

## 14.1 Internationale programmer

Når et program skal anvendes af flere kulturer og sprog opstår behov for at programtekster, beløb og datoangivelser afhænger af landet og sproget, og man må i gang med at internationalisere og lokaliseret programmet.

*Internationalisering* består i at gøre programmet sprogneutralt, ved at sørge for at al formatering og fortolkning af tal-, beløbs-, dato- og tidsangivelser sker afhængigt af sproget, og at al sproglig tekst er flyttet til resursefiler.

*Lokalisering* består i at oversætte resursefilerne til et bestemt sprog.

### 14.1.1 Formatering af tal- og tidsangivelser

DateFormat formaterer Date-objekter til strenge (og den anden vej).

```
import java.text.*;
import java.util.*;
public class BenytDateFormat
{
    public static void main(String arg[])
    {
        DateFormat klformat, datoformat, dkf;
        klformat = DateFormat.getTimeInstance(DateFormat.FULL);
        datoformat = DateFormat.getDateInstance(DateFormat.LONG);
        dkf = DateFormat.getDateTimeInstance(DateFormat.MEDIUM,DateFormat.SHORT);

        Date tid = new Date();
        System.out.println( "Kl    :"+ klformat.format(tid) );
        System.out.println( "Dato  :"+ datoformat.format(tid) );
        System.out.println( "Tid   :"+ dkf.format(tid) );
    }
}
Kl    : 23:57:34 CEST
Dato  : 25. juni 2001
Tid   : 25-06-2001 23:57
```

På samme måde formateres/fortolkes tal ved at bede om et formateringsobjekt der klarer netop denne form for tal:

```
import java.text.*;
public class BenytNumberFormat {
    public static void main(String arg[]) {
        NumberFormat fmt1 = NumberFormat.getInstance();
        NumberFormat fmt2 = NumberFormat.getCurrencyInstance();
        NumberFormat fmt3 = NumberFormat.getPercentInstance();
        double tal = 1234.5678;
        System.out.println( fmt1.format(tal) );
        System.out.println( fmt2.format(tal) );
        System.out.println( fmt3.format(tal) );
    }
}
1.234,568
kr 1.234,57
123.457%
```

Ovenstående programmer er kørt med danske sprogindstillinger. Med amerikanske sprogindstillinger bliver uddata i stedet:

```
1,234.568
$1,234.57
123,457%
```

## 14.1.2 Tekstindhold i resursefiler

Programmet `BenytDateFormat` er allerede halvt internationaliseret, idet tidsangivelserne er sprogspecifikke. Lad os nu fuldbyrde programmet ved at lægge tekstindholdet ud i en resursefil.

Det gøres således:

```
import java.text.*;
import java.util.*;
public class BenytDateFormatMedResurser
{
    public static void main(String arg[])
    {
        ResourceBundle res = ResourceBundle.getBundle("Tekster");
        DateFormat klformat, datoformat, dkf;
        klformat = DateFormat.getTimeInstance(DateFormat.FULL);
        datoformat = DateFormat.getDateInstance(DateFormat.LONG);
        dkf = DateFormat.getDateTimeInstance(DateFormat.MEDIUM,DateFormat.SHORT);

        Date tid = new Date();
        System.out.println( res.getString("Kl_")+ klformat.format(tid) );
        System.out.println( res.getString("Dato_")+ datoformat.format(tid) );
        System.out.println( res.getString("Tid_")+ dkf.format(tid) );
    }
}

Kl   : 23:57:34 CEST
Dato : 25. juni 2001
Tid  : 25-06-2001 23:57
```

Filen `Tekster.properties` indeholder de oprindelige tekster i nøgle-værdi-par. Disse vil blive brugt hvis der ikke findes resursefiler for det pågældende sprog:

```
Tid_=Tid \:
Kl_=Kl   \:
Dato_=Dato \:
```

