Labbrapport: Labb fyra

Författare: Samir Ahmad, Ludwig Lindfors

Datum: 2025-01-03

Kursnamn: GIK299 – Objektorienterad Programmering

Examinator: Elin Ekman och Ulrika Artursson Wissa

Innehåll

1. Introduktion

I denna labb har vi genomfört programmeringsuppgiften for labb fyra. Den går på en del av en större insats för att hjälpa företaget Angelic Good Guys Inc. i kampen mot Evil Eavesdrop Enterprises (EEE) ett företag som misstänks samla in persondata på olagliga grunder som betraktas mot GDPR lagen.

Målet med labben var att utveckla ett system som kan efterlikna och samla bevis för att det är möjligt att samla in data på det sätt som EEE misstänks göra. Detta för att stärka bevisen mot deras olagliga data insamling. För att uppnå detta har vi använt oss av C# och dess objektorienterade principer med olika datatyper som enum, struct och class för att organisera och representera information om en person på ett strukturerat sätt.

Denna labb gav oss en förståelse för hur man jobbar Agilt, refererar bättre enligt APA och kan använda tidigare koder på ett professionellt och etik sätt. Labben gav oss också möjlighet att fördjupa oss i hur olika objektorienterad principer kan användas för att lösa konkreta problem och felhantera koder.

2. Metod

2.1. Verktyg

I denna labb har vi använt nedan resurser och referenser:

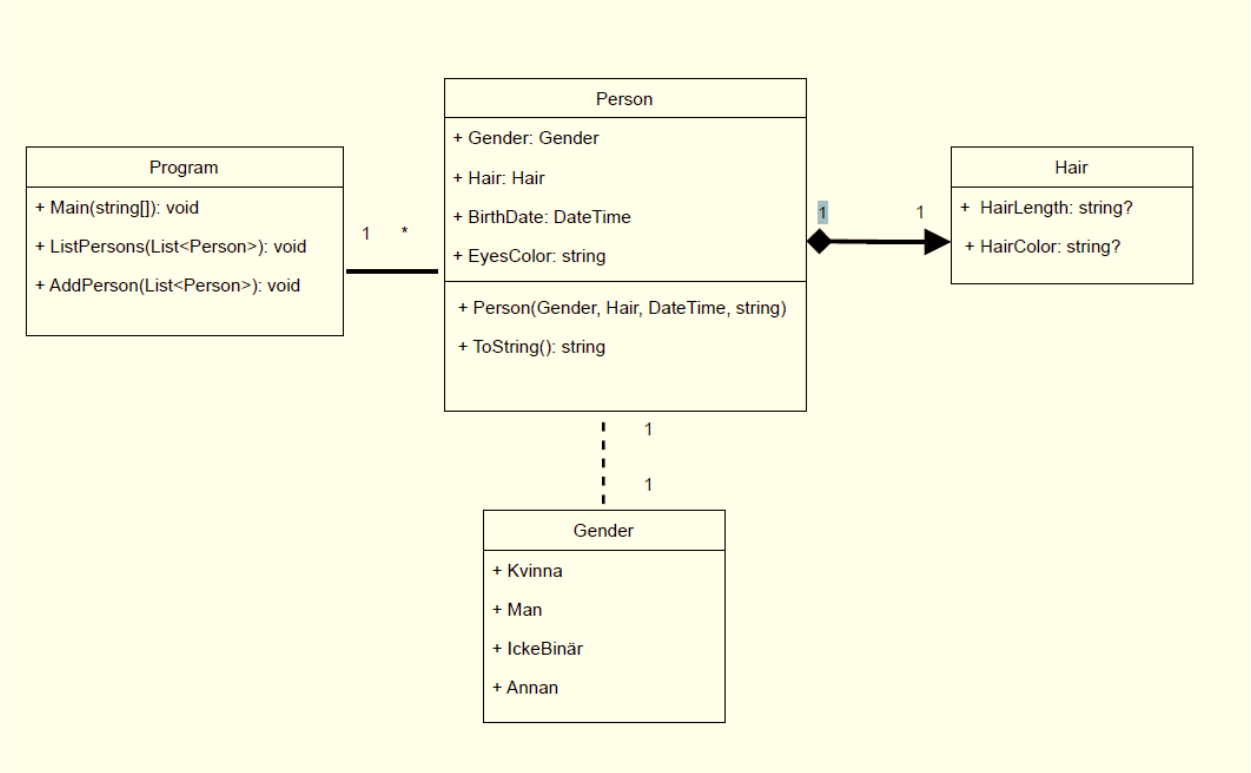
- Visual studio 2022, version 17.0, tillsammans med ramverket .NET 6.0 användes för att bygga och testa vår applikation.
- Kodexempel: Ahmad, S., & Lindfors, L. (2024). *WeatherSenseApplication console application in C#* (Lab 3, students of the Department of Computer Science at Dalarna University.).

