



Campus Nyköping

PROGRAMMERING I .NET C# 1 - 05

C# : Mer metoder och Scope

Förra gången

- Lösningsförslag Labyrinten
- Switchar
- Metoder

Idag

- Scope/Omfång
- Repetition och mer om metoder
- Code along - Gunnars Pizzeria
- Code along – Bankomaten
- Annat att repetera eller förklara bättre?

Vad är scope?

- Variabler är tillgängliga i koden beroende på var de är deklarerade
- Man skiljer på tre olika typer av **scope/omfång**
 - Klass
 - Metod
 - Block

Class Level Scope

- Att deklarera variablerna i en klass, men utanför en metod kan nås direkt var som helst i klassen.
- Dessa variabler kallas också fält eller klassmedlemmar.
- Variabeln omfång på klassnivå kan nås av metoderna i den klass där den deklareras.

```
using System;
class Program
{
    // from here class level scope starts
    // this is a class level variable
    static int a = 10;
    // declaring a method
    static void Main(string[] args)
    {
        // accessing class level variable
        Console.WriteLine(a);
    } // here method ends
} // here class level scope ends
```

Method Level Scope

- Variabler som deklarerats i en metod har omfång på metodnivå.
 - Dessa är inte tillgängliga utanför metoden.
 - Dessa variabler kan dock nås av koden inuti metoden.
 - Dessa variabler kallas lokala variablerna.
 - Det uppstår ett kompileringsfel om dessa variabler deklarerats två gånger med samma namn i samma omfång.
 - Dessa variabler finns inte när metodens körning är över
- ```
◦ public void display()
◦ {
◦ int m = 47;
◦ // Fungerar!
◦ Console.WriteLine(m);
◦ }
◦ public void display2()
◦ {
◦ // Fungerar inte!
◦ Console.WriteLine(m);
◦ }
```

# Block Level Scope

- Dessa variabler deklarerar i allmänhet inuti for eller where-loop.
  - Dessa variabler kallas också loopvariabler eller satsvariabeln eftersom de har begränsat sitt omfång till det block de befinner sig i.
  - Variabeln som deklarerar utanför looperna är också tillgänglig inom de kapslade slingorna. Det betyder att en klassnivåvariabel kommer att vara tillgänglig för metoderna och alla slingor. Metodnivåvariabeln kommer att vara tillgänglig för loop och metod inuti den metoden.
  - En variabel som deklarerar inuti en loop kommer inte att vara synlig på utsidan av looperna
- ```
◦ int i = 0;  
◦ for (i = 0; i < 4; i++)  
◦ {  
◦     Console.WriteLine(i);  
◦ }  
◦ Console.WriteLine(i);  
  
◦ for (int j = 0; j < 5; j++)  
◦ {  
◦     Console.WriteLine(j);  
◦ }  
◦ // Fungerar inte!  
◦ Console.WriteLine(j);
```

Code Along - Scope

- Kod: Scope.zip

Repetition Metoder

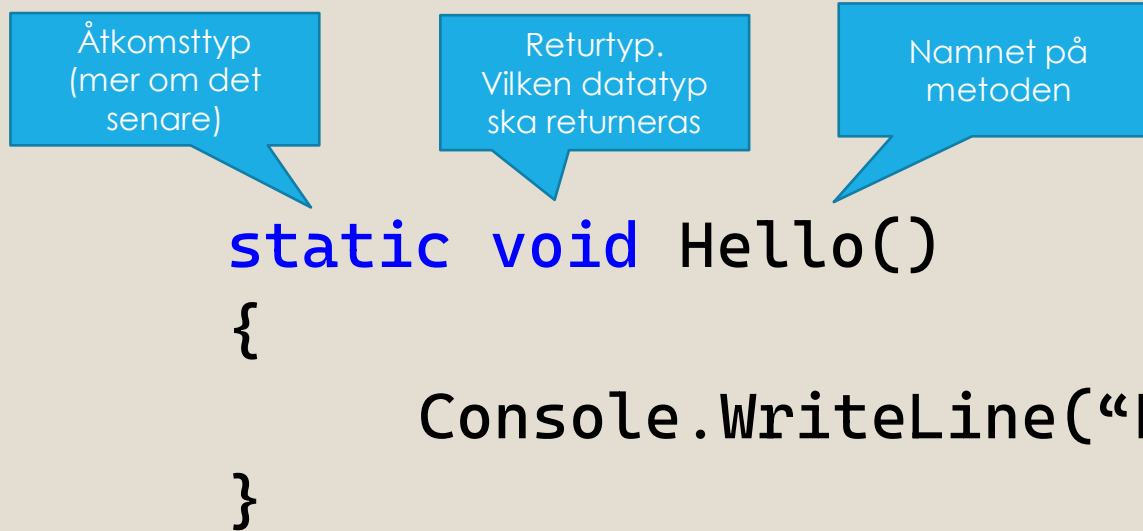
- Metoder är ett sätt att bryta isär koden i mindre hanterbara kortare kodstycken.
- Man kan jämföra metoder vid en verkstad, som gör en specifik uppgift.
- Man kan be verkstaden göra saker, såsom att räkna ut saker eller visa en text.
- Om verkstaden behöver veta något för att kunna utföra sitt arbete så måste vi förse den med information, **inparametrar**.
- Om man inte vill ha något svar tillbaka från verkstaden så anger vi **returdatatypen** till **void**. Då kan vi inte använda nyckelordet return.
- Om man vill ha ett svar från verkstaden, så måste man ange vad man vill få tillbaka, och anger då detta i returdatatypen. Då använder vi nyckelordet return.

