Házi feladat

Programozás alapjai 2.

Terv

Bozi Roland Sándor
EO7EGE

2018. április 18.

---

1. Feladat

Készítsen olyan dinamikus sztringet, melyben a sztring karaktereit 20 karakter tárolására alkalmas tárolókból kialakított láncolt listában tárolja! Valósítsa meg az összes értelmes műveletet operátor átdefiniálással (overload), de nem kell ragaszkodni az összes operátor átdefiniálásához! Legyen az osztálynak iterátora is! Legyen képes az objektum perzisztens viselkedésre!

Specifikáljon egy egyszerű tesztfeladatot, amiben fel tudja használni az elkészített adatszerkezetet! A tesztprogramot külön modulként fordított programmal oldja meg! A megoldáshoz ne használjon STL tárolót!

2. Pontosított feladatspecifikáció

A feladat egy dinamikus sztring osztály elkészítése. A MyString osztály fogja a sztringet tárolni, melyet egy absztrakt Serialize osztályból fogok származtatni. Ez biztosítja a perzisztens viselkedést, valamint ennek hála a MyString egy heterogén kollekcióba lesz gyűjthető.

A MyString osztálynak lesz egy Iterator és egy Cell nevű alosztálya. A Cell egy 20 karakter tárolására alkalmas fix méretű láncolt lista lesz. Az Iterator osztályt a RandomAccessIterator alapján próbálom megalkotni.

3. Terv

A terv egy heterogén kollekcióba gyűjthető dinamikus sztring osztály létrehozása, valamint emellé egy bemutató példaprogram megírása.

3.1. Serialize osztály

A Serialize osztálynak lesz két virtuális void metódusa, egy read() és egy write(), előbbi paraméterként egy std::istream, másik egy std::ostream referenciát kap majd paraméterként.

3.2 MyString osztály

20 karaktert tároló karaktertömbökben tárolja el a sztringet, ezek első elemének címét eltárolja dinamikusan.

A metódusokhoz ötleteket a Stringes STL tárolókból vettem, az alábbiakat terveztem:

- size(): Visszaadja a sztring hosszát.
- clear(): Kitörli a sztring tartalmát.
- capacity(): A lefoglalt terület méretét adja vissza.
- empty(): Megmondja, hogy üres-e a sztring
- at(): Megadott indexű elem referenciájával tér vissza.
- begin(): Az iterátort az elejére viszi a sztringnek.
- end(): Az iterátort a sztring vége után teszi.

Túlterhelhető operátorok: =, ==, !=, +, +=, <<, >>, [].

Konstruktornak lehet majd megadni:

- semmit
- karaktertömböt
- karaktert

Ezeken kívül lesz természetesen explicit másoló konstruktor a dinamikus memóriakezelés miatt.

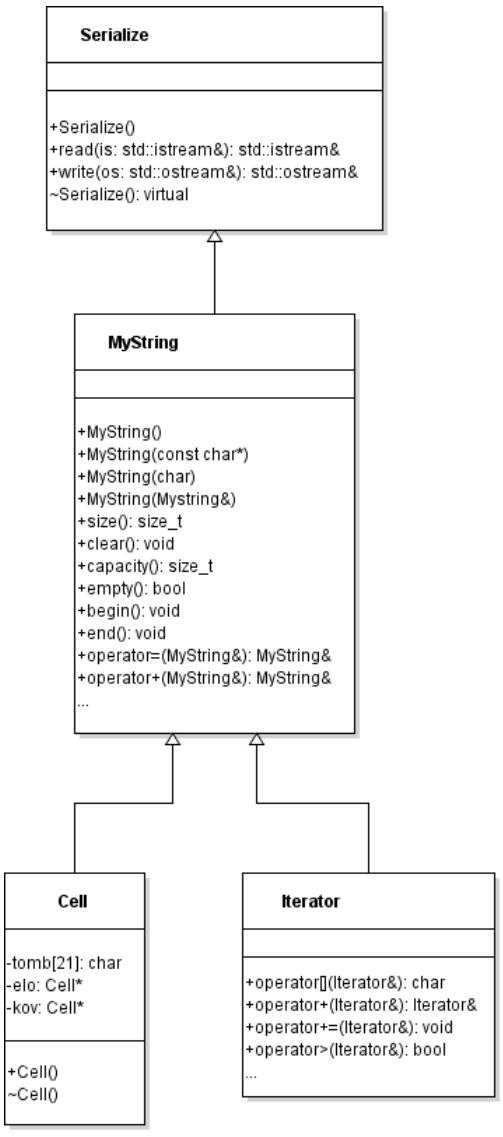
3.2.1 Iterator osztály

Ebben az osztályban a RandomAccessIterator metódusait szeretném tehát megvalósítani, úgy, hogy működjenek melyek az összeadó, kivonó, összehasonlító és a [] operátorok túlterhelései.

3.2.2 Cell osztály

Egy fix, 20 méretű karaktertömböt, és 2 darab Cell típusú pointert tartalmaz, melyek az előtte és mögötte lévő elemre mutatnak a listában.

Konstruktorja a cellát kinullázza.



3.3 Teszt program

A teszt programmal a cél A MyString osztály működésének, az Iterátornak és a Serialize osztály általános működésének bemutatása.

Felhasználói Dokumentáció

A „program” egy felhasználható dinamikus sztring osztályt hoz létre. Különlegessége, hogy a sztringeket nem egyben, hanem 20 méretű láncolt listákban tárolja.

Ahhoz, hogy tudjuk használni az osztály, először is programunk elején be kell „include-olni” a dstring.h nevű header fájlt. (Persze a hozzá tartozó dstring.cpp fájl is elérhető kell, hogy legyen.) Ezután teljességében használható lesz az osztályunk.

Fogunk tudni létrehozni üres, karaktert tartalmazó, vagy akár szöveget is tartalmazó sztringet.

Az így létrehozott objektum a Serialize osztálynak köszönhetően heterogén kollekcióba gyűjthető, valamint perzisztens viselkedésre is képes, tehát elmenthető.

Ezután a legtöbb STL string osztálynak megfelelő függvényt, illetve operátort használhatjuk. (Pl.: empty(), size(), capacity(), at(), c_str(), [],+,+=,=,==,!= stb.)

Az osztály rendelkezik iterátorral is, melyet RandomAccessIterator-szerűen próbáltam megvalósítani, tehát megfelel annak a követelményeinek.

Főbb operátorok/függvények:

size(): Visszaadja a tárolt karakterek számát.

capacity(): Visszaadja a lefoglalt terület méretét.

clear(): Kitörli a sztring tartalmát.

c_str(): Visszaadja a sztring tartalmát C sztingként.

operator=: Értékadó operátorként funkcionál. (Használható MyString-re, C sztringre, karakterre.)

operator+=: Hozzáfűz egy másik MyString-et a sztringhez.

operator==/operator!=: megmondja, hogy a két sztring egyenlő/nem egyenlő(-e).

Ezeken kívül vannak még az iterátor operátorai és függvényei is. Ezek közül a fontosabbak:

operator*(): Ezzel dereferálható az iterátor.

operator< és operator<=: Iterátorokat hasonlíthatunk vele össze. Ezekre van visszavezetve a többi összehasonlító operátor is.

value(): Visszaadja az iterátor által mutatott értéket

Ezeken kívül a többi operátor: +,++,+=,-,--,-=,==,!=,=,[].