

DV1497: Ominlämningsuppgift B

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

Lösningen får inte generera några minnesläckor.

Tävlingsdeltagare

I denna del ska Du skapa en arvshierarki för hantering av deltagare i en tävling.

De tävlande kan vara antingen motionärer eller professionella.

När det gäller motionärer behöver man registrera namn, kön och ålder. När det gäller professionella deltagare registreras namn, kön, klubb och antal säsonger som professionell.

Implementera en arvshierarki för tävlingsdeltagare enligt ovan där minst följande ska vara uppfyllt:

- arvsmekanismen ska användas logiskt och rimligt i avsikt att generalisera respektive specialisera
- basklassen ska vara abstrakt
- polymorfism som möjliggör dynamisk bindning ska användas
- alla medlemsvariabler ska vara av en logiskt lämplig typ
- varje tävlingsdeltagare ska kunna returnera sitt "innehåll" som en sammansatt sträng.
- det ska vara möjligt att ändra samt erhålla värdet på tävlingsdeltagares egenskaper

Gör dessutom en klass för ett register över tävlingsdeltagare. I denna ska tillses att objekt som instansieras av denna klass ska innehålla en (1) dynamiskt allokerad array av pekare till basklasstyp. Det ska finnas en medlemsvariabel som håller antalet inlagda tävlingsdeltagare, men det ska inte finnas ytterligare medlemsvariabler för antalet professionella deltagare eller för antalet motionärer.

Det ska finnas medlemsfunktioner för att ombesörja följande:

- Lägga till en motionär
- Lägga till en professionell deltagare
- Fylla en array av typen string med innehåll från samtliga tävlingsdeltagare
- Erhålla antalet inlagda tävlingsdeltagare (för att kunna dimensionera arrayen ovan)
- Fylla en array av typen string med innehåll från endast professionella deltagare
- Erhålla antalet inlagda professionella deltagare (för att kunna dimensionera arrayen ovan)
- Fylla en array av typen string med innehåll från endast motionärer
- Erhålla antalet inlagda motionärer (för att kunna dimensionera arrayen ovan)
- Ta bort en tävlingsdeltagare givet namnet (namnet betraktas som unikt)
- Ändra antal säsonger som professionell för en given professionell deltagare
- Sortera alla tävlingsdeltagare baserat på namnet. Här används med fördel en <-operator i arvshierarkin.

Registret ska inte vara begränsat avseende det antal tävlande som kan finnas i registret.

Du ska dessutom implementera konstruktor(er), destruktor, copy-konstruktor och tilldelningsoperator.

Observera att ingen inmatning eller utmatning ska förekomma i medlemsfunktionerna.

Vidare gör du ett textbaserat meny-styrt gränssnitt där användaren via menyval kan arbeta med ett register för tävlande enligt ovan. Hantera separata delproblem i separata funktioner.

**Du ska även bifoga en testfil i vilken copy-konstruktor, destruktor samt tilldelningsoperatörn testas för tävlingsregistret på ett tillförlitligt sätt!
Denna fil ska vara välkommenterad avseende vad som testas när i koden.**