

Øvelse 1: Terminologi

I skal reflektere over begreberne "applikation", "instruktion", "sætning (en: statement)" og "kodeblok", med udgangspunkt i følgende beskrivelse:

1. Hvert enkelt medlem i gruppen skriver sine overvejelser om begreberne ned (2 minutter).
2. Hver gruppe starter **Ordet rundt**, idet det første gruppemedlem forklarer sine overvejelser (2 minutter). Hele gruppen diskuterer bidraget (2 minutter). Hvis gruppen enes om, at det er en god forklaring, skriver det første gruppemedlem sit bidrag ind i gruppens dokument.
3. Det næste gruppemedlem forklarer sine overvejelser osv. Fortsæt med **Ordet rundt** indtil alle gruppemedlemmer har bidraget, eller tiden er gået.

Tidsramme: 30 minutter

Note: Hvis du **ikke har fået installeret** Visual Studio IDE på din computer endnu:

- Benyt <https://dotnetfiddle.net/> i de følgende øvelser 3.2, 3.3 og 3.4 (spring 3.1 over)
- Betragt øvelse 3.5 (debug) som forberedelse til næste gang

Øvelse 2.1 Oprettelse af projekt i Visual Studio

Om lidt skal du til at lave din første applikation, men før du kan komme i gang med at skrive kode, skal du vælge en passende skabelon inde i Visual Studio, når du opretter projektet. På nuværende tidspunkt (og den kommende tid) skal du ikke fokusere på det visuelle aspekt, du skal først have styr på det helt basale. Derfor skal dit første C#-projekt oprettes som en **konsolapplikation**.

Udfør følgende:

- Start Visual Studio
- Opret en ny konsolapplikation med skabelonen: "Console App" (C#). Vælg selv projektnavn.
- Kør den autogenererede kode i applikationen, så du sikrer dig, at alt fungerer fint.

Du skal i det følgende tilrette denne kode.

Øvelse 2.2: Din første C#-applikation

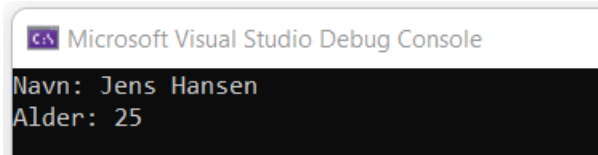
Du skal sammen med din sidemakker udvikle en applikation, der kan skrive navn og alder ud til konsolvinduet. Du skal øve dig i et skrive flere C#-sætninger (en: statements), der udføres sekventielt (dvs. i rækkefølge) efter hinanden.

Bemærk: Skriv i denne øvelse udelukkende sætninger inde i kodeblokken for **Main()** dvs. mellem de to tuborgklammer { } og ingen andre steder.

Udfør følgende (læs nedenstående omhyggeligt):

- Fjern den autogenererede linje, der skriver "Hello, World" ud til konsolvinduet.
- Tilføj en ny linje, og skriv en sætning (en: statement) med teksten "Navn: " ud til konsolvinduet (husk mellemrummet efter kolon)
- I en ny sætning skriv derefter på samme linje et navn ud til konsolvinduet, f.eks. "Jens Hansen"
- På en ny linje, skriv teksten "Alder: " ud til konsolvinduet
- Igen på samme linje skriv en alder ud, f.eks. "25"

Du skulle nu gerne have 4 sætninger fordelt på fire linjer i koden, men når du kører programmet, skal du gerne kun se følgende 2 linjer øverst i konsolvinduet:



```
Microsoft Visual Studio Debug Console
Navn: Jens Hansen
Alder: 25
```

Hvis det ser anderledes ud, så ret i de 4 sætninger, så det passer. Passer det? Tillykke! Du har nu præsenteret tekst for brugeren (dig selv) i din første C#-applikation.

