Laboration 4, Generics - Collections - GUI

Inledning

I den här laborationen ska du skapa ett GUI (Graphical User Interface) med WPF (Windows Presentation Foundation) och TFM (Target Framework Moniker) >= .NET 5.x. Du ska använda dig av Generics och Collections för att fylla listboxar i ditt GUI. Vid designen av programmet ska du försöka följa SOLID principerna.

Redovisning

Du ska skriftligt och muntligt redovisa laborationen. Den skriftliga redovisningen ska bestå av en laborationsrapport och källkodsfiler packade med zip.

Den muntliga redovisningen ska bestå av en inspelad video där du förklarar din lösning. Du kan använda valfritt verktyg (t.ex. OBS Studio) för inspelning och valfri plattform för att dela en länk (t.ex. via din skol-webbplats, OneDrive eller YouTube). Det centrala är att du kan dela en länk via Learn som är åtkomlig. Jag tar **inte** emot länkar via e-post.

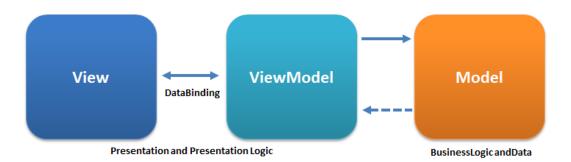
Uppgift 1

Redogör vad man använder generiska klasser och metoder till och beskriv vilka fördelar och nackdelar det kan medföra. Beskriv ett scenario där en programlösning skulle kunna dra nytta av att använda en generisk klass. Utgå från en egen vald problemställning och beskriv dina tankegångar.

Uppgift 2

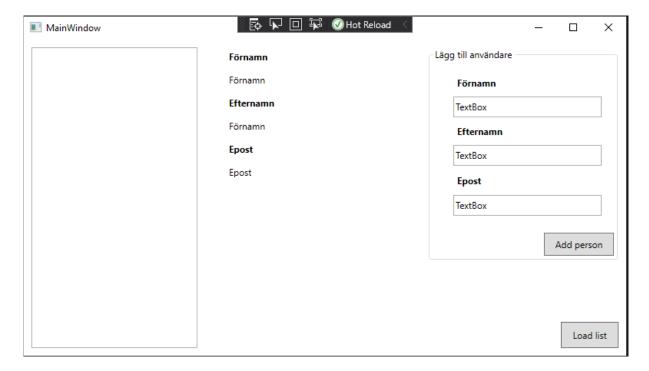
Du ska skapa ett GUI-program i Visual Studio enligt GUI-bilden på nästa sida.

För att förenkla utvecklingen bör programmet använda sig av Model–view–viewmodel (MVVM) mönstret: https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93viewmodel



Mer om MVVM finns att läsa här:

https://docs.microsoft.com/en-us/dotnet/api/system.collections.objectmodel.observablecollection-1 och via Data Binding Overview länken på den sidan.



Det som ska visas är en lista på dina klasskamrater inkluderat dig själv. Se bifogade filen: klasslista-21.txt för indata (textfilen har i nuläget ingen speciell formatering). Det ska finnas en knapp för att läsa in användare/studenter (Load list) i programmet via en Open File Dialog.

Inläsning av data ska ske async så att GUI:t inte "fryser", dvs. det ska inte sluta reagera på drag eller klick event etc. när arbete som kan ta längre tid än 50 ms inleds:

https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/async/using-asyncfor-file-access

Vid programmets start är listan till vänster tom. När användaren klickar på knappen för att fylla listan ska användare läsas in. När användaren väljer en person i listan (rad) ska personens information visas till höger.

Det ska gå att lägga till användare i listan via komponenterna till höger. Det viktigt att se till att användaren inte kan lägga till icke komplett data. Dvs. förnamn, efternamn och e-postadress måste vara ifyllt innan nya användare kan läggas till i listan.

Notera att den visuella Toolbox editorn mest ska användas som komponentreferens (enligt Tim Corey: https://www.youtube.com/watch?v=gSfMNjWNoX0 > 9.20) för att skapa programmets knappar och kontroller. Det säkraste är att hand-editera XAML koden för att skapa GUI.

Information om XAML finns via Microsofts dokumentation:

https://docs.microsoft.com/en-us/visualstudio/xaml-tools/ > XAML overview. Mer info om WPF, tutorials mm. finns i Learns kursrum.

Frivilliga fördjupningsuppgifter

Vidareutveckla ditt program. Exempel:

a) Lägg till fler egenskaper för användarna. Lägg till en radioknapp för att sortera användarna i listan efter förnamn, efternamn etc. Lägg till en spara knapp som kan spara sorterade listor till nya filer och så vidare.

Undersök andra kontroller du kan använda för att utöka eller förbättra funktionaliteten och användarvänligheten i programmet. Se till att programmet kan hantera GUI storleksförändringar.

- b) Implementera funktioner som att en befintlig användare kan raderas, editeras/uppdateras och att dubbletter av användare inte kan sparas.
- c) Dela upp programmet så att programlogiken ligger i ett library assembly/projekt och GUI koden ligger i ett annat assembly/projekt.
- d) Byt ut läsning/skrivning till fil till att använda en valfri databas (SQL/NoSQL). Även här gäller async access eftersom svarstiden är oförutsedd.

Avslutning

Redovisa dina tankar om generiska metoder och klasser i ett dokument och lämna in tillsammans med ditt projekt i uppgiftsmappen i Learn. Se rubriken redovisning för det muntliga momentet. **Deadline för sista inlämning är: 2021-10-15, kl 18:00.**

Lycka till!

/Hans