



## Inkapsling

De tre "pelarna" som objektorienterad programmering bygger på är huvudsakligen arv, polymorfism och inkapsling. I det här dokumentet djupdyker vi på konceptet inkapsling och tittar närmare på det. Objektorienterad programmering, i stort, är menad att fungera som en reflektion av verkligheten, och vissa objekt och situationer är i verkligheten inte helt tillgängliga.

Vi kan inte se, utifrån, på vissa objekt vilka egenskaper de har. Det är svårt att avgöra bara från att se en bil hur bra den fungerar; vi behöver kanske en mekaniker för att avgöra vilka delar som behöver bytas ut eller uppgraderas, eller testköra den runt kvarteret. Vi kan inte se på en gitarr huruvida den är stämd eller inte.

Inkapsling bygger på den här idén, att all information inte alltid är tillgänglig; det kan vara en strukturellt god idé att, i kod, göra information mer separerad och otillgänglig. I det här dokumentet reflekterar vi över några av de åtkomstnivåer som vi använder oss av i specifikt C#. Då åtkomstnivåer kan appliceras till en så bred grad av medlemmar (klasser, metoder, egenskaper, etc) så kallar vi allt som åtkomstnivåer påverkar enkelt för "medlemmar".

I objektorienterad programmering tenderar metoder att vara allmänna (public), medan egenskaper tenderar att vara dolda (private).

### Public

Den mest öppna av samtliga åtkomstnivåer. Det finns inga begränsningar på att komma åt en publik medlem.

### Protected

Protected innebär att medlemmen bara kan påverkas innifrån grundklassen där den existerar, eller från en ärvd medlem. Har vi en protected medlem i klass A så kan den bara kommas åt från medlem A och dess undermedlemmar, men inte från medlem B.

### Internal

Den här åtkomstnivån innebär att endast medlemmar av samma assembly kan påverka koden. Program1.cs kan påverka alla "internal" medlemmar inom samma assembly, medan Program2.cs kan inte påverka de medlemmar inom Program1.cs som är "internal".

### Protected Internal

Kombination av Protected och Internal. Grundmedlemmen eller ärvda medlemmar kan komma åt Protected Internal-medlemmar. Medlemmar inom samma assembly kan också komma åt medlemmen. Det räcker med att vi uppfyller antingen Protected *eller* Internal för att nå medlemmen.

### Private

Den mest restriktiva åtkomstnivån. Medlemmar kan endast påverkas inom sin egen grundmedlem.