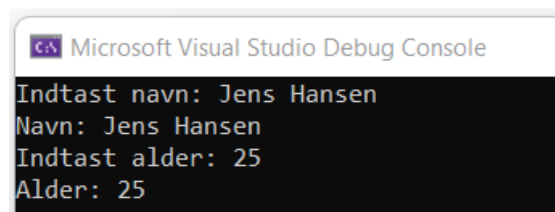
Øvelse 2.3: Brugerinput

Det næste trin er at give din applikation mulighed for at modtage tekst fra brugeren, dvs. modtage brugerinput. I stedet for at angive navn og alder i selve koden (dvs. "Jens Hansen" og "25"), da skal du indlæse disse to oplysninger direkte fra konsolvinduet.

Tilret programmet fra øvelse 2.2:

- Så den første sætning skriver teksten "Indtast navn: " ud til konsolvinduet
- Skriv dernæst "Navn: " efterfulgt af den indlæste tekst fra brugeren ud til konsolvinduet
Vink: benyt ReadLine() sammen med WriteLine()
- Skriv teksten "Indtast alder: " ud på en ny linje i konsolvinduet
- Skriv dernæst "Alder: " efterfulgt af den indlæste tekst fra brugeren ud til konsolvinduet

Du skulle igen ende ud med 4 sætninger i koden, men når du kører applikationen, burde du nu gerne kunne se følgende 4 linjer:



```
Microsoft Visual Studio Debug Console
Indtast navn: Jens Hansen
Navn: Jens Hansen
Indtast alder: 25
Alder: 25
```

Øvelse 2.4: Navn og alder med variabler

Du skal nu til sidst i konsolvinduet (på den 5. linje) skrive en tekst med formuleringen:

- "Jens Hansen er 25 år gammel"

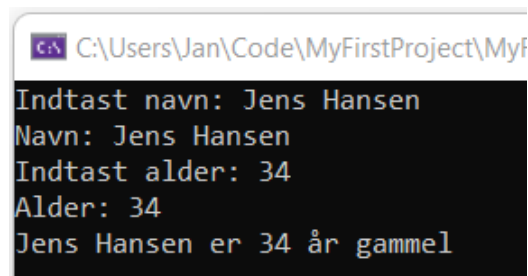
Dvs. du har brug for at sammenkæde (en: concatenate) en tekststreng med bidrag fra begge indlæsninger (navn og alder) plus noget ekstra tekst ("er" og "år gammel").¹

Fremgangsmåden i forrige øvelse med brug af ReadLine() er problematisk, da du ikke gemmer den indtastede information. Derfor skal du nu tilrette koden.

Udfør følgende (ved at tilrette koden fra forrige øvelse de rette steder):

- Erklær en variabel af typen string med navnet 'name', der kan indeholde et navn
- Erklær en variabel af typen string med navnet 'age', der kan indeholde en alder
- Tildel værdier til de to variabler via passende ReadLine()-kald
- Udskriv til konsolvinduet (via reference til de to variable 'name' og 'age') formuleringen:
"Jens Hansen er 34 år gammel", hvis det indtastede navn er "Jens Hansen", og tilsvarende alder er "34". Husk mellemrum.

Du skulle gerne ende med at se følgende i konsolvinduet:



```
C:\Users\Jan\Code\MyFirstProject\Myf
Indtast navn: Jens Hansen
Navn: Jens Hansen
Indtast alder: 34
Alder: 34
Jens Hansen er 34 år gammel
```

Øvelse 2.5: Debugging

I denne øvelse skal du for første gang afprøve debuggeren i Visual Studio på den kode, du lige har skrevet. Debugging er en proces, hvor du trinvis kan inspicere din kode, mens koden kører, og du kan lokalisere uhensigtsmæssigheder (nogle gange også fejl) i din kode, med henblik på at udbedre disse uhensigtsmæssigheder.

Udfør følgende:

- Benyt debug-kommando 'Step Over' (eller funktionstasten F10) til trinvis at udføre din kode fra øvelse 3.4. Inspicér konsolvinduet efter hvert trin, så du kan se, hvad der er skrevet ud på et vilkårligt tidspunkt.

Hint: Det kan være en god idé, at se om der er noget der kan benyttes fra forberedelsen.

Øvelse 3: Review spørgsmål

Du og dit team skal gennemgå følgende spørgsmål:

- Hvordan angiver man en tekststrengsværdi?
- Hvordan udskriver man noget til konsolvinduet?
- Hvordan modtager man brugerinput fra konsolvinduet?
- Hvad er en variabel?
- Hvad vil det sige at debugge sin kode?

