Dinamikus string - Terv

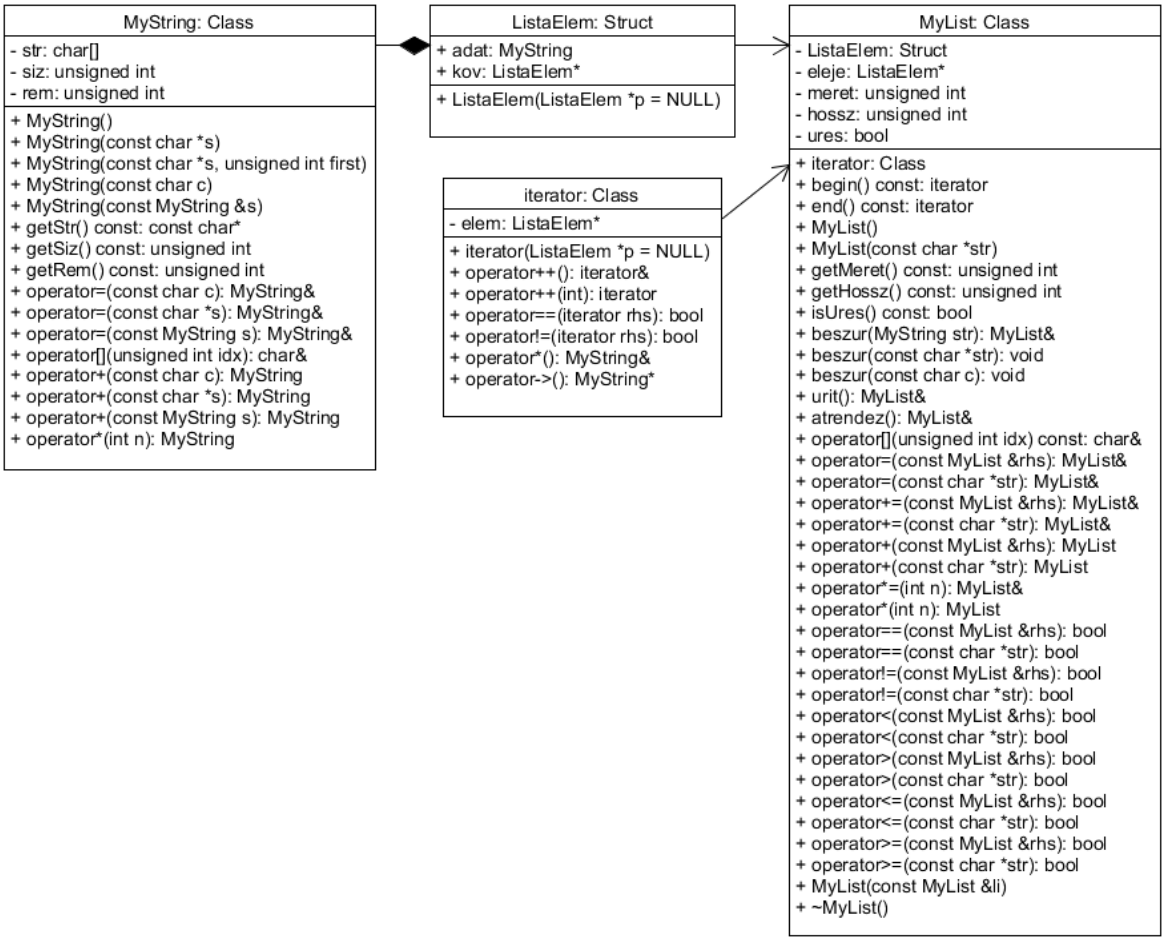
Pontosított feladatspecifikáció:

Ez egy olyan dinamikus string, ami a karaktereket egy 20 karakteres tárolókból épített láncolt lista tárolja. A feladat magában foglalja az értelmes műveletek operátorainak átdefiniálását. Az osztály rendelkezni fog iterátorral is. A program futása annyiból fog állni, hogy képesek leszünk akármilyen hosszúságú karaktersorozatot beolvasni az adatszerkezetbe. Ellenőrzésképp ki is írja azt, valamint a hosszát és a 20 karakteres tárolók számát.

Megvalósítandó operátorok:

- indexelés (adott karaktert adja vissza)
- értékadás
- összeadás (két string konkatenálása)
- szorzás (paraméterként adott string tartalmát megtöbbszörözi)
- összehasonlító operátorok (== , != , < , > , <= , >=)
- kiíró << operátor

Osztálydiagram:



Néhány algoritmus leírása:

MyString(const char *s): *Paraméterként kapott karaktersorozatból csinál 20 karakteres tárolót. Ha a karaktersorozat hosszabb, mint 20 karakter, akkor az első 20 karaktert teszi a tárolóba.*

MyString(const char *s, unsigned int first): *Ugyanazt csinálja, mint a fenti függvény annyi kivétellel, hogy meg tudjuk adni, hogy honnan kezdve csináljon egy 20 karakteres tárolót.*

atrendez(): *Amennyiben a láncolt listában vannak olyan tárolók, amelyek nincsenek feltöltve teljesen és az nem az utolsó tároló, akkor ez a függvény átrendezi a karaktereket, hogy ne legyenek üres helyek a tárolókban.*

urit(): *Kitörli a tárolókat a láncolt listából. Végeredményképp egy üres láncolt listát kapunk. Működése hasonló a destruktorhoz.*

operator=(const char *str): *Paraméterként kapott karaktersorozatot feldarabolja 20 karakteres tárolókra, majd csinál belőlük egy láncolt listát.*

operator[](unsigned int idx): *Láncolt listát tudjuk indexelni és visszaad egy karaktert. Paraméterként kapott indexből kiszámolja, hogy melyik tároló hányadik karaktere az, amit keresünk.*