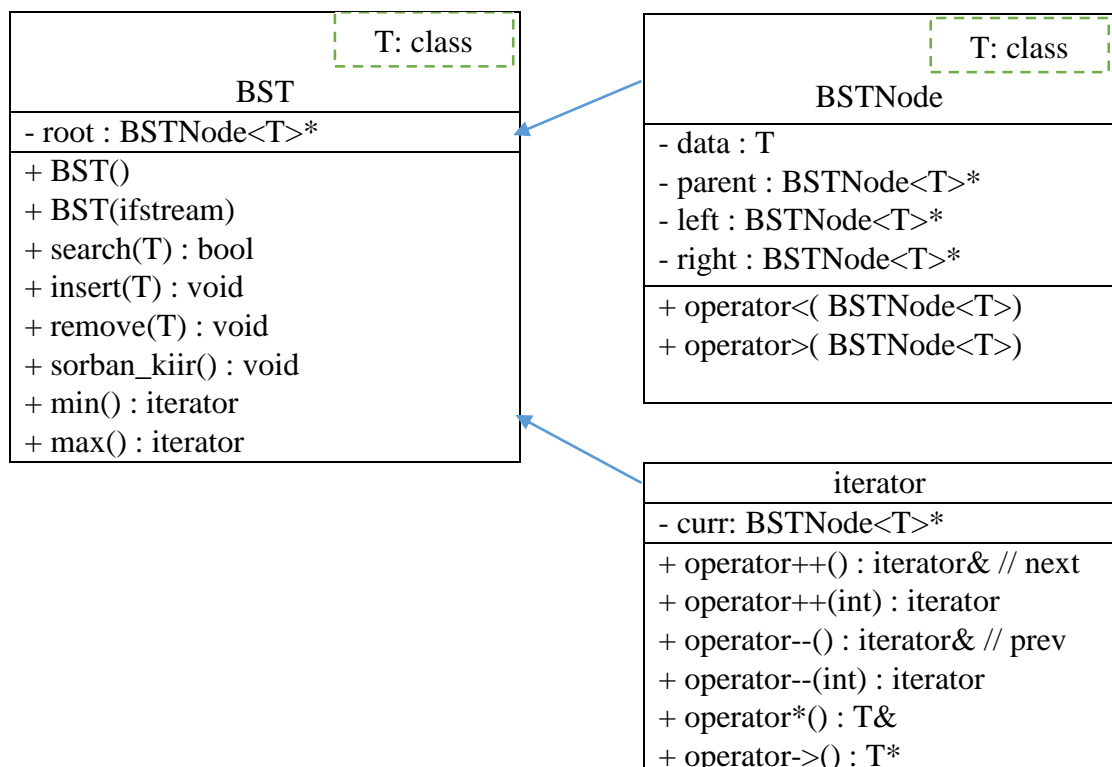
A feladat egy objektum és a tesztprogram megtervezését igényli.

4. Objektum terv

A generikus bináris fát egyetlen sablonnal fogom megvalósítani, ezen belül egy belső struktúraként a fa elemeit és egy belső osztályként az iterátort. A sablon sablonparaméterként veszi át a fa elemeinek típusát.

A BSTNode belső struktúra.

Az iterator belső osztály, csak a sorban_kiir() függvénynél használom.



5. Algoritmusok

BST konstruktor: fájlból is lehet

BST::search: megkeresi az adott értékű elemet (rekurzívan)

BST::insert: beilleszt egy adott értékű elemet (rekurzívan)

BST::remove: kitörli az adott értékű elemet (rekurzívan)

BST::sorban_kiiraz alapértelmezett kimenetre kiírja a fa elemét (iterátorral)

6. Tesztprogram algoritmusok

A tesztprogram beolvass fájlból egy bináris fát, majd ellenőrzi az algoritmusokat.