

C#- If och Switch-satser

Att göra beslut

Ofta förekommer det att vi i programmerandet måste göra beslut och avgöranden för att kunna köra den kod vi vill köra. För att uppnå detta använder vi oss av *if* och *switch* satser. Säg att vi har ett program som kollar om inmatningsvärdet är mellan 1-10, och att om värdet är under 5 vill vi skriva ut att värdet är lågt, och ifall värdet är mellan 5 eller 10 vill vi skriva ut att värdet är högt. Detta uppnår vi enkelt med en *if*-sats.

If/else-satsen

En *if*-sats består av ordet *if* följt av parenteser med ett argument därefter sluter vi koden inuti vi vill ska köras med hjälp av klaffar/brackets.

Exempel:

```
if (minInteger < 5)
{
    Console.WriteLine("Värdet är lågt");
}
else
{
    Console.WriteLine("Värdet är högt");
}
```

Här kollar vi om variabeln `minInteger` har ett värde mindre än 5, om värdet är mindre än 5 så körs koden inom de första klaffarna (" { } ") ut. Under dessa finns satsen *else*, där skriver vi vad vi vill göra om ovanstående argument (`minInteger < 5`) inte stämmer överens med värdet. Detta kan läsas som "Om `minInteger` är mindre än fem, skriv ut att värdet är lågt, annars skriv ut att värdet är högt". Vill man endast köra kod om en variabel stämmer överens med argumentet kan man utebliva *else*-satsen.

Exempel:

```
if (minInteger < 5)
{
    Console.WriteLine("Värdet är lågt");
}
```

If/else if/else

Ibland kan det förekomma att vi vill använda oss av flera argument, exempelvis där vi kollar om ett tal är över 5, under 5 och ifall talet är 5, så kan vi utöka med satsen *else if*.

Exempel:

```
if (minInteger < 5)
{
    Console.WriteLine("Värdet är lågt");
}
else if (minInteger == 5)
{
    Console.WriteLine("Värdet är medel");
}
else
{
    Console.WriteLine("Värdet är högt");
}
```

Här har vi lagt till en *else if*-sats i tidigare exempel där vi kollar om vår variabel är exakt 5. En *if*-sats kan innehålla så många *else if*-satser som man vill använda, men för att undvika att skapa överflödiga kod kan man istället använda sig av en *switch*-sats.

Nästlade if-satser

Att nästla satser gör man när man vill utföra fler beslut efter man har gått igenom ett första argument. Säg att vårt värde var lägre än 5 men nu vill vi också köra en kod bara om värdet var 1. Då kan vi nästla vår *if*-sats genom att skriva en till *if*-sats inuti vårt kodblock.

Exempel:

```
if (minInteger < 5)
{
    if (minInteger == 1)
    {
        Console.WriteLine("Lägsta värdet!");
    }
    else
    {
        Console.WriteLine("Värdet är lågt!");
    }
}
```

Regler för nästlade *if*-satser är samma som att skapa vanliga *if*-satser.

Switch-satsen

Switch används när vi behöver testa vår variabel mot en lista av värden. Säg att vi ska skapa en meny där användaren har 4 olika val och behöver mata in en siffra för vald undermeny. För att undvika att skapa överflödigt och ful kod genom att använda oss av massvis med *if*-satser så kan vi göra det enkelt och snyggt med hjälp av en *switch*-sats. Säg att vi nu har vår meny med 4 val och när användaren matat in sitt val vill vi kolla upp detta och ge användaren rätt undermeny. Med hjälp av en *switch*-sats kan det se ut såhär:

Exempel:

```
switch (userChoice)
{
    case 1:
        Console.WriteLine("Undermeny 1");
        break;
    case 2:
        Console.WriteLine("Undermeny 2");
        break;
    case 3:
        Console.WriteLine("Undermeny 3");
        break;
    case 4:
        Console.WriteLine("Du har avslutat programmet");
        break;

    default:
        Console.WriteLine("Inget stämde överens!");
}
```

`switch (userChoice)` här deklarerar vi en *switch*-sats med argument inuti parenteserna därefter omsluter vi koden inuti med klaffar. Första raden inuti står det "case 1:" och vi testar om vårt värde i `userChoice` är lika med 1, stämmer detta överens så kör vi koden på raden under och avslutar vårt "case" med `break`. Stämmer inte `userChoice` med vårt första case så jämför vi med den andra och så vidare. Det finns ingen gräns på hur många case vi får använda. När vi har matchat med någon case, kört vår kod och sedan kommit till `break` så hoppar vi ut ur vår *switch*-sats eftersom vi inte längre behöver testa `userChoice` och jämföra mot något annat case. Längst ner finns en "default", denna kod körs om vi inte har matchat med någon case. Default är valfritt att tillägga i en *switch*-sats och behöver inte vara med om vi inte vill ha den.

Nästlade switch-satser

Detta görs på samma vis som när man nästlar en *if*-sats.