



JAVAUTVECKLING – LEKTION 7

© 2016

Mahmud Al Hakim

mahmud.al.hakim@nackademin.se

www.alhakim.se

AGENDA

Olika språkversioner (Klassen Locale)

Utskrift i kommandofönster

Inläsning via kommandofönster

Standardklassen Scanner

Redigering av utskrifter med metoden format

Att läsa från en fil

Att skriva till en fil

OLIKA SPRÅKVERSIONER

Olika språk hanteras med hjälp av klassen `java.util.Locale`

Språket bestämmer texten i dialogrutor samt format för numeriska tal.

Det finns alltid ett default Locale-objekt som sätts automatiskt.

Man ändrar default objektet genom att skriva

```
Locale.setDefault(Locale.UK);
```

Eller

```
Locale.setDefault(new Locale("sv", "SE"));
```

UTSKRIFT I KOMMANDOFÖNSTER

Utskrift i kommandofönstret görs med `println` eller `print`

```
System.out.print(data);
```

```
System.out.println(data);
```

Data kan vara vilken typ som helst, ofta `String`.

`println` tömmer utskriftsbufferten

och påbörjar automatiskt en **ny rad** efter utskriften.

`print` tömmer inte utskriftsbufferten. Detta görs med en flush

```
System.out.flush();
```

ÖVNING 9.1

// version 1 med println

```
public static void main (String[] arg) {  
    System.out.println("Linda Johansson");  
    System.out.println("Storgatan 1");  
    System.out.println("123456789");  
}
```

// version 2 med print

```
public static void main (String[] arg) {  
    System.out.print("Linda Johansson\nStorgatan 1\n123456789");  
    System.out.flush(); // Rensa buffert  
}
```

INLÄSNING FRÅN KOMMANDOFÖNSTER

1. Importera paketet **java.io**;
2. Skriv **throws IOException** i huvudet på de metoder i vilka läsning skall göras.
3. Deklarera en ström av typen **BufferedReader**.
4. Läsning görs med metoden `readLine` som läser en hel rad i taget.

ÖVNING 9.4

```
import java.io.*;
public static void main (String[] arg) throws IOException {
    BufferedReader input = new BufferedReader(new InputStreamReader(System.in));
    String s = input.readLine();
    double storst = Double.parseDouble(s); // första talet
    while (true) {
        s = input.readLine();
        if (s == null) // Användaren trycker CTRL+Z / CMD+D
            break;
        double tal = Double.parseDouble(s); // nästa tal
        if (tal > storst)
            storst = tal;
    }
    System.out.println("Det största talet är " + storst);
}
```

PROBLEM MED SVENSKA TECKEN

Program som skickar svenska tecken till konsolfönstret via bl.a. metoderna print och println visar felaktiga tecken. T.ex.

```
System.out.print("åäö") // visar konstiga tecken
```

Detta beror på att terminalfönstret som finns i windows INTE stödjer Unicode.

Lösning

Kompilera med hjälp av en extra parameter som ändrar kodningen till Unicode så här:

```
javac -encoding utf-8 Fil.java
```

STANDARDKLASSEN SCANNER

Standardklassen Scanner finns i paketet java.util.

En Scanner används för att läsa data från bl.a. kommandofönstret och filer.

Ett Scanner-objekt hämtar indata från en ström av tecken, sätter ihop dessa till ord, heltal, eller flyttal.

SCANNER — EXEMPEL

```
import java.util.Scanner;
public class ScannerDemo {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in); // Skapa ett Scanner-objekt
        System.out.print("Ange ett decimaltal x: ");
        double x = scan.nextDouble();          // Läser double
        System.out.print("Ange ett heltal n: ");
        int n = scan.nextInt();                 // Läser int
        System.out.println("x^n = " + Math.pow(x, n));
    }
}
```

REDIGERING AV UTSKRIFTER MED METODEN FORMAT

Metoden format finns för strömmar av typerna String, PrintWriter och PrintStream.

Exempel 1: Visa tal med två decimaler

```
double pi = Math.PI;  
String resultat = String.format("%.2f", pi);  
JOptionPane.showMessageDialog(null, resultat);
```

Se faktaruta
sidan 197

INLÄSNING FRÅN EN TEXTFIL

```
public static void main(String[] args) throws IOException {  
    BufferedReader in = new BufferedReader(new FileReader("test.txt"));  
    String text = "";  
    while (true) {  
        String rad = in.readLine();  
        if (rad == null) // är filen slut?  
            break; // ja!  
        text = text + rad;  
    }  
    JOptionPane.showMessageDialog(null, text );  
}
```

ATT SKRIVA TILL EN TEXTFIL

```
Printwriter ut = new Printwriter(  
    new Bufferedwriter(new FileWriter("test.txt")));  
for (int i = 0; i < 10; i++) {  
    ut.println(i);  
}  
ut.close(); // OBS! viktigt
```