Fyra typer av metoder

- Metod som inte tar in några inparamterar och inte returnerar något.
- Metod som tar in inparametrar men inte returnerar något
- Metod som inte tar in inparametrar men returnerar något
- Metod som tar in inparametar och returnerar något

Metod utan inparametrar eller returdata

- En metod är ett stycke kod med ett namn
- Varje metod har:



Returtypen void betyder att inget ska returneras

Metod utan inparametrar eller returdata

Vi föräntar oss
ingen data
tillbaka

Inga inparametrar

```
static void Hello()  
{  
    int summa = 34 + 9;  
    Console.WriteLine(summa);  
}
```

Så här använder vi metoden: `Hello();`

Metoder med inparametrar, men utan returdata

- Varje metod har:

Åtkomsttyp
(mer om det senare)

Vi förväntar oss
ingen returdata

Namnet på
metoden

Eventuella
inparametrar

```
static void SayHello(string name)
{
    Console.WriteLine("Hejhej " + name);
}
```

Så här använder vi metoden: `SayHello("Micke");`

Metod utan inparametrar men med returdata

Vi förväntar oss
en int tillbaka

Inga inparametrar

```
static int Hello()  
{  
    int summa = 34 + 9;  
    return summa; // Detta ska vara en int  
}
```

Såhär använder vi metoden: `int result = Hello();`

Metoder med inparametrar och returdata

Vi förväntar oss
att en sträng
returneras

Inparametrar

```
static string SayHello(string name, int age)
{
    string text = "Hejhej " + name + " som är " + age + " år gammal";

    return text; // Detta ska vara en string
}
```

Så här använder vi metoden: `string result = Hello("Micke", 57);`

Hur kan man anropa metoder?

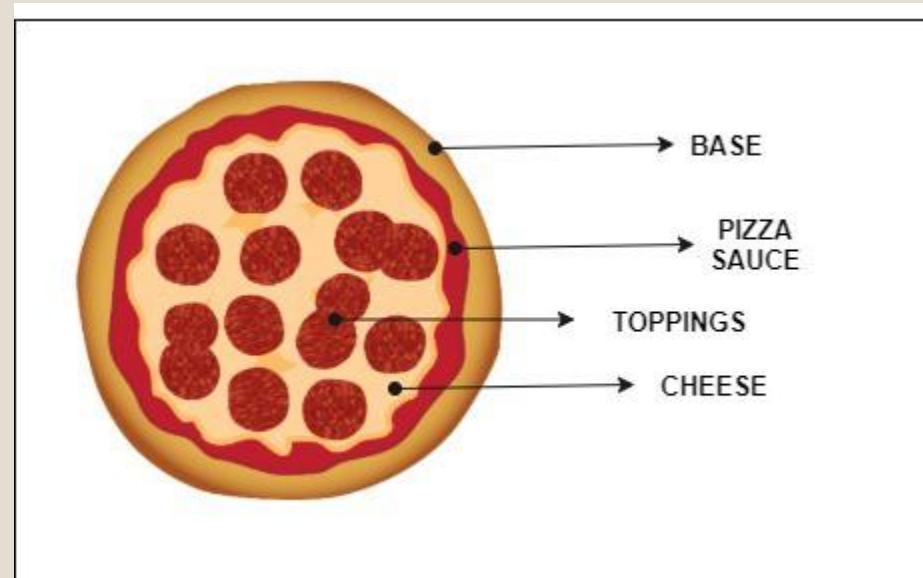
- Huvudprogrammet kan anropa metoder (Main() är faktiskt en metod också)
- Metoder kan anropa andra metoder
- Metoder kan anropa sig själv

