Inlämningsuppgift 3

Du ska implementera en **arvshierarki** samt en klass vilken hanterar objekt som är av klasstyper i den implementerade arvshierarkin.

I en glasögonbutik finns ett lager med **solglasögon** (SunGlasses) och **läsglasögon** (ReadingGlasses). Både solglasögon och läsglasögon har en båge som registreras med märke (ex-vis Borg) och ett grundpris (ex-vis 800 SEK).

För glasögon registreras dessutom styrka (ex-vis 2.5) och vilken typ av glas det är (ex-vis enkelslipade). Det finns två olika typer av glas: enkelslipade och dubbelslipade. Enkelslipade glas ingår i grundpriset men för dubbelslipade tillkommer en kostnad på 1500 SEK.

För solglasögon registreras i stället dessutom färgen på glaset (ex-vis brun). Det finns 3 olika färger: brun, gul och lila. Brun ingår i grundpriset medan lila och gul ger en tilläggskostnad på 200 SEK.

När det gäller solglasögon ska det vara möjligt att få färgen (funktionen getColor)

När det gäller läsglasögon ska det vara möjligt att få styrkan (funktionen getStrength)

Alla glasögon, dvs både solglasögon och läsglasögon, ska genom funktionen

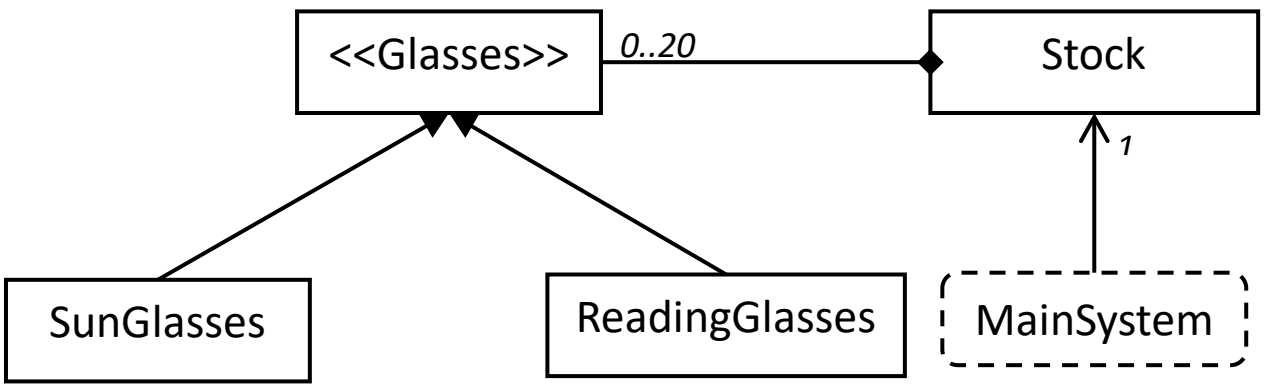
- **calculateFinalPrice(),** returnerar det slutliga priset för glasögonen, dvs summan av grundpriset och eventuell tilläggskostnad
- **description()**, returnerar en sträng som innehåller en beskrivning av glasögonen.
    - För läsglasögon ska strängen innehålla märke, styrka, typ av glas och slutligt pris
    - För solglasögon ska strängen innehålla märke, färg och slutligt pris

Arvsmekanismen användas logiskt och rimligt med avsikt att generalisera och specialisera

Basklassen ska vara **abstrakt**

Alla **medlemsvariabler ska vara privata.**

Dynamisk bindning ska möjliggöras på ett logiskt och rimligt sätt



Detaljerat klassdiagram för Stock

Stock
// only private member variables // for you to decide
+ add(make: string, basePrice:int, color: string): bool + add(make: string, basePrice:int, strength: float, typeOfGlass: string): bool + nrOfGlasses(): int + nrOfSunGlasses(color: string): int + nrOfReadingGlassesWithStrengthAbove (strength: float): int + totalStockValue(): int

Relationen mellan Stock och Glasses ska implementeras genom användande av en **statiskt allokerad array innehållande pekare av basklasstyp**

Därutöver ska du i klassen **Stock** implementera

- **Konstruktor** med tom parameterlista
- **Destruktor**
- **Kopieringskonstruktor** (copy constructor)
- **Tilldelningsoperator** (assignment operator)

Vidare behöver du tillföra funktionen nedan **för testningen:**

Glasses * getAccessToGlassesAt(int index) som returnerar adressen till de glasögon som finns på det givna index som parametern innehåller

Vid tillägg (add-funktionerna) placeras det nya objektet direkt efter redan befintliga.

Övriga medlemsfunktioners namn indikerar vad de ska uträtta.

Du har tillgång till testprogrammen **TestGlasses.cpp** för att testa viss funktionalitet för arvshierarkin och **StockTest.cpp** för att kontrollera funktionaliteten för Stock.

Du behöver dessutom exekvera testprogrammet i Debug-läge för att kontrollera om programmet genererar minnesläckor eller ej.