Tidsramme: 15 minutter

I denne opgave fortsætter du med anvendelse af data i din kode via variable, hvor du introduceres til to nye simple (og centrale) datatyper, nemlig `int` og `bool`, nu også med lidt mere fokus på, hvordan du repræsenterer data, hvad en datatype er, og hvordan du håndterer data. Du introduceres også til en ny sætningstype (`if-else`) til at styre programflowet, dvs. styre hvilke sætninger der udføres (og ikke udføres); denne sætningstype betegnes en forgrening (en: conditional).

Til sidst snuser du ganske kort til begrebet 'konceptuel klasse'. **Sørg for at have lavet øvelse 3, 4 og 6 i denne opgave, da du skal arbejde videre med dem i næste opgave Ex03-CSharpData.**

Øvelse 4:

Del teamet i to grupper.

Benyt **Ordet rundt** i hver gruppe, og kig på følgende læringsudbyttebeskrivelser en for en. Sørg for, at alle bliver hørt i gruppen. og din gruppe mener er rigtig:

- Du kan nævne enkelte kendetegn ved C# programmeringssproget
- Du forstår løkker og kan implementere løkker i flere programmeringssprog
- Du forstår løkker og kan vælge den optimale blandt flere løkketyper

Tidsramme: 15 minutter

Øvelse 5: Programmering

I denne øvelse (og de resterende øvelser) benytter du dig af **parprogrammering** (forklaret i forberedelsen) sammen med en partner i teamet. Hvis I er et ulige antal til stede i dit team, skal I være tre i en gruppe.

Så find dig en partner, sæt dig ved siden af ham/hende ved en computer, og find ud af, hvem der skal skrive først, dvs. den der 'kører' eller 'føreren' (en: driver), og hvem skal observere eller navigere (en: navigator/observer). Husk at skifte rolle en gang imellem, så det ikke er den samme person, der programmerer under hele øvelsen.

Øvelse 5.1: Terminologi

Del teamet i to grupper, og reflektere over datatyperne "int" og "bool" samt begrebet "udtryk (en: expression)".

Tidsramme: 10 minutter

Øvelse 5.2: Genopfrisk 'navn og alder' med variable

Du skal sammen med din sidemakker kode en applikation, der skriver navn og alder ud til konsolvinduet. Mon du har set det før (genbrug er tilladt!) Denne gang skal du inddrage **både** string **og** int datatyper til de variable, hvor du midlertidigt gemmer og læser data.

Udfør følgende:

- Opret en ny konsolapplikation, igen med skabelonen: "Console App" (C#). Vælg selv et projektnavn.
- Erklær en variabel med navnet 'name', der kan indeholde et navn; **vælg en datatype til tekst**
- Erklær en variabel med navnet 'age', der kan indeholde en alder; **vælg en datatype til heltal (ikke string!)**
- Tildel literale værdier til de to variabler, f.eks. "Jens" og 34.
- Udskriv til konsolvinduet (via reference til de to variable 'name' og 'age') formuleringen: "Jens er 34 år gammel", hvis navnet er Jens, og alderen er 34

Tidsramme: 15 minutter

Øvelse 5.3: Brugerinput

I stedet for at tildele literale værdier for navn og alder til variabler i selve koden, da udvid jeres kode til at acceptere brugerinput, både tekst og tal.

Udfør følgende:

- Tilret koden til at indlæse navn og alder direkte fra konsolvinduet (genbrug fra tidligere opgave er tilladt)
 - Overvej, om der er noget fra forberedelsesmaterialet, der kan hjælpe dig med dette, specielt konvertering fra tekst til heltal.
- Inden du skriver navn og alder ud til konsolvinduet, træk da 10 fra alderen

Tidsramme: 30 minutter

Øvelse 6: Forgreninger

Et centralt element i alle programmeringssprog er styring af programflowet, dvs. at styre rækkefølgen af sætninger eller kodeblokke, der bliver udført og (lige så vigtigt) ikke udført.

