

Övningsuppgift v. 42 - Miniräknare

Grundläggande Programmering 2IS048

I denna uppgift förväntas du skapa en miniräknare med grundläggande funktionalitet. Denna miniräknare ska vara skapas som en s.k. konsol-applikation, dvs. samma typ av applikation som du har arbetat med hittills i kursen. För att lösa denna uppgift så behöver du applicera samtliga av de koncept som har behandlats i kursen. Gällande applikationen som sådan så förväntas miniräknaren kunna utföra följande:

- Läs input, i ett angivet format
- Skriva ut output, i ett angivet format
- Applikationen ska köras tills dess att användaren väljer att avsluta den med ett specifikt kommando

En lista över de faktiska funktionella kraven hittar du senare i dokumentet.

Notera att denna uppgift *inte* ska lämnas in. Uppgiften är enbart till för dig att öva på. Till ditt förfogande så har du tillgång till lektion- och föreläsningsmaterial via Studentportalen samt ytterligare övningar och beskrivningar på koncepten via Programmeringshubben.

1 Funktionella krav

- Miniräknaren ska stödja funktionalitet för addition, subtraktion, multiplikation samt division. Denna funktionalitet ska skrivas som egna metoder för resp. funktion.
- Miniräknaren ska fortsätta köras tills dess att användaren anger ett kommando för att avsluta applikationen. Dvs. att om en användare anger t.ex. "0" eller "Exit" så ska applikationen avslutas, annars ska den fortsätta be om nya operationer. För att avsluta en konsol-applikation så kan `Environment.Exit(0)`; nyttjas.
- Applikationen förväntas ha viss felhantering och ska inte "krascha" pga. felaktig användning. Dvs. att om en användare exempelvis anger

en bokstav istället för en siffra vid addition så ska programmet kunna hantera detta. Grundläggande matematiska regler ska även efterföljas.

- Miniräknaren ska kunna returnera decimaltal.
- Samtliga av dina egen-definierade metoder ska vara kommenterade enligt ett s.k. XML-format. En guide för hur man nyttjar och skriver denna typ av kommentarer finner du via följande länk:
<https://docs.microsoft.com/en-us/visualstudio/ide/reference/generate-xml-documentation-comments?view=vs-2019>

2 Extrauppgifter

Här finner du förslag på ytterligare funktionalitet som du kan lägga till i din miniräknare:

- Funktionalitet för att hantera historik i miniräknaren. Här handlar det om att vi vill spara ett fastställt antal operationer som har genomförts, t.ex. 5 eller 10 st. En operation avser här resultatet av en funktion samt informationen om vilka värden som tillsammans gav resultatet. T.ex. avser " $10 + 20 = 30$ " en operation.
- För att ovan ska fungera smärtfritt så måste du hantera det fall när vi har slut på plats i "minnet". Dvs. när vi exempelvis har lagrat 10 operationer och vill utföra ännu en operation. För att hantera detta fall så ska du implementera funktionalitet för att "börja om". Ett exempel på detta är att när du har slut på plats i "minnet" så flyttar vi alla operationer i listan, tar bort den äldsta operationen och lägger till den nya operationen i slutet. Hur du väljer att implementera detta är dock upp till dig.
- Om vi har implementerat minnesfunktionaliteten så känns det rimligt om vi kan nyttja denna, exempelvis genom att ge användaren möjligheten att lista historik. Detta bör göras enligt nedan format:

Operation 1, Operation 2, Operation 3

Notera att om det exempelvis bara finns 3 operationer utav 5 (eller 10 etc.) möjliga så är det enbart dessa som ska skrivas ut och att, för den sista operationen, så skrivs det inte ut något kommatecken.

- Reflektera kring mindre vidareutvecklingar som kan genomföras för att utöka miniräknarens funktionalitet. Exempelvis för att beräkna logaritmer, exponenter, modulus eller dylikt.