Inlämningsuppgift B

Generella krav:

Alla medlemsvariabler ska vara **privata**.

Inga variabler får vara globala.

Alla funktioner som kan vara konstanta (const) ska vara det.

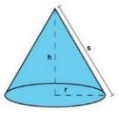
Alla parametrar som bör vara konstanta (const) ska vara det.

Alla klasser delas upp i h-fil och cpp-fil.

Lösningen får inte generera några minnesläckor.

Geometriska figurer

Några exempel på geometriska figurer är

Kon  och Låda (Parallellepiped). 

En kon (Cone) beskrivs av sin radie och höjd. Volymen beräknas enligt:

volym = π*(radie)^2*höjd/3

En låda (Box) beskrivs av sin längd, bredd och höjd. Volymen beräknas enligt:

volym = längd * bredd * höjd

Implementera en arvshierarki för *kon* och *låda* där

- arv nyttjas på ett logiskt och rimligt sätt för att generalisera och specialisera
- inga överflödiga medlemsvariabler förekommer
- basklassen (Shape) är abstrakt
- inga funktioner har tomma kroppar (undantaget destruktorer)
- dynamisk bindning kommer att utnyttjas på ett logiskt och rimligt sätt
- ingen onödig duplicering av kod görs
- alla figurer kan beräkna och returnera sin volym
- alla figurer ska kunna returnera sitt ”innehåll” som en sammansatt sträng (funktionssignatur string toString())
- ingen utskrift sker i någon av medlemsfunktionerna

Vidare ska nödvändiga set- och get-funktioner implementeras.

Skapa därefter en klass (ShapeRegister) som representerar ett register över geometriska figurer. Objekt av denna klasstyp ska innehålla koner och lådor. Det finns ingen begränsning avseende antalet geometriska figurer och du ska därför implementera denna relation genom användande av en (1) **dynamiskt allokerad array innehållande pekare av basklasstyp** (ej vector-objekt).
Inga överflödiga medlemsvariabler i denna klass heller!

I denna klass ska följande medlemsfunktioner finnas:

• **bool addCone(<parametrar för en kon>)** för att lägga till en kon. Det ska inte förekomma mer än 1 geometrisk figur med en viss höjd, dvs höjden ska vara unik.

• **bool addBox(<parametrar för en låda>)** för att lägga till en låda. Det ska inte förekomma mer än 1 geometrisk figur med en viss höjd, dvs höjden ska vara unik.

• **bool removeShape(<parameter för höjd>)** för att ta bort en geometrisk figur givet höjden (höjden betraktas som unik).

• **bool getAllShapesAsStrings(string arr[], int capOfArr)** för att få strängar som representerar alla geometriska figurer i arrayen arr med kapaciteten capOfArr.

• **bool getAllConesAsStrings(string arr[], int capOfArr)** för att få strängar som representerar alla koner i arrayen arr med kapaciteten capOfArr.

• **bool getAllBoxesAsStrings(string arr[], int capOfArr)** för att få strängar som representerar alla lådor i arrayen arr med kapaciteten capOfArr.

• **bool editACone(<parametrar för en kon>)** för att kunna editera en kon (ändra värde på en eller flera egenskaper). Funktionens returvärde ska indikera om editeringen lyckades eller om den inte gjorde det (konen hittades inte / inga ändringar utfördes).

• **bool editABox(<parametrar för en låda>)** för att kunna editera en låda (ändra värde på en eller flera egenskaper). Funktionens returvärde ska indikera om editeringen lyckades eller om den inte gjorde det (lådan hittades inte / inga ändringar utfördes).

• **int nrOfShapes()** för att få antalet geometriska figurer.

• **int nrOfCones()** för att få antalet koner.

• **int nrOfBoxes()** för att få antalet lådor.

De medlemsfunktioner som har en array med strängar och capOfArray (kapaciteten på sträng arrayen) som parametrar samt bool som retur ska ”fylla i” arrayen med strängar.
Om capOfArray inte är tillräcklig utförs ingen ”ifyllnad” av strängar.
Funktionens returvärde ska visa om arrayen fylldes på eller ej.

Vidare ska du implementera konstruktor(er), destruktor, kopieringskonstruktor och tilldelningsoperator.

Ingen inmatning eller utmatning får förekomma i medlemsfunktionerna.

Gör en testfil TestShapeRegister.cpp i vilken kopieringskonstruktor, destruktor samt tilldelningsoperatorn för ShapeRegister testas på ett tillförlitligt sätt!
Tips: Gör gärna en motsvarighet till testprogrammet för Inlämningsuppgift A.

Denna fil, som också ska lämnas in, ska vara väl kommenterad avseende vad som testas när i koden.

Du ska **även göra ett textbaserat meny-styrt användargränssnitt (ShapeRegisterSystem.cpp)** där användaren via menyval kan arbeta med ett register för geometriska figurer enligt ovan.

Varje delproblem hanteras i en separat funktion (minst en funktion för respektive menyalternativ).

Du kan utgå från att inga orimliga värden matas in av användaren (ex-vis negativt värde på höjd osv).

Du paketerar Dina .h- och .cpp-filer i en .ZIP-fil och det är endast denna .ZIP-fil som lämnas in.