

Különböző nyelvekhez igazítás Internationalization - i18n

Java tutorial

Copyright © 2000-2002, Kozsik Tamás

Ha a szoftvert több országban is el akarjuk adni

- Helyi igényekhez igazítás
 - A felhasználó anyanyelvének megfelelően
 - Formátumokra vonatkozó helyi szabályok
 - Dátumok, számok, pénz
- Internacionalizáció
 - Jó, ha a programunkban el tudjuk választani ezeket a részleteket a lényegtől
 - A Java segít ebben
 - Formátumok, hibaiüzenetek, címkék

Ugyanaz a program két nyelven



Locale

- Egy nyelvi/területi egység
- java.util.Locale
- Nyelv
- Ország
- Változat/nyelvjárás

_hu_HU _en_US _fr_FR _fr_CA

Locale-ok beszerzése

- Létrehozhatjuk a nekünk kellőt:
`Locale locale = new Locale("en", "US");`
- Lekérhetjük az alapértelmezettet
`Locale locale = Locale.getDefault();`
- Lekérhetjük a rendelkezésre állókat
`Locale locales[] =
 Locale.getAvailableLocales();`

Különböző nyelvű szövegek

- Címkek, hibaüzenetek, stb.
- Különböző változatok: ResourceBundle
 - ListResourceBundle
 - forrásszövegben
 - PropertyResourceBundle
 - tulajdonságleíró fájlban
- Megkülönböztetés: Locale-okon keresztül

ListResourceBundle

```
import java.util.*;
public class Feliratok
extends ListResourceBundle {
    static final Object[][] contents =
        { { "date", "Időpont: " },
          { "price", "Jegyár: " },
          { "title", "Koncertek" }
        };
    public Object[][] getContents()
    { return contents; }
}
```

ListResourceBundle

```
import java.util.*;
public class Feliratok_en
extends ListResourceBundle {
    static final Object[][] contents =
        { { "date", "Date: " },
          { "price", "Price: " },
          { "title", "Concerts" }
        };
    public Object[][] getContents()
    { return contents; }
}
```

Hogyan használjuk?

```
import java.util.*;
class Feliratok_Test {
    public static void main(String args[]){
        ResourceBundle rb1 =
            ResourceBundle.getBundle("Feliratok",
                                    Locale.getDefault() );
        ResourceBundle rb2 =
            ResourceBundle.getBundle("Feliratok",
                                    new Locale("en","US") );
        System.out.println(rb1.getString("title"));
        System.out.println(rb2.getString("title"));
    }
}
```

A Locale-nak megfelelő változat

```
ResourceBundle rb =
    ResourceBundle.getBundle("Feliratok",
                            new Locale("en","US") );
System.out.println(rb.getString("title"));
```

- Két változatot készítettünk, egy *angolt* és egy *alapértelmezett*: **Feliratok_en** és **Feliratok**
- Keresés: a specifikusabbtól az általánosabb felé
 - Feliratok_en_US
 - Feliratok_en
 - Feliratok

Java tutorial

Copyright © 2000-2002, Kozsik Tamás

Feladat

- Írj programot, ami a bekéri egy körnek a sugarát, és kiszámolja a területét. A program két különböző nyelven tudjon kommunikálni a felhasználóval!

Property ResourceBundle

- Nem forrásfájlban, hanem erőforrás-leíró fájlban vannak a különböző változatok.

```
date=Időpont:                feliratok.properties
price=Jegyár:                feliratok.properties
title=Koncertek

date=Date:                   feliratok_en.properties
price=Price:                  feliratok_en.properties
title=Concerts
```

Java tutorial

Copyright © 2000-2002, Kozsik Tamás

Ugyanúgy használjuk!

```
import java.util.*;
class Feliratok_Test {
    public static void main(String args[]){
        ResourceBundle rb1 =
            ResourceBundle.getBundle("feliratok",
                                    Locale.getDefault() );
        ResourceBundle rb2 =
            ResourceBundle.getBundle("feliratok",
                                    new Locale("en", "US") );
        System.out.println(rb1.getString("title"));
        System.out.println(rb2.getString("title"));
    }
}
```

Feladat

- Alakítsuk át az előző feladatra adott megoldást property-fájlosra!

Locale-specifikus formátumok

- NumberFormat
- DateFormat
- MessageFormat
- A java.text csomag osztályai

Dátum-formátum

- 1998. június 3.
 - Magyarország: 98/6/3
 - USA: 6/3/98
 - Anglia: 3/6/98
 - Finnország: 03.06.98

Java tutorial

Copyright © 2000-2002, Kozsik Tamás

Dátumformátumok kezelése

```
DateFormat dfin = DateFormat.getDateInstance(  
    DateFormat.SHORT,  
    new Locale("en", "US")  
);  
Date date = dfin.parse("6/3/98");  
  
DateFormat dfout = DateFormat.getDateInstance(  
    DateFormat.LONG,  
    Locale.getDefault()  
);  
System.out.println( dfout.format(date) );
```

MessageFormat

- Szövegek felparaméterezésére
- "A {1} fájlrendszer mérete {0} megabájt."

"A /home fájlrendszer mérete 3.3 gigabájt."
"A /tmp fájlrendszer mérete 1.2 gigabájt."

```
String minta =  
    "A {1} fájlrendszer mérete {0} megabájt.";  
Integer méret = ...  
Object[] args = {méret, "/home"};  
String s = MessageFormat.format(minta, args);
```

Összetett példa

```
ResourceBundle rb = ResourceBundle.getBundle(  
    "hibauzenetek", Locale.getDefault());  
Object[] param = {"Alma.java"};  
System.out.println(MessageFormat.format(  
    rb.getString("Mes"), param ));  
  
hibauzenetek_en.properties  
Mes=File {0} cannot be found.  
  
hibauzenetek_hu_HU.properties  
Mes=Nem találom az alábbi fájlt: {0}
```