Lad os nu lokalisere programmet til engelsk. Det består i at vi opretter `Tekster_en.properties` med de engelske tekster:

```
Tid_=Time \:
Kl_=Time of day\:
Dato_=Date \:
```

Starter vi derefter programmet med engelske sprogindstillinger (fra UNIX med kommandoen `LC_CTYPE=en java BenytDateFormatMedResurser`) får vi uddata:

```
Time of day: 12:43:02 AM CET
Date       : June 25, 2001
Time       : Jun 25, 2001 12:43 AM
```

Når programmet skal finde en programtekst ud fra sprogindstillingerne sker det ved at

### Binære resursefiler

I stedet for at gemme teksterne i tekstfiler kan de også som binære `.class`-filer (oversatte `.java`-filer). Hvis de findes, vil de binære resursefiler blive brugt fremfor tekstfilerne. Man skal ikke rette

Binære resursefiler laves ved at arve fra `ListResourceBundle` og angive nøgle-værdi-parrene i et array. Her er `Tekster`-resurserne igen, men nu i javafilen `Tekster.java`

```
import java.util.*;
public class Tekster extends ListResourceBundle {
    static final Object[][] contents = new String[][]{
        { "Kl_", "Kl : " },
        { "Dato_", "Dato: " },
        { "Tid_", "Tid : " } };
    public Object[][] getContents() {
        return contents;
    }
}
```

Her er Tekster\_en.java:

```
import java.util.*;
public class Tekster_en extends java.util.ListResourceBundle {
    static final Object[][] contents = new String[][]{
        { "Kl_", "Time of day : " },
        { "Dato_", "Date : " },
        { "Tid_", "Time : " }};
    public Object[][] getContents() {
        return contents;
    }
}
```

### 14.1.3 Avanceret tekstformattering

Det er klart at man ikke kommer langt med internationalisering af sit program hvis man ikke i resursefilerne kan angive tekster med parametre. I disse parametre kan være tal- og tidsangivelser af forskellig form.

Til dette bruges MessageFormat-klassen. MessageFormat tager en formatteringsstreng og et array af argumenter og returnerer en streng med argumenterne formatteret.

Herunder bruger vi det samme sæt argumenter og prøver med forskellige formatteringsstreng.

```
import java.text.*;
import java.util.*;

public class BenytMessageFormat {
    public static void main(String[] args) {

        //
        Object[] a = { new Date(), "A", new Double(0.75), new Integer(2), new Integer(5) };
        String[] s = {
            "Klokken {0,time} d. {0,date} var der {4} tilhørere i sal {1}.",
            "Klokken {0,time,short} d. {0,date,long}.",
            "Klokken {0,time,h:mm a} d. {0,date,short}.",
            "Samlet pris: {2,number,currency}.",
            "Avance: {2,number,percent}.",
            "Der kom {2,choice,0#ikke en sjæl!|1#en person|2#to personer|2<{2} personer}.",
            "Der kom {3,choice,0#ikke en sjæl!|1#en person|2#to personer|2<{3} personer}.",
            "Der kom {4,choice,0#ikke en sjæl!|1#en person|2#to personer|2<{4} personer}.",
        };

        for (int i=0; i<s.length; i++)
            System.out.println( MessageFormat.format(s[i],a) );
    }
}
```

```
Klokken 11:08:03 d. 31-01-1980 var der 5 tilhørere i sal A.
Klokken 11:08 d. 31. januar 1980.
Klokken 11:08 AM d. 31-01-80.
Samlet pris: kr 0,75.
Avance: 75%.
Der kom ikke en sjæl!.
Der kom to personer.
Der kom 5 personer.
```

MessageFormat forstår følgende syntaks (EBNF?? -diagram):

```
messageFormatPattern := string ( "{" messageFormatElement "}" string )*
messageFormatElement := argument { "," elementFormat }
elementFormat := "time" { "," dateTimeStyle }
                  | "date" { "," dateTimeStyle }
                  | "number" { "," numberStyle }
                  | "choice" { "," choiceStyle }
dateTimeStyle := "short"
                | "medium"
```

```

        "long"
        "full"
        dateFormatPattern

numberStyle := "currency"
            "percent"
            "integer"
            numberFormatPattern

choiceStyle := choiceFormatPattern

```

MessageFormat anvender NumberFormat, DateFormat og ChoiceFormat til at hjælpe med formateringen.

