

Objektorienterad Programkonstruktion

Övning 3

Stränghantering
Filer

Christian Smith

ccs@kth.se



**KTH Datavetenskap
och kommunikation**

Strängar

- Strängar definieras i klassen String
- Strängar kan instansieras med bokstavliga (eng: literal) strängar, dvs uttryck av typen "detta är en sträng"



**KTH Datavetenskap
och kommunikation**

```
String s1 = new String("Detta fungerar!");
```

```
String s2 = "Detta fungerar också!";
```

```
System.out.println("Detta går också bra!");
```

```
System.out.println(s2.toUpperCase());
```

```
System.out.println(new String("Detta går också!").toUpperCase());
```

```
System.out.println("Detta går faktiskt också!".toUpperCase());
```

Konstruktörer i String



KTH Datavetenskap
och kommunikation

- `String(String str)`
 - skapar en ny sträng med innehållet i `str`
- `String(char[] ca)`
 - skapar en ny sträng utifrån `ca`
- `String(byte[] bytes)`
- `String(byte[] bytes, String charsetName)`
- `String(byte[] bytes, Charset charset)`
 - skapar en ny sträng av `bytes`, enligt teckenkodningen `charset`, default är systemets teckenkodning
- `String(StringBuilder builder)`
 - skapar en ny sträng av en `StringBuilder`

Exempel på metoder i String

```
String substring(int beginIndex, int endIndex)
```

Returnerar den del av strängen som finns mellan indexen (inklusive)

```
char charAt(int index)
```

Returnerar ett tecken från indexet

```
public int compareTo(String str)
```

```
public int compareToIgnoreCase(String str)
```

Gör jämförelse baserad på tecknens värde (alfabetisk)

```
public boolean contains(CharSequence s)
```

returnerar true om s ingår minst en gång i strängen

```
int indexOf(String str, int fromIndex)
```

returnerar index (eller -1) till första förekomst av str, efter fromIndex



KTH Datavetenskap
och kommunikation

Exempel på metoder i String

```
public char[] toCharArray()
```

Returnerar en `char`-array gjord av strängens tecken.



KTH Datavetenskap
och kommunikation

```
public String toString()
```

Gör vad man tror

```
public string toLower()
```

```
public string toUpper()
```

Byter mellan gemener och versaler

```
static String valueOf(boolean b)
```

```
static String valueOf(int i)
```

```
static String valueOf(double d)
```

```
static String valueOf(Object o)
```

returnerar strängen som skulle komma från `toString()`

Övningar med String



KTH Datavetenskap
och kommunikation

- Skriv en metod som tar en String och sorterar om alla tecken i den alfabetiskt, t.ex
"En exempeltext" -> " Eeeeeelnmptttx"
- Skriv en metod som räknar hur många gånger ett visst ord förekommer i en sträng.
- Skriv en metod som plockar bort alla förekomster av ett visst ord ur en sträng

Fler String-metoder

`String replace(String str1, String str2)`
ersätter alla förekomster av `str1` med `str2`

`String[] split(String regex)`
delar strängen vid varje förekomst av `regex`, och returnerar en String-array

`StringAB = StringA + StringB;`
Gör så att `StringAB` blir en sammanslagning av `StringA` och `StringB`



KTH Datavetenskap
och kommunikation

StringBuilder:

`StringBuilder(String initString)`

Skapar ett objekt av typen `StringBuilder`, som mer effektivt kan sätta ihop nya strängar, se `StringBuilderDemo`

`public StringBuilder append(String|int|boolean|etc s)`
lägger till `s` sist i `StringBuilderns` sträng

`public StringBuilder insert(int pos, String|etc s)`
stoppas in `s` vid position `pos` i `StringBuilderns` sträng

`public StringBuilder delete(int startpos, int endpos)`
plockar bort innehåll från `startpos` till `endpos` i `StringBuildern`

Scanner

- Bra när man vill ta sig igenom en text från början till slut, och göra något med alla delar i tur och ordning.
- Kan ta en `String` som indata, men också strömmar, som t.ex filer eller text som skrivs på en kommandorad.

ex för att läsa från fil:

```
try{
    Scanner sc = new Scanner(new File("infil.text"));
}
catch(FileNotFoundException e){
    System.out.println("File not found!");
}
```



