

Grundkurs i C# och .NET Framework

Kurs för NFI DEL2

Stefan Holmberg, Systemmentor AB

Vad är en loop ?

- Iteration betyder upprepning
- En loop innebär att man upprepar något till dess att ett visst villkor är uppfyllt eller att man väljer att avbryta.
- Avbryta en loop gör man med kommandot **break** ;

For loop

- Grunden för en for loop är

`for (startvärde; villkor för loopen; uppräkning)`

- Uppfylls inte villkoret går koden aldrig in i koden i loopen
- Följande loop upprepas 10 gånger. För varje gång ökas värdet på variabeln raknare med 1.

```
for (int raknare=1; raknare < 11; raknare++)  
{  
    //Kod som körs i loopen  
}
```



For loop

Startvärde räknare

Villkor

Ökning räknare

```
for (int i = 1; i < 11; i++)  
{  
    Console.WriteLine(i);  
}
```

While loop

Ett annat sätt att skriva en loop är att använda **while**. Detta exempel gör samma sak som for loopen vi skrev tidigare.

```
int i = 1;
```

Startvärde

```
while (i < 11)
```

Villkor

```
{
```

```
    Console.WriteLine(i);
```

```
    i++;
```

Ökning räknare

```
}
```

Break, continue, goto

Tillsammans: vi skriver en oändlig loop

Vi ska göra en SPELMENY:

1. Starta spelet
2. See highscore
3. Avsluta



For each loop

Används när man arbetar med objekt. Detta exempel loopar igenom alla bokstäver i en textsträng.

```
string text = "Detta är min text";

foreach (char bokstav in text)
{
    Console.WriteLine( bokstav.ToString());
}
```

C#: Loops

PDF Chapter 6. Loops

<https://introprogramming.info/english-intro-csharp-book/read-online/chapter-6-loops/>

Labbar Loopar 1-12

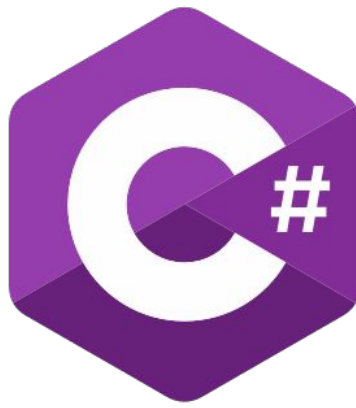


C#: String handling

Chapter 13. Strings and Text Processing

<https://introprogramming.info/english-intro-csharp-book/read-online/chapter-13-strings-and-text-processing/>

Labbar Stränghantering 1-11



C#: Metoder

Vad är en metod?

- Vi har hittills arbetat med en metod `main()`, som är den första metod som anropas i ett konsollprogram i C#. Det går även att skapa egna metoder.
- En metod är ett sätt att dela in koden i mindre delar. Kod som läggs i en metod bör hänga ihop på något sätt, ha en gemensam funktion tex hämta största värde i en array, räkna ut ett fakturabelopp.
- Metoden kan sedan anropas på flera ställen i koden och kan därför vara ett sätt att minska ned mängden kod som behöver skrivas.

Grunderna

- En metod kan ha en eller flera inparametrar. Det är värden som skickas in och används i metoden. Det går även att skapa en metod utan inparametrar.
- En metod kan returnera ett, flera eller inga värden. En metod kan även skapas utan att returnera något. Skriver man void returneras inga värden annars returneras den datatyp man anger



```
public int BeraknaFakturaBelopp(int fakBelopp)
{
}
```

Exempel

I detta exempel returnerar metoden en int. Inparametern används inne i metoden och det nya värdet returneras

```
public int BeraknaFakturaBelopp(int fakBelopp)
{
    //Här skapas ett värde som returneras
    int vardeSomReturneras = fakBelopp * 1.25;

    return vardeSomReturneras ;
}
```

Anropa en metod

En metod kan sedan anropas någonstans i koden tex från Main.

```
static void Main(string[] args)
{
    int number = 20;

    int returVarde = BeraknaFakturaBelopp (number);

    Console.WriteLine(" Detta är mitt fakturabelopp" + returVarde );
}
```


Demo

- Olika typer av parametrar:
 - Värdeparameter
 - out-parameter
 - ref-parameter
 - params-parameter
 - optional parameter

C#: Metoder

Chapter 9. Methods

<https://introprogramming.info/english-intro-csharp-book/read-online/chapter-9-methods/>

Structs

- Klumpa ihop många tillhörande data till en samma variabel



Vi skapar en `GetSeasonStats(int jerseyNumber)`

Vi vill ju returnera mål OCH assists OCH poäng OCH plus/minus etc etc etc