Programmeringssproget C# (og stort set alle programmeringssprog) kan naturligvis bruges til at implementere algoritmer, hvis algoritmen er beskrevet tilstrækkelig detaljeret. Algoritmer har trin i CT, og i C# repræsenteres trin af sætninger. Du har allerede set eksempler på enkelte sætninger.

Øvelse 6.1: Terminologi

Del teamet i to grupper, og reflektere over begreberne "betingelse", "booleske udtryk" og "forgrening".

1. Reflektér over begreberne individuelt (2 min.)
2. Hvert gruppemedlem præsenterer dernæst resultatet af deres refleksion (2 min.)
 - Gruppen diskuterer det præsenterede (2 min.)
 - Bliv ved, indtil alle gruppemedlemmer er blevet hørt

Tidsramme: 10 minutter

Øvelse 6.2: Videre med navn og alder

I denne øvelse skal du arbejde med at styre valg gennem betingelser udtrykt via booleske udtryk, dvs. arbejde med at koble flere sætninger (CT: trin) sammen i kodeblokke (CT: sekvenser) og lade betingelser styre, hvilke sætninger der skal udføres. Konkret skal du anvende forgreninger via sætningstypen if-else i C#.

Inden du går i gang, da overvej først, hvad programmet skal gøre (dekomponér). Med den viden du allerede har nu, overvej, hvilke data og datatyper der skal bruges, samt hvilken logik (betingelser og forgreninger) der er nødvendig til styring af programflowet.

Udfør da følgende:

- Opret et nyt konsolprojekt i Visual Studio

Skriv dernæst kode i Main()-metoden med følgende funktionalitet:

- Brugeren skal kunne indtaste sit navn og alder via konsolvinduet og gemme oplysningerne i henholdsvis en string og int variabel (husk konvertering). Genbrug kode fra tidligere øvelse.
- De indtastede oplysninger udskrives til konsolvinduet sammen med en angivelse, om man er et barn, teenager etc. ud fra den indtastede alder med følgende forslag til en aldersinddeling og -beskrivelse:
 - 0-12: "et barn"
 - 13-19: "en teenager"
 - 20-25: "en studerende"
 - 26-67: "i arbejde"
 - > 67: "en pensionist"

Beskeden kunne have følgende fulde format:

"Jens er 34 år gammel og er i arbejde" eller
"Anders er 72 år gammel og er en pensionist"

- Implementér programmet, og test det med forskellige navne og aldre

Øvelse 7: Menu

Du har på nuværende tidspunkt byggestenene til at implementere en simpel menu-struktur og styre resultatet af menuvalget.

Øvelse 7.1: Styring af menuvalg

Først benyt de funktioner, som du allerede kender nu til at skrive følgende menu ud på konsolvinduet:

```
Min fantastiske menu:

1. Gør dit
2. Gør dat
3. Gør noget
4. Få svaret på livet, universet og alting

(Tryk menupunkt 1, 2, 3 eller 4)
```

Dernæst få styr på logikken, dvs. styring af menuvalget og udskrivning af resultatet:

- Afhængig af, hvad brugeren har trykket, da skriv følgende ud på konsolvinduet:
 - Hvis '1', skriv "Punkt 1 er valgt: Gør dit"
 - Hvis '2', skriv "Punkt 2 er valgt: Gør dat"
 - Hvis '3', skriv "Punkt 3 er valgt: Gør noget"
 - Hvis '4', skriv "Punkt 4 er valgt: 42"
 - Ellers skriv "Forkert valg"
- Implementér og test programmet

Øvelse 7.2: Tilpasning

Inspicér din kode, og overvej om, der noget der kan gøres nemmere, smartere eller mere kompakt.

Skriv følgende to linjer ind sidst i din kode, og tilret den foregående kode, så det passer med nedenstående:

```
Console.WriteLine("Punkt {0} er valgt: {1}", choice, message);
Console.ReadLine();
```

Øvelse 8: Review spørgsmål

- Hvad er en forgrening (en: conditional)?
- Hvad kan man benytte datatypen bool til?
- Hvad er et virkefelt (en: scope)?
- Hvilke roller er der i parprogrammering, og hvilke ansvar har de?
- Hvordan konverterer man en string til en int?
- Hvilke typer af operators findes der?