**KTH Datavetenskap
och kommunikation**

Scanner

`boolean hasNext() | hasNextInt() | hasNextLine() mm`
returnerar true om det finns något kvar att scanna



**KTH Datavetenskap
och kommunikation**

`String next(String pattern)`
returnerar nästa `String` som passar mönstret, och flyttar
Scannern till första positionen efter denna

`String nextLine()`
returnerar allt fram till nästa radbrytning, och flyttar fram
Scannerns position till nästa rad

`useDelimiter(String pattern)`
Definierar vad som ska anses vara ett mellanrum mellan
två tokens

Regexp (Reguljära uttryck)



KTH Datavetenskap
och kommunikation

- X - tecknet X
- X^* - en sekvens av 0 eller fler X
- $XX(X^*)$ - en sekvens av 2 eller fler X
- $[ABC]$ - ett A , ett B eller ett C
- $\backslash s$ - ett mellanrum (tab, mellanslag, mm)
- $a|b$ - antingen ett a eller ett b
- $a^*|b^*$ - en sekvens med 0 eller fler a :n, alternativt 0 eller fler b :n
- $[0-9A-F]^*$ - en sekvens med 0 eller fler siffror eller någon bokstav A till F
- $(a[ab]^*)| \backslash "$ - antingen en sekvens som börjar med ett a , fortsätter med 0 eller fler a :n eller b :n, eller ett ensamt citattecken

Scannerexempel

```
String s = "<detta>är texten vi vill<scanna>";
```

```
Scanner sc =
```

```
    new Scanner(s).useDelimiter("[<>\\s*]");
```

```
while (sc.hasNext()) {
```

```
    System.out.println(sc.next());
```

```
}
```

```
sc.close();
```



**KTH Datavetenskap
och kommunikation**

Att skriva till fil, snabbkurs

- Se `FileWriterDemo.java`



**KTH Datavetenskap
och kommunikation**

HTML

- Minimalt giltigt HTML-dokument



**KTH Datavetenskap
och kommunikation**

```
<!DOCTYPE html>
<html>
  <head>
    <title>Hello World</title>
  </head>
  <body>
    <p>Hello world!</p>
  </body>
</html>
```

HTML i Java Swing



KTH Datavetenskap
och kommunikation

- Vissa av de Swing-komponenter som kan visa text, kan också visa HTML, som t.ex `JEditorPane`, `JButton` eller `JLabel`
- Detta gör att man kan formatera texten i dessa ganska fritt, och det är lätt att inkludera bilder, tabeller, mm
- Om man vill använda HTML i texten i en Swingkomponent måste texten i `String`-objektet som man vill skriva ut inneslutas av tagg-paret `<html>` `</html>`
- I `JEditorPane` behöver man inte `<html>`-taggarna om man angett att dokumentmodellen ska vara `"text/html"`

HTML i Swing - ett Demo

```
import javax.swing.*;

public class HTMLDemo{
    public static void main(String[] args){
        JFrame myFrame = new JFrame();
        JLabel myLabel =
            new JLabel("<html>"
                + "<font color=\"red\"> RED </font>"
                + "<font color=\"blue\"> BLUE </font>"
                + "</html>");
        myFrame.add(myLabel);
        myFrame.pack();
        myFrame.setVisible(true);
    }
}
```



**KTH Datavetenskap
och kommunikation**

Olika Swingklasser för text



KTH Datavetenskap
och kommunikation

- JLabel - för text användaren inte kan ändra
- JTextField - för en rad som användaren kan ändra
- JPasswordField - som ovan, men texten är dold
- JFormattedTextField - som JTextField, men med automatisk formatering av t.ex datum, valuta, mm
- JTextArea - för inmatning med flera rader text
- JEditorPane - för text med avancerad formatering, t.ex HTML

Textinmatning



KTH Datavetenskap
och kommunikation

- `TextField` och `PasswordField` kan skicka `ActionEvent` när man trycker på retur. `ActionEvent:ets` `getActionCommand()` returnerar texten som har matats in
- I en `TextArea` vill man (oftast) inte skicka iväg texten så fort man tryckt på retur. Då kan det vara lämpligt att ha en separat knapp för att skicka. Lämpligtvis hämtar man då texten med hjälp av metoden `getText()`
- I en `TextArea` kan man behöva ställa in beteendet som man vill ha det innan det blir fint. t.ex

```
setPreferredSize(Dimension d)  
setFont(Font f)  
setLineWrap(boolean b)  
setWrapStyleWord(true)
```

