

# Genomgång Objektorientering – Grunder

När man programmerar som ni gjort hittills kallas det procedurell programmering. Kompilatorn läser koden tills man når slutet.

När det gäller objektorienterad programmering skapar man s.k. objekt som sedan kommunicerar med varandra. Objekt ligger i minnet på datorn.

Allt i Java är objekt.

Ett objekt skapar man från en **klass**.

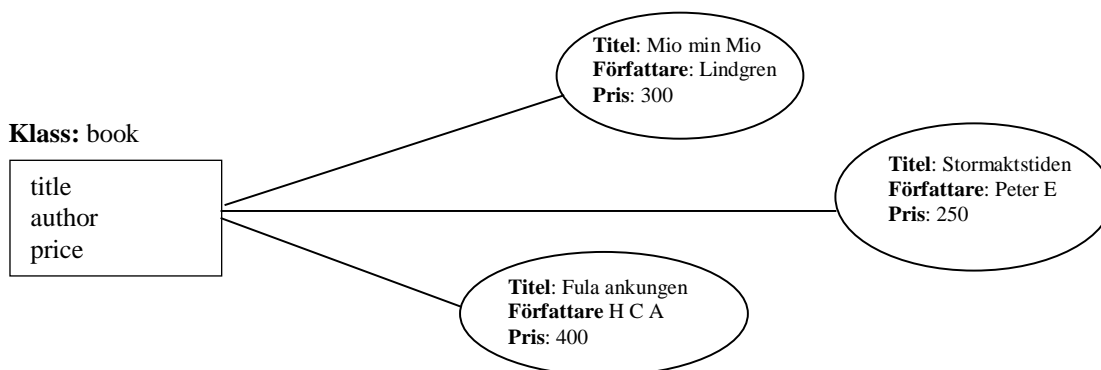
En **klass** kan ses som en mall eller form som man gjuter efter. Man skapar sedan objekt baserade på mallen.

Man kan tänka sig hur en bok ser ut: Den har en titel, den har en författare, den är av ett pris.  
Detta är mallen.

Sedan kan man göra olika böcker från den mallen, som har olika egenskaper. Detta kallas objekt.

T.ex.

Titel	Författare	Pris
Mio min Mio	Lindgren	300
Stormaktstiden	Peter E	250
Fula ankungen	H C A	400



En klass består av flera delar. Vi börjar med det grundläggande.

**Namn** – Varje klass har ett namn. Namnet blir sedan en typ som t.ex. int.

Ex. **book** enBok = new **book**();

book är alltså klassen namn och blir typen på den instans (objekt) som vi skapar.

**Fält** – En variabel som bara gäller inom klassen.

**Åtkomst** – För att skydda de variabler som finns i klassen kan man sätta dom till **private**. Det innebär att man inte kan komma åt dom utanför klassen. Istället använder man **egenskaper** för att sätta och läsa variabelns värde.

Vill man att andra ska komma åt dom kan man sätta variablerna till **public**.

## Exempel klass

```
Public class book //Här talar vi om att det är en klass vi skapar och att den ska heta book.
{
    private string title; //Variabel som endast kan nås i klassen.
    private string author;
    private int price;

    //egenskaper (properties). Används för att förhindra åtkomst till privata variabler.

    //Getter for "title"
    public String Title() {
        return title;
    }

    // Setter for "title"
    public void setTitle(String title) {
        this.title = title;
    }

    //Getter for "author"
    public String Author() {
        return author;
    }

    // Setter for "author"
    public void setAuthor(String author) {
        this.author = author;
    }

    //Getter for "price"
    public int Price() {
        return price;
    }

    // Setter for "price"
    public void setPrice(int price) {
        this.price = price;
    }
}
```

För att komma åt eller sätta ett värde i ett objekt använder man get och set-metoderna.

### Exempel på hur man skapar objekt

```
//Skapar objekt  
book historyBook = new book();
```

```
//Sätter värde till titel, författare och pris för objektet med s.k. punktnotation.  
historyBook.setTitle("Stormaktstiden");  
historyBook.setAuthor("Peter Englund");  
historyBook.setPrice(300);
```

Här använder vi egenskapen **set** eftersom vi **sätter** värdet på titel, författare och pris.

```
//Hämtar värde från objekt  
System.out.println("Titel " + historyBook.Title());
```

Här använder vi egenskapen **get** eftersom vi hämtar värdet från titel i vårt objekt.

### Skapa fler objekt

När vi nu har vår klass kan vi skapa fler objekt av **samma** klass.

Vi skapar två objekt till:

```
//Skapar objekt  
Book fairytaleBook = new book();  
Book programmingBook = new book();  
  
//Sätter värden  
fairytaleBook.setTitle("Mio min Mio");  
fairytaleBook.setAuthor("Astrid Lindgren");  
fairytaleBook.setPrice(500);  
  
programmingBook.setTitle("C++ Direkt");  
programmingBook.setAuthor("Jan Skansholm");  
programmingBook.setPrice(700);  
  
System.out.println("Titel bok 2 " + fairytaleBook.Title());  
System.out.println("Titel bok 2 " + fairytaleBook.Author());  
System.out.println("Titel bok 2 " + fairytaleBook.Price());  
  
System.out.println("Titel bok 3 " + programmingBook.Title());  
System.out.println("Titel bok 3 " + programmingBook.Author());  
System.out.println("Titel bok 3 " + programmingBook.Price());
```

Ovan satte vi ju titel till "Stormaktstiden". Nu hämtar vi det värdet och lägger i en ny variabel (enTitel) som vi sedan skriver ut.

Vi kan komma åt variabeln för titel direkt genom att sätta den till public.

Ändra åtkomst för titel till **public** i klassen.

```
historyBook.title = "Java är bäst";  
  
String enBokTill = historyBook.title;  
  
System.out.println("Boken är " + enBokTill);
```

Nu kan vi komma åt titeln direkt men det är inte bra.

Ändra tillbaka till private och se vad som händer i koden ovan.

### Uppgift:

Skapa en konsolapplikation.

Skapa en klass fordon som med följande:

Tre fält: märke, färg och pris.

Märke och färg ska vara **string** och pris ska vara **int**.

Variablerna ska vara privata och endast kommas åt med via objektens egenskaper.

Skapa två objekt och ge värden till märke, färg och pris.

Skriv ut märke och färg på båda objekten.

### Lite svårare:

Plocka ut priset för båda och summera det. Skriv sedan ut summan.