

Reflektioner och motiveringar, Inlämningsuppgift 1 Sanna Höglund WU23

När jag började med uppgiften så planerade jag först hur "skalet" skulle se ut med ett MainProgram och två klasser: Book och Library. Sedan planerade jag för att skapa en meny där användaren kunde få information och ge input i 4 switch-cases för enligt uppgiftens mål. Började skriva den koden för att sedan skriva metoderna i de olika klasserna. För att arbeta objektorienterat så gjorde jag menyns delar i metoder i libraryklassen som jag anropar från Program.cs i en Start-metod.

När jag hade gjort case 1 så hade jag deklarerat new book inne i case 1. När jag gick vidare till nästa case som var case 3(skriva ut alla böcker i listan upptäckte jag att när klassen book var deklarerad i case 1 kunde jag inte komma åt den i något annat case i min switch så flyttade den utanför switchen. Använde sedan en foreach loop för att kunna skriva ut alla böcker från listan inklusive deras egenskaper. Ofta är det såna "småsaker" som hänger upp sig för mig. Att jag har ett mål men inte riktigt vet hur jag ska göra så det kan vara väldigt tidsödande för mig och till slut har jag googlat så mycket o ändrat så mycket i koden så jag har fått börja om från början. Antar att det är något som blir enklare ju mer man kodar.

I case 2 ville jag kunna radera valfri bok från listan. Då valde jag att be användaren skriva in titel. För att sedan kunna söka igenom listan och kunna använda metoden Remove() för att ta bort den valda boken från listan.

Varför vissa metoder och klasser skapades

Jag skapade två klasser, en bok-klass där jag skapade propertiesen av klassen med get och set för att kunna ta emot och anpassa dem. Gjorde också en konstruktor så att varje bok som las till i klassen skulle innehålla alla förvalda element.

I library-klassen skapade jag dels metoder för menyn. PrintMenu(), MenuChoiches() och ClearConsole(); Det gjorde jag för att skapa en mer läsbar och tydlig kod samt att man ska skriva all kod i metoder när man jobbar på ett objektorienterat sätt.

Därför gjorde jag också en metod för att lägga till böcker AddBook();, radera böcker: RemoveBook(); och för att skriva ut alla böcker i biblioteket: ListBooks();.

SOLID-principerna

Upplever att det SOLID-principerna är mer avancerade än min kunskapsnivå av objektorienterad programmering än så länge men försöker beskriva hur jag tänkt nedan:

Det jag kan säga att jag använt är Single responsibility principle. Bok-klassen har endast ett ansvarsområde men som jag också skriver i min reflektion använde jag library-klassen både till att lägga till böcker och skriva ut menyn. Skulle kunna ha gjort en egen klass för Menyn men å andra sidan ska man enligt uppgiften ha två klasser och det har jag gjort.

Hur koden uppfyller principerna för Clean Code.

Genom att ha tydliga och förklarande variabel-namn och namn på mina metoder har jag försökt skriva kod som är lätt att förstå även för andra. Jag har även försökt att göra metoderna så korta som möjligt och gjort en metod för varje del i programmet. Försökte också jobba enligt Clean Code- koceptet genom att dela upp menyn i flera metoder och en ClearConsole(); metod som jag i sin tur använde efter att de andra metoderna körts för att få ett "snyggare" program. Därför har jag inte heller kommenterat jättemycket i själva metoderna utan tanken är att jag har kommenterat en sammanfattning ovanför varje metod som förklarar vad metoden gör. Har försökt tänka på formateringen/indenteringen för att koden ska vara läsbar och tydlig och använda mig av korrekt felhantering för att programmet inte ska krascha om användaren skriver in fel input(tex string istället för int).

Eventuella kompromisser eller beslut som togs för att balansera funktionalitet och läsbarhet.

En sak som jag kunde gjort som jag valde att hoppa över då det inte var ett skarpt krav i uppgiften, var att användaren själv skulle skriva i om boken var utlånad eller inte. Då hade jag gjort en till metod för att fråga användaren om boken fanns inne eller inte och med hjälp av det svaret ändra boolean isAvailable till true eller false och sen lägga till det till varje bok i klassen.

Skulle också kunna ha gjort en egen klass för meny-metoderna för att bättre jobba objektorienterat genom att varje klass ska ha ett eget och tydligt användningsområde. Nu använda jag ju library-klassen både till Menyn och till att hantera bok-klassen i. Har satsat på att nå upp i VG-nivå med koden och med ovanstående motiveringar.