ChoiceFormat afgører ud fra et tal hvilken del af dens formateringsstreng der skal vises (kan fornemmes af ovenstående eksempel).

#### 14.1.4 Selv bestemme sproget

Land og sprog repræsenteres ved et Locale-objekt.

Normalt fungerer et program kun på eet sprog af gangen, og så er det ikke nødvendigt at bekymre sig om Locale-objektet.

Ved opstart af et program opretter Java et Locale-objekt, der svarer til sprogindstillingerne på maskinen (på mit UNIX-system ud fra miljøvariablen LC\_CTYPE).

Dette Locale-objekt bruges til al formatering og lignende. Det kan fås med Locale.getDefault(). Ønsker man at ændre standardsproget gøres det med Locale.setDefault(Locale nyt).

Ønsker man mere kontrol over sproget kan man oprette et Locale-objekt selv: Første parameter i konstruktøren er sprogkoden (f.eks. da, sv, no, en, fr), anden parameter landekode (f.eks. DK, GB, DE, FR, en tom streng "" kan også bruges).

Dette objekt bruges til fremskafning af formateringsobjekter til det pågældende sprog:

```

import java.text.*;
import java.util.*;
public class BenytDateFormat2
{
    public static void main(String arg[])
    {
        DateFormat klformat, datoformat;

        Locale fransk = new Locale("fr","FR");
        klformat = DateFormat.getTimeInstance(DateFormat.SHORT,fransk);
        datoformat = DateFormat.getDateInstance(DateFormat.LONG,fransk);

        Date tid = new Date();
        System.out.println( "Kl: " + klformat.format(tid) );
        System.out.println( "Dato: " + datoformat.format(tid) );
    }
}

```

```

Kl: 14:34
Dato: 15 juillet 2001

```

## 14.2 Status for lokalisering af Java

### 14.2.1 Tids- og talformater

Java understøtter dansk fuldt ud.

### 14.2.2 Systemmeddelelser

Java mangler resursefiler til dansk.

## Vindues-systemet

```
størrelse
(byte)
3964 nov 25 01:05 ./java/awt/resources/awt.properties
4436 nov 25 01:05 ./java/awt/resources/awt_de.properties
4728 nov 25 01:05 ./java/awt/resources/awt_es.properties
4788 nov 25 01:05 ./java/awt/resources/awt_fr.properties
4516 nov 25 01:05 ./java/awt/resources/awt_it.properties
4293 nov 25 01:05 ./java/awt/resources/awt_ja.properties
4242 nov 25 01:05 ./java/awt/resources/awt_ko.properties
4222 nov 25 01:05 ./java/awt/resources/awt_sv.properties
4022 nov 25 01:05 ./java/awt/resources/awt_zh_CN.properties
4015 nov 25 01:05 ./java/awt/resources/awt_zh_TW.properties
```

Udklip af den svenske fil:

```
# @(#)awt.properties 1.9 99/10/01 1.9, 10/01/99
```

```
#
# AWT-specific properties
#
```

```
# Modifier names
AWT.shift=Skift
AWT.control=Ctrl
AWT.alt=Alt
AWT.meta=Meta
AWT.altGraph=Alt Graph
```

```
# Key names
AWT.enter=Enter
AWT.backSpace=Backsteg
AWT.tab=Tabb
AWT.cancel=Avbryt
AWT.clear=Ta bort
AWT.pause=Paus
AWT.capsLock=Caps Lock
AWT.escape=Esc
AWT.space=Mellanslag
AWT.pgup=Page Up
AWT.pgdn=Page Down
AWT.end=End
AWT.home=Home
AWT.left=V\u00e4nsterpil
AWT.up=Upp\u00e5tpil
AWT.right=H\u00f6gerpil
AWT.down=Ned\u00e5tpil
```

