




Curs practic de Java

Curs 10

Cristian Frăsinaru

`acf@infoiasi.ro`

Facultatea de Informatică
Universitatea "Al. I. Cuza" Iași



Internaționalizarea

Cuprins

- Introducere
- Clasa Locale
- Formatarea numerelor și datelor
- Localizarea mesajelor
- Compararea șirurilor



Introducere



Ce este internaționalizarea ?

- **Internaționalizarea** - pregătirea unei aplicații astfel încât să poată fi personalizată funcție de o regiune geografică, politică sau culturală.
- **Localizarea** - adaptarea unei aplicații internaționalizate la o anumită regiune.

Clasa Locale

Un obiect de tip **Locale**:

- Reprezintă o regiune geografică, politică sau culturală specifică
- Va fi utilizat în operații *regional-sensibile* cum ar fi:
 - formatarea datelor și numerelor
 - afișarea mesajelor

• Crearea

```
Locale(String language)
```

```
Locale(String language, String country)
```

```
Locale(String language, String country, String variant)
```

Determinarea regiunilor

```
import java.util.Locale;

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

        System.out.println("Default locale:");
        localeInfo(Locale.getDefault());

        System.out.println("Available locales:");
        Locale available[] = Locale.getAvailableLocales();
        for(int i=0; i<available.length; i++)
            localeInfo(available[i]);
    }
    static void localeInfo(Locale locale) {
        System.out.println(locale.getDisplayLanguage() + "\n" +
            locale.getDisplayCountry());
    }
}
```

Formatarea și parsarea

Formatarea datelor



```
Locale locale = Locale.FRENCH;

// Format with a custom format
DateFormat formatter = new SimpleDateFormat("HH:mm:ss zzzz", locale);
String s = formatter.format(new Date());
// 21:44:07 Heure d'Europe de l'Est

// Format with a default format
s = DateFormat.getTimeInstance(DateFormat.MEDIUM, locale).
    format(new Date());
System.out.println(s);
// 21:44:07
```



Formatarea numerelor



```
// Regiunea: Canada :)
locale = Locale.CANADA;
string = NumberFormat.getNumberInstance(locale).format(-1234.56);
// -1,234.56

// Regiunea: Romania :(
locale = new Locale("ro", "RO");
string = NumberFormat.getNumberInstance(locale).format(-1234.56);
// -1.234,56

// Formatare pentru regiunea curenta
string = NumberFormat.getNumberInstance().format(-1234.56);
```



Formatarea monedei



```
locale = Locale.CANADA;
string = NumberFormat.getCurrencyInstance(locale).format(123.45);
// $123.45

locale = new Locale("ro", "RO");
string = NumberFormat.getCurrencyInstance(locale).format(123.45);
// 123,45 LEI

// Parsarea unui sir ce reprezinta o suma de bani
try {
    Number number = NumberFormat.
        getCurrencyInstance(locale).parse("$123.45");
    if (number instanceof Long) {
        // Valoare de tip Long
    } else {
        // Valoare de tip Double
    }
} catch (ParseException e) { ... }
```



Localizarea mesajelor

Definirea fișierelor de resurse

MyResources_ro.properties

```
hello=Salut  
bye=La revedere
```

MyResources_fr.properties

```
hello=Bonjour  
bye=Au Revoir
```

MyResources.properties

```
hello=Hello  
bye=Goodbye
```

Mesaje parametrizate:

```
welcome=User {0} logged in at {1} from IP {2}
```

Crearea unei clase cu resurse

```
package demo;
import java.util.*;

public class MyResources_ro extends ListResourceBundle {
    public Object[][] getContents() {
        return contents;
    }
    static final Object[][] contents = {
        {"hello", "Salut"},
        {"bye", "La revedere"}
    };
}
```

Clasa ResourceBundle



```
String baseName = "MyResources";
try {
    // Selectam fisierul cu resurse implicit
    ResourceBundle rb = ResourceBundle.getBundle(baseName);
    String key = "hello";
    String s = rb.getString(key); // Hello
    key = "bye";
    s = rb.getString(key);        // Goodbye

    // Selectam alt fisier cu resurse
    rb = ResourceBundle.getBundle(baseName, Locale.FRENCH);
    key = "hello";
    s = rb.getString(key);        // Bonjour
    key = "bye";
    s = rb.getString(key);        // Au Revoir
} catch (MissingResourceException e) {
    // Fisierul cu resurse sau cheia nu exista
}
```



Compararea șirurilor



```
Locale locale = new Locale("ro", "RO");
Collator collator = Collator.getInstance(locale);
int compare = collator.compare(sir1, sir2);
if (compare < 0) {
    // sir1 < sir2
} else if (compare > 1) {
    // sir1 > sir2
} else {
    // sir1 = sir2
}
```

"ramură" < "rămurică" < "rățușcă" < "repede" < "rîu"

