Inlämningsuppgift A

Generella krav:

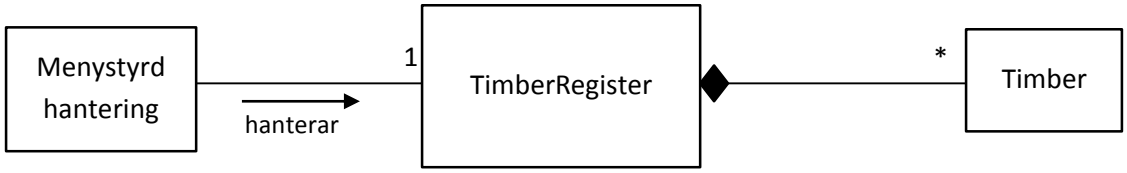
Alla klasser ska ha privata medlemsvariabler.
Inga variabler får vara globala.
Konstanter bör vara globala.
Alla funktioner som kan vara konstanta (const) ska vara det.
Alla parametrar som bör vara konstanta (const) ska vara det.
Alla klasser delas upp i h-fil och cpp-fil.
Lösningen får inte generera några minnesläckor.
continue, goto, early returns är inte tillåtna, break endast i samband med ev switch/case.
Använd `_CrtSetDbgFlag(_CRTDBG_ALLOC_MEM_DF | _CRTDBG_LEAK_CHECK_DF);`
i main()-funktionen för att detektera minnesläckor.

Bygghandel

Din uppgift är att skriva ett program för att hjälpa en bygghandel att hantera sitt virkeslager.
Följande funktionalitet ska finnas:

1. Lägga till ett nytt virke till lagret (inga dubletter ska finnas, vilket löses mha == operatorn vilken beskrivs längre ner)
2. Presentera all information om alla virken som finns på lager
3. Presentera endast virkesdimensionen för de virken där antal meter på lager understiger ett av användaren angivet värde
4. Presentera totalkostnaden för allt virke som finns på lagret
5. Ta bort ett virke från lagret givet virkesdimensionen (inga ”hål” i arrayen)
6. Ändra uppgifter (pris per löpmeter samt antal löpmeter på lager) för en av användaren angiven virkesdimension
7. Spara alla virken på fil där filnamnet matas in av användaren
8. Läsa alla virken från fil där filnamnet matas in av användaren

Ditt system ska vara uppbyggt enligt:



Klassen **Timber** beskriver virke. Egenskaper för virke ska vara dimension (ex-vis 28x120), antal löpmeter på lager samt pris per löpmeter (ex-vis 13:95). Tillför de medlemsfunktioner som behövs (bl a string toString()) samt implementera == operatorn vilken baseras på dimension.

Klassen **TimberRegister** ska ha en medlemsvariabel av typen Timber**. Konstruktorn i denna klass ska skapa en array med en initial storlek men arrayen ska kunna expandera vid behov.
Utöver funktionalitetsrelaterade medlemsfunktioner ska också kopieringskonstruktor(copykonstruktor), tilldelningsoperator och destruktor implementeras.
För att kunna uppfylla funktionalitetskrav 2 ovan behövs en medlemsfunktion som tar emot en string-array med tillräcklig kapacitet. Denna array ska fyllas på med strängarna från Timber-objektens toString()-funktion.
För att användaren kunna dimensionera denna array behövs även en medlemsfunktion från vilken antalet inlagda virken kan erhållas.
För funktionalitetskrav 3 krävs ytterligare varianter av ovan nämnda medlemsfunktioner.

Ingen in- eller utmatning får ske i medlemsfunktionerna i klasserna TimberRegister och Timber.

Nedan finner du ett första utkast på testningsförfarande för klassen TimberRegister.
Utför detta och gör tillägg (även denna testfil (TestTimberRegister.cpp) lämnas in).

Testning:

1. Skapa ett objekt av typen TimberRegister (låt startkapaciteten vara 2).

2. Lägg in 2 virken.
Skriv ut de strängar som erhålls i arrayen som medlemsfunktionen för TimberRegister-objektet fyller i.

3. Lägg till ytterligare ett virke (vilket ska medföra att arrayen i TimberRegister-objektet expanderar).
Skriv ut de strängar som erhålls i arrayen som medlemsfunktionen för TimberRegister-objektet fyller i.

4. Försök ta bort ett virke som inte finns inlagt och kontrollera att ingen borttagning gjordes samt generera en utskrift av detta.

5. Ta bort ett virke som finns inlagt och kontrollera att borttagningen genomförts.

6. Tag bort de resterande 2 virkena från TimberRegister-objektet och kontrollera att det därefter inte innehåller några virken.

7. Lägg till 2 nya virken till TimberRegister-objektet och kontrollera att dessa finns inlagda.

8. Testa på lämpligt sätt kopieringskonstruktor(copykonstruktor) och tilldelningsoperator.

Avsluta programmet och kontrollera slutligen att inga minnesläckor upptäckts!

Därefter implementerar du den menystyrda delen av systemet (TimberRegisterSystem.cpp).

I det menystyrda systemet ska separata delproblem hanteras i separata funktioner.