Extra grej - Slump

- `Random rnd = new Random();`
- `int number = rnd.Next(1, 4);` // Ger en heltalssiffra mellan 1 och 3

Code along - Gunnars Pizzeria

Kod: Pizzerian.zip





Övning 5 – Switch och metoder

- Bankomaten
 - Skapa en Console Application som:
 - med hjälp av en switch-sats skapar en inmatningsmeny för en bankomat. Alternativen ska vara:
 - [I]nsättning
 - [U]ttag
 - [S]aldo
 - [A]vsluta
 - Skapa fyra metoder som hanterar dessa funktioner
 - Det ska inte gå att ta ut mer pengar än vad det finns på kontot
 - Det ska inte gå att sätta in mer än 10.000 kr per gång (lagen om penningtvätt)
 - Saldot ska hela tiden vara uppdaterat.

Code along - Bankomaten

- Spoiler Alert!
- Kod: Bankomaten.zip



Övning 1 - Bilfabriken

- Bygg en application som "bygger" slumpmässiga bilar.
- Delarna kan vara:
 - Underrede
 - Rostfritt eller stål
 - Hjulen
 - T ex 3 eller fyra, dubbdäck eller sommardäck
 - Motor
 - Olika antal hästkrafter
 - Inredning
 - Läderklädsel eller tyg
 - Kaross
 - Olika färg på lacken
- Skapa metoder som "bygger" varje del av bilen.
- Beräkna också hur lång tid varje moment tar.
- När bilen är klar, ska den beskrivas, och anges hur lång tid den tog att byggas.
- Genom att loopa allt detta så skapar du en "fabrik" med flera olika bilar.
- Exempel: "En bil med rostfritt underrede, med 3 hjul, 145 hästkrafter, tygklädsel med röd lack, som tog 125 minuter att bygga."
 - Osv...

Övning 2 – En dag I livet

- En dag I livet
 - Du ska simulera de olika sakerna du brukar göra en vanlig skoldag, och hålla reda på din energinivå.

Aktivitet	Din energinivå
Vakna på morgonen	70
Äta frukost	+30
Lyssna på föreläsning	-35
Lunch	+30
Jobba med övningar	-50
Middag	+20
Se TV/Träna/Plugga	-65
 - Skapa en metod som tar in två värden, aktivitet, och nuvarande energinivå.
 - Metoden returnerar din energinivå efter aktiviteten, som uppdaterar energinivå-variabeln I huvudprogrammet.
 - Anropa metoden flera gånger, med vilken aktivitet du utför.
 - Efter varje aktivitet så skriver du ut nuvarande energinivå:
 - Vaknat: 70%
 - Ätit frukost: 100%
 - Lyssnat på föreläsning 65%
 - ...OSV
 - Det går bra att hitta på egna aktiviteter, och “energi-poäng”

Övning 3 – Dice game

```
=====  
|0 0| |0 0| | 0|  
|0 0| |0 0| |  |  
|0 0| |0 0| |0  |  
=====  
Roligt, du fick två lika. Du har nu 12 poäng.
```

```
=====  
|0 0| | 0| |  |  
|0 0| |0  | |0  |  
|  | |  | |  |  
=====  
Tråkigt, summan är mindre än 9. Du har nu 9 poäng.
```

- Bygg ett tärningsspel enligt följande:
 - Tre slumpmässiga tärningar visas i konsollen.
 - Om du får tre lika tärningar, så får du 10 poäng
 - Om du får två lika tärningar så får du 3 poäng
 - Om summan av tärningarna är mindre än 9, så får du 5 minuspoäng
- Skapa ett huvudprogram, som sköter slumpalen
- Skapa en metod som ritat ut EN tärning
- Skapa sedan en metod som anropar ovanstående metod tre gånger, så totalt tre tärningar ritas ut. Det är okej att de ritas upp nedanför varandra, vilket är enklare.
- Skapa en metod som beräknar poäng, och skriver ut en text då något händer i spelet.

Annat att repetera?



Idag fram till fredag

- Gamla övningar
- Nya övningar
 - Bilfabriken
 - En dag i livet
 - Dice Game
- Alla länkar ni fått

Länkar

- [Scope of Variables in C# - GeeksforGeeks](#)
- [c# random number - W3schools](#)
- [C# Methods \(w3schools.com\)](#)
- [C# Method Parameters \(w3schools.com\)](#)
- [8. How to program in C# - METHODS - Beginner Tutorial - YouTube](#)