2.2. Stegvis beskrivning av tillvägagångssätt

För att lösa denna labbuppgift började vi med att noggrant läsa instruktionerna och referensmaterialet i uppgiften. Det tog oss några timmar att förstå uppgiften och dess krav. Vi skrev först pseudokod och visualiserade koden med ett flödesschema och klassdiagram för att utveckla klassar, loopar och switch-satser. För att validera användarens input och effektivisera arbetet använde vi delar av koden från labb tre. Eftersom datastrukturen var central för uppgiften lade vi stor vikt vid att utforma den noggrant.

1- Vi delade upp arbetet men genomförde det i olika steg tillsammans:
2- Vi skapade ett enum för kön.
3- Vi definierade en struct för hår.
4- Vi designade en klass för en person och en string-metod.
5- Vi instansierade ett person-objekt i Main-metoden.
6- Vi skapade en lista för att lagra personuppgifter.
7- Vi implementerade en meny med hjälp av switch och while-loop.
8- Vi utvecklade en AddPerson-metod, som tog mycket tid på grund av validering, koppling mellan enum, struct och klass samt krav på noggrannhet.
9- Vi skapade en ListPerson-metod för att skriva ut samtliga personers uppgifter.
10- I varje steg tog vi tid och testade programmet för att snabbt lösa felet.
11- Därefter testade vi programmet enskild och ren skrev kommentererna.

Klassdiagram



2.3. Förutsättningar för att göra labben

För att genomföra labben behövde vi ha kunskap inom objektorienterad programmering inklusive begrepp som klasser, objekt och datastrukturer. Vi använde en dator och Visual Studio 2022 för att skriva och köra koden. För att köra koden behövde vi också ha C# programmeringsspråk installerat. Vi använde vissa kodstycken från vår labb 3 för att effektivisera utvecklingen och valideringen av användarens input.

2.4. Testning av koden

Vi testade koden flera gångar under utveckling och även efter utveckling. Vi matade in olika typer av data inklusive både giltig och ogiltig input. Vi fokuserade på att testa validering av användarens inmatning som att skriva in siffror och olika tecken där bokstäver förväntades och hantering av felaktiga värden. I födelsedag matade vi in framtidsdatum och hanterade felet och villkorade det så att den inte tar födelsedag mindre än 1899.

2.5. Etiska överväganden

Inga etiska överväganden var nödvändiga för denna laboration eftersom ingen verkliga persondata eller känslig information samlades in eller behandlades.

3. Resultat

Vi lyckades implementera funktionaliteten enligt kraven i labbuppgiften. Testmetoder som validering av användarinmatning och användning av debuggningsverktyg hjälpte oss att identifiera och lösa flera problem. Till exempel upptäckte vi att vår AddPerson-metod accepterade födelsedatum för framtid samt år före 1900. Vi tyckte att de inte var praktiskt. Vi införde if-villkor för att validera användarens inmatning. Vid inmatning av hårfärg och längd kunde användaren även mata in siffror vilket vi rättade med liknande valideringar.

En intressant observation var att när vi försökte skriva ut sparade personlistan visades projektets namn tillsammans med en punkt och klassnamn vilket såg oväntat ut.

Men vi tittade igen labb fyras uppvärmningsuppgifter för att hitta en lösning på problemet. Där insåg vi att metoden behövde en override för att anpassa hur objekten representerades som strängar. Efter att vi implementerat override löstes problemet och listan visade korrekt innehåll. Detta gav oss en bättre förståelse för användningen av override.

```
1 reference
public override string ToString()

{
    return $"Kön: {Gender}, Hår: {Hair.HairLength}, {Hair.HairColor}, Födelsedatum: {BirthDate.ToShortDateString()},
}
```

4. Diskussion och reflektion

Denna uppgift gav oss värdefulla insikter både tekniskt och praktiskt. Vi lärde oss att strukturera en lösning med hjälp av objektorienterad programmering och hur viktiga datastrukturer är för att organisera information effektivt.

Eftersom vi har grundläggande kunskapar inom objektorienterad programmering i C# hade vi lite svårt initialt med kopplingen mellan enum och strukturer och klass. Detta ledde till omskrivningar i koden och tog mycket tid. För framtida projekt kan vi förbättra planeringen och lägga större vikt vid pseudokod och flödesscheman och spara dem efter vår kodning som dokumentation.

4.1. Diskussion kring resultat

Vi uppnådde det resultat vi förväntade oss men utvecklingen tog längre tid än vad vi planerade. Vi var skeptiska till att återanvända kod från tidigare laborationer men insåg att detta var ett misstag. Om vi kopierade och anpassade tidigare valideringskod hade vi kunnat spara tid och minska arbetet. Detta är något vi kommer att ha i åtanke vid framtida projekt för att effektivisera vår arbetsprocess.