JScrollPane

- Innehåller ett annat Swing-objekt, och scrollbars för att titta på olika delar



KTH Datavetenskap
och kommunikation

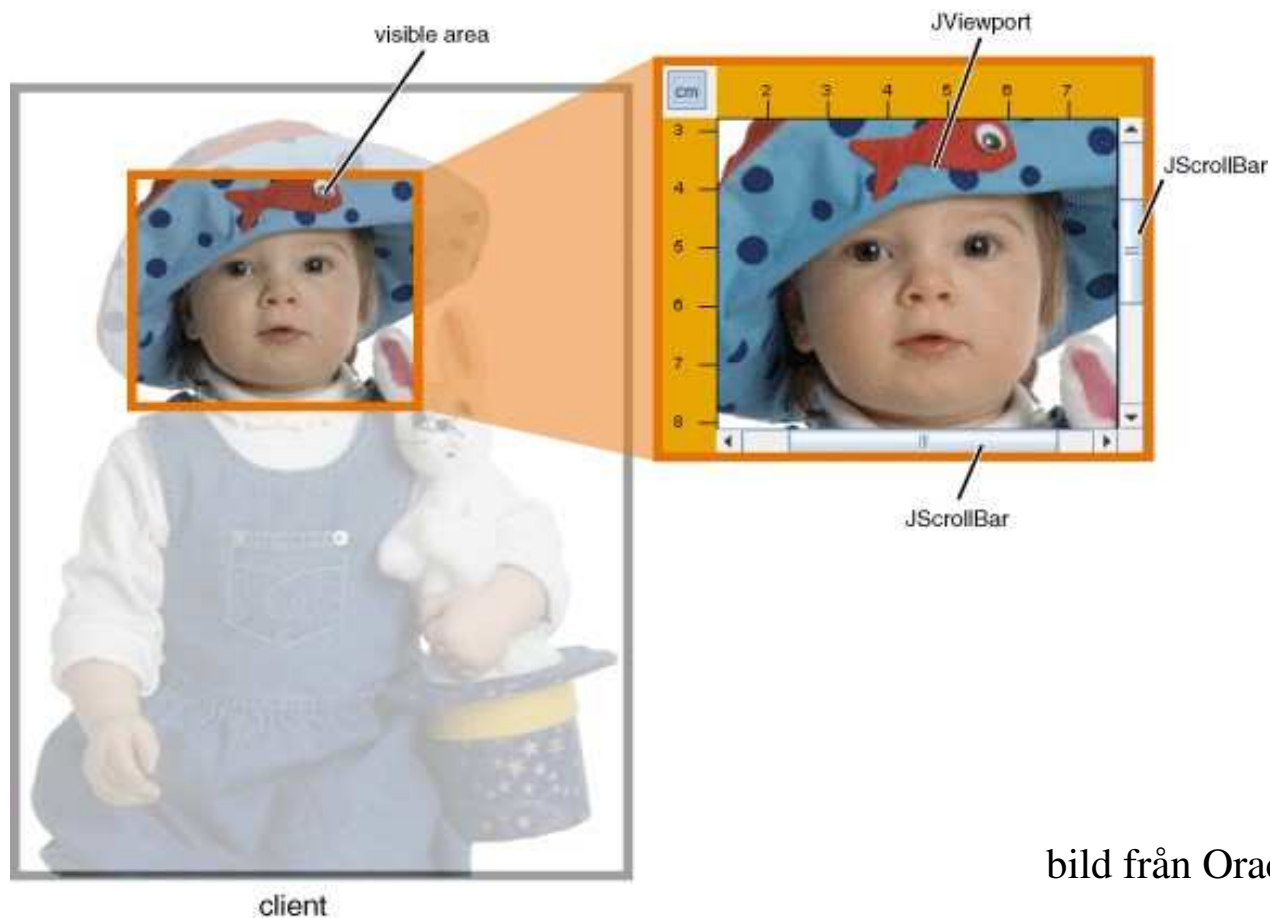


bild från Oracles tutorial

Några demoprogram



**KTH Datavetenskap
och kommunikation**