# Genomgång Klasser del 2: Metoder och konstruktorer

#### Metoder

En klass kan ha metoder. En metod är ett stycke kod som kan anropas.

En metod kan t.ex. skriva ut alla uppgifter om ett objekt. T.ex:

#### Konstruktor - tom

**Konstruktor** - När ett objekt skapas anropas en särskild sorts metod som kallas konstruktor. Man behöver inte skapa en konstruktor men skapar man en egen konstruktor måste man också skapa en tom.

#### En konstruktor har samma namn som klassen och liknar metod men har inget returvärde.

En konstruktor har inget returvärde.

En konstruktor kan vara tom: public book ();

Ex. tom konstruktor.

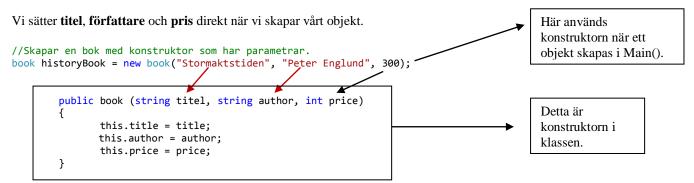
```
//Skapar objekt
book historyBook = new book();
Lägg märke till att parentesen som är tom.
```

Inga värden sätts när objektet skapas. Det gör man istället i efterhand, så här:

```
//Ger värde till titel för objektet historyBook av klassen book med s.k. punktnotation.
historyBook.Titel("Stormaktstiden);
historyBook.Author("Peter Englund");
historyBook.Price(300);
```

#### Konstruktor - med parametrar

Istället för att skapa en tom konstruktor och sätta värden i efterhand så kan man sätta värden redan när man skapar objektet.



```
public class book {
    private String title;
                             //Variabel som endast kan nås i klassen.
    private String author;
    private int price;
    private String ISBN;
    //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;
    //Contructor
    public book(String title, String author, int price)
               this.title = title;
               this.author = author;
               this.price = price;
    }
    //Empty contructor
    public book()
    }
    public void skrivUt()
          System.out.println("Titel: " + title);
          System.out.println("Författare: " + author);
System.out.println("Pris: " + price);
          System.out.println("");
    }
}
```

©Lars Persson 2016

# Exempel på hur man skapar objekt med tom konstruktor

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

//Ger värde till titel för objektet historyBook av klassen book med s.k. punktnotation.
computerBook.setTitel("Stormaktstiden);
computerBook.setAuthor("Peter Englund");
computerBook.setPrice(300);

Console.WriteLine("Titeln på boken är " + computerBook.Author());
```

# Exempel på hur man skapar objekt med en egen konstruktor.

När man skapar en egen konstruktor kan man sätta värden på variablerna redan när objektet skapas.

```
//Skapar en bok med konstruktor som har parametrar.
book EnBok = new book("Liftarens guide till galaxen", "Douglas Adams", 400);
Här slipper vi alltså att sätta värdena i efterhand utan gör det så fort objektet skapas.
//Skriver ut medlemsvariablerna med klassmetoden skrivUt();
EnBok.skrivUt();
```

# **Uppgift:**

- Skapa en klass person med följande (Eller hitta på egna värden):
  - o Tre fält: Namn, adress och email.
- Alla tre kan vara string
- Variablerna ska vara privata och endast kommas åt med setter och getter.
- Gör en konstruktor som är tom och en konstruktor med 3 parametrar.
- Gör en metod för att skriva ut variablerna.
- Skapa minst 2 objekt. Det ena objektet ska skapas med en tom konstruktor och värdena sätts då i efterhand. Det andra ska skapas med parametrar.
- Skriv sedan ut objekten med metoden du gjorde.

# Extra uppgift:

- Lägg till ett fält (variablel) t.ex. telefon.
- Gör en konstruktor till som även sätter värdet för telefonnumret.
- Gör även en metod som skriver ut alla värden.
- Testa din nya konstruktor.