

Labbrappo~~r~~t: Labb 4

Författare: Isak Andelius, Sofia Vadell, Lorenzo Giangulio

Datum: 2025-12-18

Kursnamn: GIK299 – Objektorienterad Programmering

Examinator: Elin Ekman och Ulrika Artursson Wissa

Innehåll

1.	Introduktion	3
2.	Metod.....	4
2.1.	Verktyg.....	4
2.2.	Stegvis beskrivning av tillvägagångssätt	4
2.3.	Förutsättningar för att göra labben	4
2.4.	Testning av koden.....	4
2.5.	Etiska överväganden.....	4
3.	Resultat	5
4.	Diskussion och reflektion	9
4.1.	Diskussion kring resultat	9
4.2.	Reflektion kring sprint 1	9
4.3.	Reflektion kring sprint 2	9
4.4.	Reflektion kring alternativa lösningar	10
	Frågor till AI-verktyg.....	10
	Referenser.....	11

1. Introduktion

Syftet med denna laboration var att utveckla ett konsolbaserat program i C# där användaren kan registrera och hantera information om personer. Programmet är uppbyggt med objektorienterade principer och använder klasser, listor och enum-typer för att strukturera data på ett tydligt sätt.

2. Metod

2.1. Verktyg

Vi använde Visual Studio 2022 tillsammans med .NET 6.0 som ramverk för att bygga och testa vår applikation. Kodexempel från lektion 4 (Artursson Wissa & Ekman, 2025) och en spellista på YouTube (Height Above Sea Level, 2022).

2.2. Stegvis beskrivning av tillvägagångssätt

Då vi startade labben precis efter en lektion då vi gick igenom utförandet av klasser, enum och structs (Artursson Wissa & Ekman, 2025) hade vi grunden rykande färskt i minnet och utgick därifrån. Det gav oss en bra struktur som vi därefter fortsatte arbeta utifrån och vi kunde snabbt lösa sprint 1.

Eftersom vi är en grupp på 3 har vi gjort det mesta i ett grupperum där vi kopplat upp en dator på stora skärmen och de två som inte skrivit kod har följt med live på skärmen. Vi bytte placeringar varje timme ungefär men alla har varit delaktiga hela tiden antingen genom att skriva eller att sitta bredvid och styra/diskutera/planera.

2.3. Förutsättningar för att göra labben

Vi jobbade på som vanligt i visual studio 2022.

2.4. Testning av koden

När vi var klara med labben och funkade när vi fyllde i allt ”rätt” började vi fylla i helt fel resultat (vi satte exempelvis in både en double och string där vi endast tänkt att man skulle fylla i en int) och insåg då att vi inte alls var klara. Vi hade många problem att lösa vid olika typer av fel i inmatning. Vi fortsatte testa på fel sätt tills allt funkade smidigt.

2.5. Etiska överväganden

Inga etiska överväganden var nödvändiga eftersom endast fejkade personuppgifter användes i testandet av denna labb. Vi lånade namnen på Elon Musks barn för testandet av input men inga andra uppgifter.

3. Resultat

Vi stötte på många problem och var tvungna att göra om och testa olika saker nästan hela tiden under sprint 2. Störst problem hade vi när vi försökte få indata att stämma i listan, samt när vi skulle ge användaren felmeddelanden och låta hen skriva om på nytt. Varje gång vi löste ett problem hade vi skapat oss ett nytt men genom kontinuerlig testning och felsökning lyckades vi successivt förbättra programmets funktionalitet. Ett specifikt problem rörde användningen av DateTime som vi kämpade med att få bort utskrift av tid löste vi tillslut med DateOnly som var ett enkelt sätt att bara ha datumen vilket passade just denna labb perfekt då vi endast behövde födelsedatum. Utöver de krav som angavs i beskrivningen för labben valde vi att lägga till namn på varje person, det ökade tydligheten för personerna i listan. Vi la också till bestämt antal bokstäver man kan fylla i, då ett namn exempelvis är mer än en bokstav.

Med hjälp av en person-klass hanterar vi allt som rör en person och lägger till i en lista. Personerna vi lägger till i listan har fått välja kön bland redan färdiga val som vi har i en enum. Och en struct för att hålla info om hår.

Resultatet visar att programmet fyller laborationens krav. Det är så likt en anka vi kan klara med dagens kunskaper.

Här kommer screenshots på när vi fyller i uppgifter, får ett felmeddelande samt ser innehållet i listan

Name: thecno mexanikus
Hair Color: brown
Hair Length: 10cm
Gender: Man
BirthDate: 1970-12-12
Eyecolor: brown

Name: X Æ A-Xii
Hair Color: blond
Hair Length: 5cm
Gender: Other
BirthDate: 2000-01-01
Eyecolor: Green'

Name: Exa Dark Sideræl
Hair Color: Grey
Hair Length: 10cm
Gender: Woman
BirthDate: 1939-12-24
Eyecolor: Grey

