Opgave 18

Klasser, objekter

# Formål

Formålet med opgaven er at lære hvordan man opretter klasser og få en begyndende forståelse for OOP tankegangen.

# Mål

Opgavens mål er

1. at kunne oprette et objekt af en klasse
2. at kunne skelne mellem de forskellige objekter
3. at kunne oprette en klasse med auto-properties og metoder

# Car klasse

Opret et nyt Console Application og navngiv det *EX18SimpleClasses*

## Opgave 1

I denne opgave skal du tilføje en ny klasse (Car) til dit projekt. Klassen skal indeholde fem properties (make, model, color, price, isSold) til at starte med.

1. Tilføj en klasse ved at højreklikke på projektet, og vælge *Add -> Class…*
2. Navngiv klassen: *Car*.*cs*
3. Opret fem properties:
   1. Make – skal kunne indeholde informationer om bilens producent
   2. Model – skal kunne indeholde informationer om bilens model
   3. Color – skal kunne indeholde informationer om bilens farve
   4. Price – skal kunne indeholde en heltalspris for bilen
   5. IsSold – skal kunne indeholde informationerne om bilen er solgt eller ej

Opret nu to nye objekter af klassen i din *Main*, tilskriv følgende værdier:

Bil 1:  
Make = ”Dacia”  
Model = ”Logan”  
Color = ”Beige”  
Price = 170000  
IsSold = false

Bil 2:  
Make = ”Toyota”  
Model = ”Yaris”  
Color = ”Rød”  
Price = 89000  
IsSold = true

Opret også en metode, *public void PrintInfo(),* som udskriver informationerne til konsollen. Udskriften skal se sådan ud hvis bilen ikke er solgt:

Bilen er en Dacia Logan i farven beige. Prisen er 170000 DKK.

Hvis bilen er solgt (*IsSold == true*)skal udskriften være:

SOLGT! Bilen er en Toyota Yaris i farven rød. Prisen er 89000 DKK.

## Opgave 2

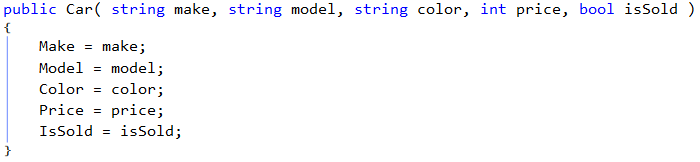
Det er lidt bøvlet, at man er nødt til at starte med oprette et objekt, for bagefter at tilskrive værdierne til objektet. Det er der dog mulighed for at ændre! Når et objekt oprettes, bruges klassens *constructor*. Alle klasser har en nedarvet *constructor*, som blot opretter objektet uden at gøre noget ved objektet. Den kan skrives sådan her:

public Car()

{

}

I stedet for bare at bruge en tom *constructor*, så vil vi oprette en *constructor* der tager i mod oplysningerne om bilen med det samme, og gemmer dem i objektet:



Med denne *constructor* kan man så oprette et objekt med informationerne direkte:



Prøv at oprette en Honda Civic med de informationer der er lige ovenover og udskriv informationerne om bilen.

## Opgave 3

Ligesom en klasse nedarver en *constructor*, så nedarver den også en metode der hedder *ToString()*. Det er blandt andet den metode der kaldes, hvis vi laver en *Console.WriteLine( myCar )*. Prøv at udskrive en af dine biler ved at bruge *Console.WriteLine()*:



Resultatet du får i konsollen er ikke særligt brugbart, vel?

I stedet kan vi overskrive den indbyggede *ToString()* metode med vores egen. Det gøres sådan her:

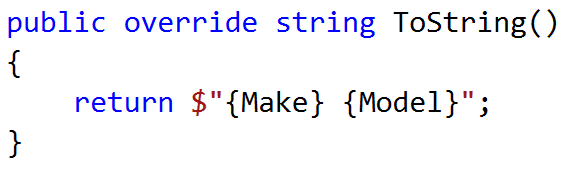


*Override* er et keyword du muligvis ikke har set før. *Override* betyder at man overskriver en nedarvet metode med samme navn. *Return base.ToString()* betyder faktisk, at den bare bruger den nedarvede metode alligevel. Vi skal derfor udskifte *return base.ToString()* med vores egen string. Altså, hvordan skal vores bil præsenteres hvis man forsøger at konvertere et *Car* objekt til en string.

Vi vil gerne have, at bilen præsenteres sådan her:

Dacia Logan

Vi skal derfor ændre vores *ToString()* metode til følgende:

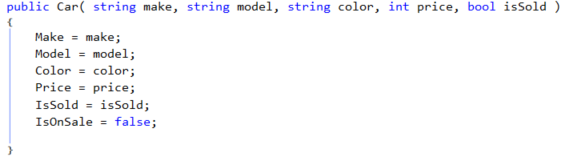


Prøv at ændre din *ToString()* metode så den matcher det der står ovenfor, og brug *Console.WriteLine()* til at udskrive bilen til konsollen.

## Opgave 4

Du skal nu tilføje en metode til dit program, som kan sætte bilen på tilbud. Det gøres ved at tilføje en *property* kaldet IsOnSale.

I din *constructor* skal du tilføje en linje kode, således at alle biler som standard **ikke** er på tilbud. Din *constructor* skal altså se sådan her ud:



Nu vil alle biler der oprettes have en IsOnSale variabel med værdien *false*.

Tilføj nu en *public void PutOnSale()* metode. Denne metode skal ændre *IsOnSale* til *true* og udregne en ny pris. Den nye pris skal være 10% lavere end den gamle.