## JAR-værktøjet (minder om ZIP)

```
2647 nov 25 01:05 ./sun/tools/jar/resources/jar.properties
2987 nov 25 01:05 ./sun/tools/jar/resources/jar_de.properties
2936 nov 25 01:05 ./sun/tools/jar/resources/jar_es.properties
3206 nov 25 01:05 ./sun/tools/jar/resources/jar_fr.properties
2887 nov 25 01:05 ./sun/tools/jar/resources/jar_it.properties
7107 nov 25 01:05 ./sun/tools/jar/resources/jar_ja.properties
4747 nov 25 01:05 ./sun/tools/jar/resources/jar_ko.properties
2886 nov 25 01:05 ./sun/tools/jar/resources/jar_sv.properties
3798 nov 25 01:05 ./sun/tools/jar/resources/jar_zh_CN.properties
3524 nov 25 01:05 ./sun/tools/jar/resources/jar_zh_TW.properties
```

Udklip af den svenske fil:

```
error.cant.open=\
    kan inte \u00f6ppna: {0}
error.illegal.option=\
    Ogiltigt alternativ: {0}
error.bad.option=\
    Ett av alternativen -{ctxu} m\u00e5 anges.
error.bad.cflag=\
    'c'-flaggan kr\u00e4ver att indatafilerna anges!
error.bad.uflag=\
    'u'-flaggan kr\u00e4ver att manifest- eller indatafilerna anges!
error.nosuch.fileordir=\
    {0} : Det finns ingen s\u00e5dan fil eller katalog.
error.write.file=\
    Det uppstod ett fel vid skrivning till befintlig jar-fil.
error.create.dir=\
    {0} : Det gick inte att skapa n\u00e5gon katalog.
error.incorrect.length=\
    ogiltig l\u00e5ngd vid bearbetning: {0}
out.added.manifest=\
    extra manifestfil
```

```
out.update.manifest=\
  uppdaterad manifestfil
out.ignore.entry=\
  ignoreerar posten {0}
```

## Printerdialoger

```
3604 nov 25 01:05 ./sun/awt/print/resources/pdialog.properties
3710 nov 25 01:05 ./sun/awt/print/resources/pdialog_de.properties
3756 nov 25 01:05 ./sun/awt/print/resources/pdialog_es.properties
4011 nov 25 01:05 ./sun/awt/print/resources/pdialog_fr.properties
3596 nov 25 01:05 ./sun/awt/print/resources/pdialog_it.properties
5629 nov 25 01:05 ./sun/awt/print/resources/pdialog_ja.properties
4944 nov 25 01:05 ./sun/awt/print/resources/pdialog_ko.properties
3694 nov 25 01:05 ./sun/awt/print/resources/pdialog_sv.properties

#
# @(#)pdialog_sv.properties      1.4 01/03/06
#
button.cancel=Avbryt
button.help=Hj\u00e4lp
button.ok=OK
button.print=Skriv ut
button.print.mnemonic=P
button.save=Spara...
button.save.mnemonic=V
checkbox.collate=Sortera
checkbox.collate.mnemonic=L
checkbox.duplex=Duplex
checkbox.duplex.mnemonic=d. \td
checkbox.titlesheet=Titelsida:
checkbox.titlesheet.mnemonic=t \td
checkbox.tumble=Tumble
checkbox.tumble.mnemonic=U
combobox.printers=[skapa lista \u00f6ver skrivare]
dialog.title=Skriv ut
label.color=F\u00e4rg:
label.commands=Kommandon:
label.commands.mnemonic=M
label.copies=Antal kopior:
label.copies.mnemonic=c. \td
label.landscape=Liggande
label.landscape.mnemonic=n\td
label.options=Alternativ:
label.orientation=Riktning:
label.pages=Sidor:
label.paper=Papper:
```

