



Av Erik
"Zkall"
Svensson
@esdev.se



C# Enum

Vad är enum? Enligt wikipedia:

Uppräkningstyp (engelska *enumerated type*, ofta förkortat *enum*) är inom [programmering](#) en [datatyp](#) vars tillåtna värden består av en fix mängd [konstanter](#), till exempel årets månader eller korten i en kortlek.

Vissa uppräkningstyper kan vara fördefinierade i [programspråket](#), till exempel [booleska datatyper](#) med [sanningsvärdena](#) sant (*true*) och falskt (*false*).

Många programspråk tillåter användaren att definiera egna uppräkningstyper, vilket kan ske i exempelvis [C++](#), [Java](#), [C#](#), [Ada](#) och [Pascal](#).

Så vad innebär det?

Om du ska ha en typ istället för att ha en string och kolla med if om strängen är lika med x så har du en enum med satta värden som du kan välja i koden.

Exempel i kod

```
12 references
public enum Typer
{
    PersonNr,
    OrgNr,
    SamordningsNr
}
```

Så deklarerar du t.ex. en enum för en class innehållande 3 olika typer. För att sedan kolla om det är så i koden:

```
#region SamordningsNR Test
var PNR = new PersonOrgNr("900985-3494");
PNR.ToString(PersonOrgNr.Format.TioSiffrorBindestreck).Should().Be("900985-3494");
PNR.ToString(PersonOrgNr.Format.TolvSiffrorBindestreck).Should().Be("19900985-3494");
PNR.ToString(PersonOrgNr.Format.TioSiffror).Should().Be("9009853494");
PNR.ToString(PersonOrgNr.Format.TolvSiffror).Should().Be("199009853494");
#endregion
```

Så refererar du classen typensnamn.värde. Ignorera should be och det som står ovan och fokusera på det inom ToString(). Dom andra termerna går jag in på i ett dokument angående Test driven utveckling.