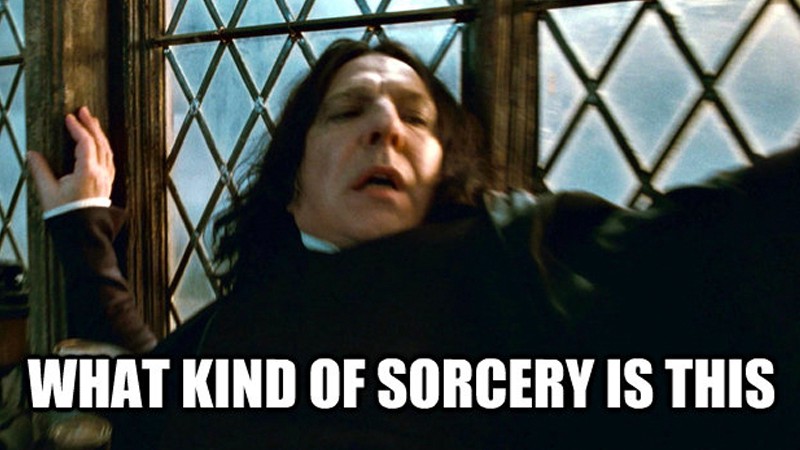
# Extension Methods

|  |  |
| --- | --- |
| **Læringsmål** | Du kan:   * **1Pf1**: anvende centrale metoder til at specificere og konstruere algoritmer […] * **1Pf2**: anvende centrale faciliteter i programmeringssproget til realisering af algoritmer, designmønstre, abstrakte datatyper, datastrukturer, designmodeller og brugergrænseflader. |
| **Forventet læringsudbytte [SOLO]** | Programmeringssporet:   * [Unistrukturel] Du kan genkende enkelte kendetegn ved C# **Extension Methods** |
| **Din forberedelse** | * Overview – ExtensionMethods (video: 8:48; se i planen i ItsLearning) * [Extension methods](https://www.linkedin.com/learning/c-sharp-essential-training-1-syntax-and-object-oriented-programming/extension-methods) (video 6:03) * [How to: Implement and Call a Custom Extension Method](https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/classes-and-structs/how-to-implement-and-call-a-custom-extension-method) |

I denne opgave skal du se nærmere på udvikling og brug af extension-metoder. Selvom det ikke er tit, at du kommer til at lave dem, støder du ofte på dem, og det er vigtigt, at du forstår konstruktionen, da den (sammen med lambda-udtryk) også er afgørende for bl.a. at kunne forstå og anvende LINQ.



# Øvelse 1: Terminologi

Del teamet op i mindre grupper og brug **Ordet rundt** til at reflektere over begreberne: ”static”, ”Arrange, Act, Assert (AAA)”, ”nullable”, ”this”, ”extension methods”.  
Sørg for at alle får mulighed for at tale.

*Tidsramme: ca. 20 minutter*

**Benyt parprogrammering til alle følgende øvelser.**

# Øvelse 2: Implementering af Shift()-metoden

Af uransagelige grunde ønsker du at udvide string-klassen med en ekstra metode *Shift(int shift),* som returnerer en ny tekststreng, hvis indhold er roteret et antal karakterer (antal er angivet i int-parameteren shift). F.eks. s.Shift(2) roterer alle strengens karakterer to pladser til højre, hvor de to sidste karakterer bliver rykket ind i starten af strengen, dvs. ”abcdef”.Shift(2) returnerer strengen ”efabcd”.

Da du ikke kan ændre implementeringen af selve string-klassen, må du ty til at lave en extension-metode til klassen.

## Øvelse 2.1: Implementér Shift() som extension-metode

Udfør følgende:

* Opret et nyt konsolprojekt i Visual Studio med navnet ”ExtensionMethods”
* Opret en ny klassefil StringExtensions.cs i projektet
* Implementér extension-metoden Shift(), som skal kunne kaldes med én int-parameter ”shift”, der angiver antal karakterer, der skal rykkes mod højre eller venstre
  + Positive tal rykker karaktererne til højre
  + Negative tal rykker karaktererne til venstre
  + Hvis tekststrengen er null, skal der returneres en tom tekststreng

## Øvelse 2.2: Test af Shift()-metoden

Udfør følgende:

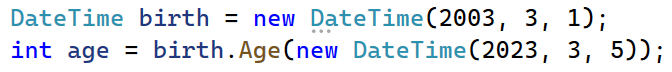
* Opret derefter et testprojekt i løsningen
* Indsæt testen angivet i testfilen ”Ex49-ExtensionMethods-UnitTest-Shift.txt” (fra materialemappen i GitLab)
* Kør testen, og ret fejl om nødvendigt

# Øvelse 3: Extension-metoden Age()

Fortsæt med konsolprojektet ”ExtensionMethods” fra øvelse 2. Nu skal du selv oprette en række af unittests som følger Arrange, Act, Assert.

## Øvelse 3.1: Extension-metoden Age()

Udfør følgende:

* Tilføj en extension-metode *Age(DateTime dt)* til DateTime-datatypen, som beregner og returnerer den nuværende alder. Et eksempel på brugen af denne extension-metode er:  
    
    
    
  hvor Age()-metoden returnerer alderen, dvs. antal år der er gået fra fødselsdatoen d. 1/3-2003 til referencetidspunktet angivet som inputparameteren, i dette tilfælde 5/3-2023 – så 20 år.
* Overvej nu, hvad der kunne være relevant at teste i forbindelse med ovenstående extension-metode – lav en liste
  + *Kan du ikke finde på noget? Helt sikker? Så se nedenstående note vedrørende ”Til inspiration: Hvad kan være relevant at teste?”*
* Ud fra din ovenstående list (*eller inspirationslisten*), opret nu selv dine egne unittests:
  + Tilføj dem til testprojektet fra øvelse 2
  + Benyt dig af Arrange, Act, Assert (se på tidligere øvelser for inspiration)
* Implementér extension-metoden, så alle unit tests bliver grønne
  + *Kan du ikke få dine unittests til at blive gode? Så benyt ” Ex49-ExtensionMethods-UnitTest-Age(Inspiration).txt” fra materialemappen på GitLab*

### Til inspiration: Hvad kan være relevant at teste? (Kun for dem der er på bar-bund)

Kan du ikke finde på noget? Så er her lidt inspiration til forskellige relevante unit tests:

* Hvis fødselsdatoen og referencetidspunktet er ens, skal metoden returnere 0
* Hvis referencetidspunktet er et år efter fødselsdatoen, skal metoden returnere 1
* Hvis referencetidspunktet er før fødselsdatoen, skal metoden returnere 0
* Hvis referencetidspunktet er 10 år efter fødselsdatoen, skal metoden returnere 10
* (Lidt mere udfordrende krav)
  + Hvis referencetidspunktet er 29. februar, og referencetidspunktet er en dag før d. 28. februar i et skudår, skal metoden returnere 0
  + Hvis der ikke angivet noget referencetidspunkt, skal der benyttes DateTime.Now

# Øvelse 4: Lav dine egne extension-metoder

Fortsæt med at implementere extension-metoder for udvalgte datatyper, som du selv har manglet i din egen kode.