

Hur fungerar "switch"-satsen?

Istället för att skriva "if-else if"-satser där exakt samma kontrolluttryck utvärderas kan du göra programmet mer lättläst genom att istället använda en "switch"-sats, som ser ut på följande sätt:

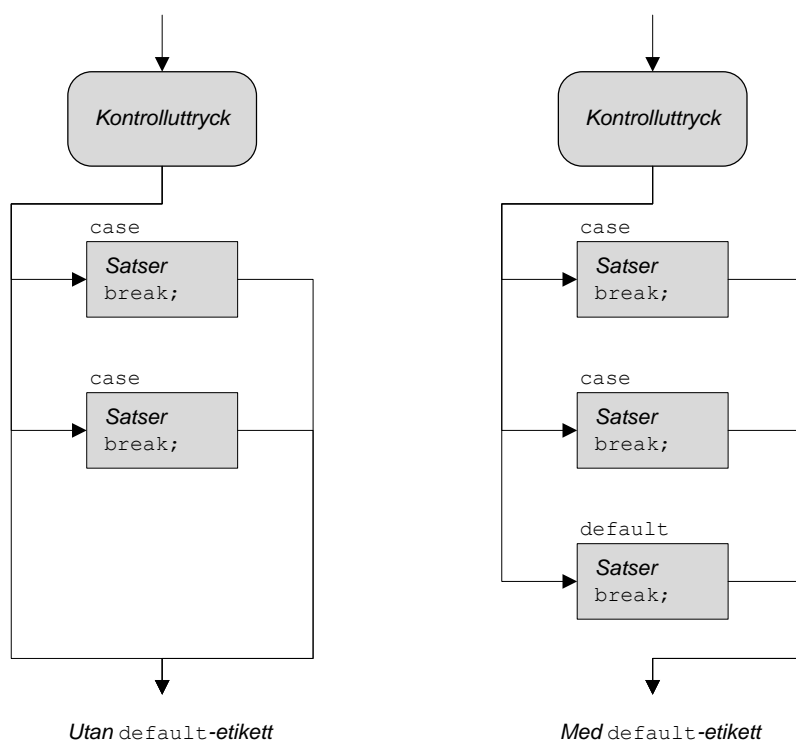
```
switch (Kontrolluttryck)
{
    case Konstantuttryck1:
        Satser
        break;

    case Konstantuttryck2:
        Satser
        break;

    default:
        Satser
        break;
}
```

Kontrolluttryck utvärderas en gång och staserna under den *case*-etikett vars *Konstantuttryck* är lika med resultatet av *Kontrolluttryck* körs fram till *break*-satserna.

Om inget av *Konstantuttryck*-värdena är lika med värdet av *Kontrolluttryck* kommer satserna under *default*-etiketten att köras. Det är valfritt att använda *default*-etiketten.



Figur 1. Flöden genom switch-satser.

Om värdet av Kontrolluttryck inte är lika med någon av `case`-etiketterna och det saknas en `default`-etikett fortsätter körningen av programmet efter ”switch”-satsens avslutande klammerparentes.

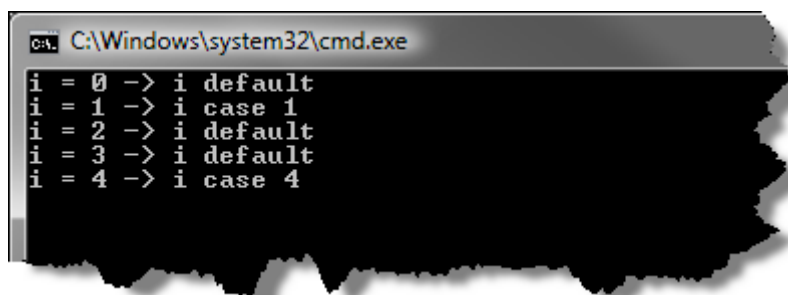
Följande kod kör ”switch”-satsen fem gånger, och värdet på `i` går från 0 till 4.

```
for (int i = 0; i < 5; ++i)
{
    switch (i)
    {
        case 1: // om i är lika med 1
            Console.WriteLine("i = {0} -> i case 1", i);
            break;

        case 4: // om i är lika med 4
            Console.WriteLine("i = {0} -> i case 4", i);
            break;

        default: // // om i inte är lika med 1 eller 4
            Console.WriteLine("i = {0} -> i default", i);
            break;
    }
}
```

Genom att titta på det som koden skriver ut kan du säga vilken av etiketterna som kördes för de olika värdena `i` hade.



Figur 2. Utskrift som tidigare kod skapar.