

Labb 5 Iterators

Implementera iteratorer som itererar över den String klass som ni gjorde i Labb 3. Läs gärna på hur iteratorerna för STL containers fungerar. STL står för C++ Standard Template Library.

Syftet med en iterator är att den ska fungera på nästan samma sätt som en pekare. Läs mera i C++ primer page 106 ff. Se även <http://en.cppreference.com/w/cpp/string>.

Observera att göra en helt korrekt implementering av iterators som stämmer helt med STL-systemet är ganska mycket kod. Vi nöjer oss med att implementera det som krävs för att `std::stable_sort` ska fungera.

En iterator itererar över elementen i ett container objekt. I vårt fall så är det String objekten från den tidigare labben vi skall iterera över. Observera att vi behöver lägga till några saker i String klassen för att det ska fungera.

Uppgift:

Ni ska göra alla två/fyra iteratorer typerna:

- iterator
- const_iterator (VG)
- reverse_iterator
- const_reverse_iterator (VG)

I String klassen lägg till: (const versionerna bara för VG).

- typedefs för iterator, const_iterator, reverse_iterator och const_reverse_iterator
- funktionerna begin, end, cbegin, cend, rbegin, rend, crbegin och crend.

Iteratorerna ska kunna göra:

- `*it`, `--it`, `++it`, `it++`, `(it+i)`, `(it-it)`, `it[i]`, `==`, `!=` och `<`
- default constructor, copy constructor och tilldelning (`=`)
- iteratorn ska innehålla alla typedef (använd `std::iterator`)

Ett tips: Programmet nedan skriver ut "jeh"

```
String s("hej");
for (String:: reverse_iterator it=s.rbegin(); it!= s.rend(); ++it)
    cout << *it;
```

Observera att ni inte får använda iterator adapters från STL för att lösa er uppgift.

Testprogrammet i Main.cpp

Observera att det testprogram som finns i Main.cpp bara är en hjälp och varken fullständigt eller garanterat helt korrekt. Det är möjligt att testprogrammet kör felfritt fast er lösning är felaktig. Det är även möjligt – men inte troligt – att er lösning är korrekt fast testprogrammet inte kör/kompilerar felfritt. Programmet består av flera filer. Se filerna på ITSL. I filen VG.h definieras om ni testar VG eller ej.

Krav för G: Gör ovanstående, utom const-iteratorerna

Krav för VG:

Gör även const-iteratorerna. Ert program skall även ha alla const på rätt plats, vara effektiva och sakna onödig kodupprepning (dvs. någon form av ärvning behövs).

Ett sätt är att ärva från en IteratorBase som tar "iterator typen" och "value typen" som templateparametrar. Detta gör att implementeringen av `const_iterator` och `iterator` kan dela på mycket av koden. Om man lägger till en tredje templateparametrar "bool Reverse" så kan man ytterligare minska på koddupliceringen.