## Fil-vælger

```
1336 nov 25 01:05 ./com/sun/java/swing/plaf/motif/resources/motif.properties
1383 nov 25 01:05 ./com/sun/java/swing/plaf/motif/resources/motif_sv.properties
1148 nov 25 01:05 ./com/sun/java/swing/plaf/windows/resources/windows.properties
1158 nov 25 01:05 ./com/sun/java/swing/plaf/windows/resources/windows_sv.properties
1346 nov 25 01:05 ./com/sun/java/swing/plaf/windows/resources/windows_zh_CN.properties
1345 nov 25 01:05 ./com/sun/java/swing/plaf/windows/resources/windows_zh_TW.properties
193 nov 25 01:05 ./com/sun/jndi/cosnaming/jndiprovider.properties
943 nov 25 01:05 ./com/sun/jndi/ldap/jndiprovider.properties
2170 nov 25 01:05 ./javax/accessibility/resources/accessibility.properties
2443 nov 25 01:05 ./javax/accessibility/resources/accessibility_de.properties
2598 nov 25 01:05 ./javax/accessibility/resources/accessibility_es.properties
2785 nov 25 01:05 ./javax/accessibility/resources/accessibility_fr.properties
2538 nov 25 01:05 ./javax/accessibility/resources/accessibility_it.properties
3802 nov 25 01:05 ./javax/accessibility/resources/accessibility_ja.properties
3410 nov 25 01:05 ./javax/accessibility/resources/accessibility_ko.properties
2316 nov 25 01:05 ./javax/accessibility/resources/accessibility_sv.properties
2852 nov 25 01:05 ./javax/accessibility/resources/accessibility_zh_CN.properties
3043 nov 25 01:05 ./javax/accessibility/resources/accessibility_zh_TW.properties
3322 nov 25 01:05 ./javax/swing/plaf/basic/resources/basic.properties
3630 nov 25 01:05 ./javax/swing/plaf/basic/resources/basic_de.properties
3556 nov 25 01:05 ./javax/swing/plaf/basic/resources/basic_es.properties
3668 nov 25 01:05 ./javax/swing/plaf/basic/resources/basic_fr.properties
3488 nov 25 01:05 ./javax/swing/plaf/basic/resources/basic_it.properties
4438 nov 25 01:05 ./javax/swing/plaf/basic/resources/basic_ja.properties
3998 nov 25 01:05 ./javax/swing/plaf/basic/resources/basic_ko.properties
3450 nov 25 01:05 ./javax/swing/plaf/basic/resources/basic_sv.properties
3828 nov 25 01:05 ./javax/swing/plaf/basic/resources/basic_zh_CN.properties
3989 nov 25 01:05 ./javax/swing/plaf/basic/resources/basic_zh_TW.properties
1379 nov 25 01:05 ./javax/swing/plaf/metal/resources/metal.properties
1435 nov 25 01:05 ./javax/swing/plaf/metal/resources/metal_de.properties
1430 nov 25 01:05 ./javax/swing/plaf/metal/resources/metal_es.properties
1476 nov 25 01:05 ./javax/swing/plaf/metal/resources/metal_fr.properties
1424 nov 25 01:05 ./javax/swing/plaf/metal/resources/metal_it.properties
1639 nov 25 01:05 ./javax/swing/plaf/metal/resources/metal_ja.properties
1548 nov 25 01:05 ./javax/swing/plaf/metal/resources/metal_ko.properties
1396 nov 25 01:05 ./javax/swing/plaf/metal/resources/metal_sv.properties
1638 nov 25 01:05 ./javax/swing/plaf/metal/resources/metal_zh_CN.properties
1586 nov 25 01:05 ./javax/swing/plaf/metal/resources/metal_zh_TW.properties
```