Skilaverkefni II

Lýsing á íslensku

Í þessu verkefni eigið þið að útfæra hugræna gagnatagið `ArrayList` sem sniðmátsklasa. Grunnurinn að klasanum (arraylist.h og arraylist.cpp) er í möppunni `Verkefni` í repo https://github.com/reykjavik-university/2016-T-201-GSKI/. Grunnurinn er það sem við fórum yfir í fyrirlestri í viku 1 en núna fylgir með breytt aðalforrit sem notar sniðmát.

Ykkar verkefni er að breyta grunninum þannig að `ArrayList` virki sem sniðmátsklasi og að auki að bæta við þeim meðlimaföllum (og hugsanlega hjálparföllum) sem á vantar þannig að hið nýja aðalforrit virki.

Upphafsstærð (`maxSize`) á `ArrayList` er sett í smiðnum. Ef `ArrayList` þarf á meira plássi að halda (sem getur komið upp þegar `append` eða `insert` föllin eru keyrð) þá á klasinn að tvöfalda stærð hins undirliggjand kviklega fylkis. Þetta skal vera gert þannig að nýju kviklegu fylki er úthlutað, stökin afrituð yfir í nýja fylkið og að lokum skal eyða gamla fylkinu.

Aðalforritið les fyrst inn eina heiltölu, `initialSize`, sem segir til um hver upphafsstærð listans á að vera. Næsta heiltala, `count`, segir til um hversu mörg stök fylgja á eftir og síðan koma `count` gildi sem mynda stök listans. Ofangreint er fyrst gert fyrir heiltölulista og síðan fyrir strengjalista.

Nokkur atriði:

- `insert(T elem)` fallið setur stakið `elem` inn til vinstri við núverandi stak (sem er í `currElemPos`). Eftir keyrslu á `insert()` hefur gildið á `currElemPos` ekki breyst.
- `moveToEnd()` fallið færir `currElemPos` til hægri við síðasta stak, þ.e. eftir keyrslu `moveToEnd()` er `currElemPos = currSize`.
- `remove()` fallið fjarlægir stakið í `currElemPos` , en `clear()` fallið hreinsar listann.
- Þið megið nota `assert()` þar sem verið er að reyna að nálgast stök sem eru ekki í listanum eða fara í stöðu sem er ekki í listanum. Þetta á við t.d. í `remove()`, `value()` og `moveToPos()`.

English description

In this project, you need to implement the abstract data type `ArrayList` as a template class. The base for the class (arraylist.h og arraylist.cpp) is in the folder `Verkefni` in the repo https://github.com/reykjavik-university/2016-T-201-GSKI/. The base is what we covered in class in week 1, but now you are given a new main program which uses templates.

Your task is to change the base such that `ArrayList` works as a template class and, additionally, you need to add the missing member functions (and possibly helper functions) for the main program to work.

The initial size (`maxSize`) for `ArrayList` is set in the constructor. If `ArrayList` needs more space (which can happen when the `append` or `insert` functions are called) then the class should double the size of its underlying dynamic array. You should carry this out by dynamically allocating a new array, copy the elements to it, and, finally, delete the old array.

The main program first reads in a single integer, `initialSize`, denoting what the inital size should be. The next integer, `count`, denotes how many elements follow, and then `count` values are given which comprise the list elements. The above is first carried out for a list of integers and then a list of strings.

A few issues:

- The `insert(T elem)` function inserts the `elem` on the left to the current element (which is in `currElemPos`). After the execution of `insert()`, the value of `currElemPos` has not changed.
- The `moveToEnd()` function moves `currElemPos` to the right of the last element, i.e. after the execution of `moveToEnd()`, `currElemPos = currSize`.
- The `remove()` function removes the element in `currElemPos`, whereas the `clear()` function clears the list.
- You can use `assert()` when the user tries to access elements which are not in the list or move to a position which is not part of the list. This can, for example, happen in `remove()`, `value()` and `moveToPos()`.

Dæmi um inntak / Example input:

```
5
10
1 2 3 4 5 6 7 8 9 10

10
4
joi siggi magga gunna
```