SWC – Elementer i klassedefinition

Indhold

[Eksempel 2](#_Toc494610529)

[Karakteristika for en **klasse** 3](#_Toc494610530)

[Karakteristika for et **instance field** 4](#_Toc494610531)

[Karakteristika for en **constructor** 5](#_Toc494610532)

[Karakteristika for en **property** 6](#_Toc494610533)

[Karakteristika for en **metode** 7](#_Toc494610534)

# Eksempel

public class Car

{

private string **\_**model;

private int **\_**price;

public Car(string model, int price)

{

\_model = model;

\_price = price;

}

public string Model

{

get { return \_model; }

}

public int Price

{

get { return \_price; }

set { \_price = value; }

}

public void PrintInformation(string header)

{

Console.WriteLine(header)

Console.WriteLine($"{Model} costs {Price} kr.");

}

}

# Karakteristika for en **klasse**

1. Er altid **public**
2. Navngives med STORT bogstav først
3. En klasse pr. fil (f.eks.: klassen **Car** bor i filen **Car.cs**)
4. Vil rumme definitioner af
   1. Instance fields
   2. Constructor
   3. Properties
   4. Metoder

public class Car

{

private string **\_**model;

private int **\_**price;

public Car(string model, int price)

{

\_model = model;

\_price = price;

}

public string Model

{

get { return \_model; }

}

public int Price

{

get { return \_price; }

set { \_price = value; }

}

public void PrintInformation(string header)

{

Console.WriteLine(header)

Console.WriteLine($"{Model} costs {Price} kr.");

}

}

# Karakteristika for et **instance field**

1. Er altid **private**
2. Før navnet skal **typen** for et instance field angives, f.eks. **string** , **int** osv.
3. Navngives med \_ (underscore) først, derefter navnet. Selve navnet starter med småt begyndelsesbogstav
4. Hvert objekt har sit eget sæt af instance fields

public class Car

{

private string **\_**model;

private int **\_**price;

public Car(string model, int price)

{

\_model = model;

\_price = price;

}

public string Model

{

get { return \_model; }

}

public int Price

{

get { return \_price; }

set { \_price = value; }

}

public void PrintInformation(string header)

{

Console.WriteLine(header)

Console.WriteLine($"{Model} costs {Price} kr.");

}

}

# Karakteristika for en **constructor**

1. Er altid **public**
2. Navngives med STORT bogstav først
3. Hedder det samme som klassen den er defineret i
4. Har ikke nogen returværdi, heller ikke **void**
5. Udføres når et nyt objekt skabes, og har til formål at sikre, at det skabte objekt er i en veldefineret og brugbar tilstand fra start. Dette sikres ved at initialisere alle instance fields i constructoren.
6. En constructor kan godt have et antal parametre
7. Man kan godt definere flere constructors for en klasse

public class Car

{

private string **\_**model;

private int **\_**price;

public Car(string model, int price)

{

\_model = model;

\_price = price;

}

public string Model

{

get { return \_model; }

}

public int Price

{

get { return \_price; }

set { \_price = value; }

}

public void PrintInformation(string header)

{

Console.WriteLine(header)

Console.WriteLine($"{Model} costs {Price} kr.");

}

}

# Karakteristika for en **property**

1. Er altid **public**
2. Navngives med STORT bogstav først
3. Hvis vi gerne vil kunne spørge på værdien af en property: definér **get**-delen (er altid defineret)
4. Hvis vi gerne vil kunne ændre værdien af en property: definér **set**-delen (kun defineret hvis behov)
5. Ofte – men ikke altid – er en property knyttet direkte til et ”bagvedliggende” instance field
6. Andre properties kan være ”beregnede” – ingen simpel sammenhæng med et instance field. Disse har som regel kun **get**-delen defineret

public class Car

{

private string **\_**model;

private int **\_**price;

public Car(string model, int price)

{

\_model = model;

\_price = price;

}

public string Model

{

get { return \_model; }

}

public int Price

{

get { return \_price; }

set { \_price = value; }

}

public void PrintInformation(string header)

{

Console.WriteLine(header)

Console.WriteLine($"{Model} costs {Price} kr.");

}

}

# Karakteristika for en **metode**

1. Er som regel **public**, men i større klasser kan det give god mening også at have **private** metoder, for at opnå en klarere struktur af koden
2. En definition af en metode skal altid rumme
   1. **Access modifier**: **public** eller **private**
   2. **Type** for returværdien (evt. **void** hvis ingen returværdi)
   3. Metodens **navn.** Navngives med STORT bogstav først
   4. **Start-parentes** (
   5. Hvis metoden benytter **parametre**, angives disse med **type** og **navn**. Hvis der er flere parametre, adskilles disse med **komma** ,
   6. **Slut-parentes** )
   7. Metodens **krop.** Skrives mellem **{** og **}**. Hvis metoden skal returnere en værdi, skal der være en **return**-statement i kroppen

public class Car

{

private string **\_**model;

private int **\_**price;

public Car(string model, int price)

{

\_model = model;

\_price = price;

}

public string Model

{

get { return \_model; }

}

public int Price

{

get { return \_price; }

set { \_price = value; }

}

public void PrintInformation(string header)

{

Console.WriteLine(header);

Console.WriteLine($"{Model} costs {Price} kr.");

}

}