4.2. Reflektion kring sprint 1

Den första sprinten var mycket enklare att utveckla än den andra. Efter att vi avslutat den första sprinten startade vi dock inte direkt med den andra sprinten. Denna paus i vårt arbete resulterade till att vi delvis glömde bort delar av koden vilket ledde till extra tid för att läsa och förstå vår egen kod igen. Lärdomen från detta är att det är gynnsamt att undvika långa pauser mellan sprintar och att dokumentera koden mer noggrant för att underlätta återgången till projektet vid behov.

4.3. Reflektion kring sprint 2

Sprint två var svårare och krävde mer noggrannhet eftersom vi behövde kombinera struct, enum och klass. Det tog mycket tid att läsa om och titta på resurser online för att förstå hur dessa datastrukturer skulle användas tillsammans. Vi lärde oss mycket om datastrukturer som är värdefullt. Att skapa metoden för att skriva ut listan var också en utmaning då vi fick inte förväntat resultat trots vår kod enligt oss var ok men vi löste det genom att använda override i ToString() . Vår test fas var en viktig del i detta projekt då vi ville säkerställa att programmet inte kraschar och inte tillåter felaktig användarinmatning. En viktig lärdom var att ha bra kodkvalitet syn från början för att undvika problem vid testning.

4.4. Reflektion kring alternativa lösningar

Vi ville gärna utveckla detta projekt ännu mer avancerat till exempel i Window Form APP men vi hade inte tillräckligt kunskap i det.

Frågor till AI-verktyg

Verktyg: ChatGPT

Fråga/prompt: Skillnad mellan enum, struct och klass?

På vilket sätt svaret användes: svaret hjälpte oss att förstå att enum är ett utmärkt val för att representera konstanta värden eftersom det ger bättre struktur och läsbarhet jämfört med att använda en klass med attribut.

Verktyg: ChatGPT

Fråga/prompt: Varför ska man använda override i en method?

På vilket sätt svaret användes: Vi förstod att metoden ToString() är virtuell i bas-klassen och att det är viktigt att överrida den i underklasser annars kommer objektet inte att ge korrekt utskrift vid körning.

Verktyg: ChatGPT

Fråga/prompt: Vad ska vi tänka vid testning av kod?

På vilket sätt svaret användes: Vi fick tillgång till mycket information och valde att filtrera bort det som inte var relevant eller som var överkurs, men ändå vi läste mycket om testning. Svaret hjälpte oss att koda effektiv och hantera användarens felinmatning i god tid.

Verktyg: ChatGPT

Fråga/prompt: Vad är symbolen? i C# ?

På vilket sätt svaret användes: Svaret hjälpte oss att förstå begreppet nullable types i C# och hur vi kan hantera dem. Vi använde ”?” i vår switch-meny i koden string? choice = Console.ReadLine();. Därefter försvann erorr som tidigare uppstod trots programmet kunde hantera null ett på annat sätt. Vi lärde oss att ”?” infördes i C# 8.0 för att explicit markera att en referenstyp (som en string) kan vara null. För att fördjupa oss ställde vi samma fråga till Google och fann mycket användbar information på Stack Overflow och Raddit som vi har inkluderat i referenslistan.

Referenser

- Troelsen, A., & Japikse, P. (2022). Pro C# 10 with .NET 6 : foundational principles and practices in programming (11 uppl.). New York: APress. ISBN: 9781484278680.
- Vision. (n.d.). *Guide: Så leder du agilt*. Vision. https://vision.se/chef/amnen/agilt-ledarskap/guide-sa-leder-du-agilt/
- Panjuta, D. (n.d.). *Complete C# masterclass: Master C# programming from A to Z. Dive deep into .NET, OOP, Clean Code, LINQ, WPF, Generics, Unit Testing, and more*. Udemy. https://www.udemy.com/course/complete-csharp-masterclass/?srsltid=AfmBOooEShZoQHB-VlqQzIskJ_I6ABITjVHKFpL30BFwis4JMN4R5T3O&couponCode=KEEPLEARNING
- Ahmad, S., & Lindfors, L. (2024). *WeatherSenseApplication console application in C#* (Lab 3, students of the Department of Computer Science at Dalarna University).
- Caney, J. (2020, April 10). Why do we get possible dereference null reference warning, when null reference does not seem to be possible? *Stack Overflow*. https://stackoverflow.com/questions/59524568/why-do-we-get-possible-dereference-null-reference-warning-when-null-reference-d
- Wazupdanger. (2023, December 25). *Forgive me ……* Reddit. https://www.reddit.com/r/csharp/comments/1682yux/forgive_me_if_im_like_new_or_something_but_for