Name: Romulus
Hair Color: brown
Hair Length: 10cm
Gender: Man
BirthDate: 2012-01-31
Eyecolor: brown

```
Name:  
Exa Dark Sideræl  
Hair color:  
g5
```

```
-----  
Invalid Input! You can't leave it empty or use any numbers.  
-----
```

```
Hair color:  
Grey  
Hair length(CM) :  
d
```

```
-----  
Invalid input. Please enter a valid number for hair length:  
-----
```

```
Hair length(CM) :  
10  
Birth Date (YYYY-MM-DD):  
1924
```

```
-----  
Invalid input. Please enter a valid number for birth date:  
-----
```

-
- 1. Add a person to the list
 - 2. Write out the list
 - 3. End the program

1

Name:

Romulus

Hair color:

brown

Hair length(CM) :

10

Birth Date (YYYY-MM-DD):

2012-01-31

Eye Color:

brown

Gender, choose a number:

- 1. Woman
- 2. Man
- 3. NonBinary
- 4. Other

2

Person added.

4. Diskussion och reflektion

Generellt sett har vi utvecklats i att förstå teorin bakom klasser, metoder och struct/enum som ligger utanför Main-programmet kan användas och anropas och hur vi kan använda data smidigt för tydligare och mer strukturerad kod och "snyggare". Vi har också fått testa på att arbeta då en person skrivit kod medan de andra suttit bredvid (ofta via stor skärm i grupprum) och hängt med samt styrt och hjälpt. Man lär sig ibland mer av att få ut sina tankar om någon annan skriver ner det. Vi hade inget flödesschema med det hade förmodligen underlättat speciellt när det gäller felhantering. Som vi definitivt fått jobba hårt kring i denna labb.

4.1. Diskussion kring resultat

Vi fick resultat som stämmer överens med kravlista och så mycket mer än det kan vi inte göra nu. Ju mer vi lär oss destå mer nya egna idéer kommer vi kunna få och kunna implementera för framtida projekt. Mycket av det vi gjort har vi lärt oss på lektioner innan men det som saknades var klurigt att hitta och lösa själva, mycket om just felhanteringen av input och hur mycket med kod det kräver för att lösa så man både får felmeddelande och hanteringen. Den stora utmaningen låg i felhanteringen samt att få till parametrar och in- och utdata. Att arbeta i grupprum på stor skärm fungerar väldigt bra för oss speciellt som tre personer och låta alla kunna vara delaktiga.

4.2. Reflektion kring sprint 1

Vi hjälptes mycket av lektion 4 om klasser, enums och structs (Artursson Wissa & Ekman, 2025) som vi hade precis innan. Det underlättade mycket för oss mycket med kod och upplägg samt förståelse för vad vi skulle göra och hur. När vi kodade delade vi till stor skärm i ett grupprum så alla kunde se samtidigt, och ifall något blev fel.

4.3. Reflektion kring sprint 2

I sprint 2 krävdes mer av alla i gruppen. Vi kunde inte längre gå helt efter material rakt av. Vi tittade mycket på alla tidigare lektioner och labbar och försökte slå ihop all info. Kanske var vi också inställda på att det skulle gå lika smidigt men vi stötte på väldigt mycket problem istället, som blev problemlösning och vi alla utvecklades mycket i sökandet efter rätt lösning.

4.4. Reflektion kring alternativa lösningar

Med mer tid kanske vi hade kunnat fundera ut både bättre och alternativa lösningar likaså att vi för varje lektion och programmeringsstuga blir bättre så blir vår nybörjarkod väldigt snabbt uppdaterad och man inser att man kunnat göra koden på andra sätt. Vi hade eventuellt velat ha tydligare felmeddelanden och större förståelse för hur vår data i olika klass/struct/enum "samarbetar". Vi hade också velat göra de frivilliga uppgifterna då vi hade velat lära oss inte bara hur man inte bara för in data och skriver ut utan även raderar det.

Frågor till AI-verktyg

Verktyg: Gemini

Vi fastnade rejält då vår Gender-enum inte vill fungera alls. Efter många försök kopierade vi in texten och frågade varför, vi saknade då numrering i enumen (vi valde numrering som alternativ när man fick välja gender) och efter vi testat lägga i matchande siffror i enumen efter numreringen vi använde fungerade det mycket riktigt. Vi forskade vidare i varför detta funkade och fick lära oss att alla enums har siffror i bakgrunden och vi hade inte behövt skriva ut det om vi valt att parsea istället. Så nu lärde vi oss det, en misslyckande av att omvandla userinput fick oss att lösa problemet på annat sätt.

Referenser

Ekman, E., & Artursson Wissa, U. (2025). *GIK299 OOP Lektion 4 HT25*. [powerpointpresentation].
Canvas: <https://canvas.du.se/>

OpenAI. (2023). *ChatGPT* (Version GPT-3.5) [Stor språkmodell]. <https://chat.openai.com/chat>

Google. (2025). *Gemini* [Stor språkmodell]. <https://gemini.google.com>

Height Above Sea Level. (10 maj 2022). *Dates and Times in C# and How to Use Them*. [Video]. YouTube.
<https://www.youtube.com/watch?v=0B6u71YKwdc&list=PLKCFItnebRSVQHQ9tf5PSmk4X0KNqO6FU&index=15>