



# Bài 6

I18N & L10N

# Mục tiêu

- Giới thiệu I18N
- Giới thiệu L10N
- Định nghĩa Locale
- Lớp ResourceBundle
- Định dạng số, tiền tệ, phần trăm
- Định dạng Date và Time
- Định dạng chuỗi với MessageFormat



# I18N

**I18N** là viết tắt của **Internationalization**.

- Với sự phổ biến của Internet, toàn cầu hóa các sản phẩm phần mềm là điều tất yếu.
- Việc sử dụng phần mềm bởi người dùng sử dụng ngôn ngữ khác nhau đòi hỏi không gặp phải rào cản.
- Việc thích ứng quốc tế hóa đòi hỏi phải dễ dàng và không yêu cầu thay đổi mã nguồn.
- Ứng dụng có thể phân phối tới thị trường quốc tế, mọi hoạt động nhập/xuất có thể khác nhau tùy vào vị trí địa lý và văn hóa người dùng.



# I18N

Tại sao lại cần toàn cầu hóa sản phẩm phần mềm nói chung và Java Form nói riêng?

1

- Vì không phải tất cả các quốc gia trên thế giới đều sử dụng ngôn ngữ tiếng Anh.

2

- Biểu tượng tiền tệ của mỗi quốc gia là khác nhau.

3

- Định dạng thời gian, ngày tháng của mỗi quốc gia cũng khác nhau.

4

- Chính tả có sự khác biệt giữa các ngôn ngữ.



# I18N

Có 2 giải pháp cho vấn đề toàn cầu hóa phần mềm:

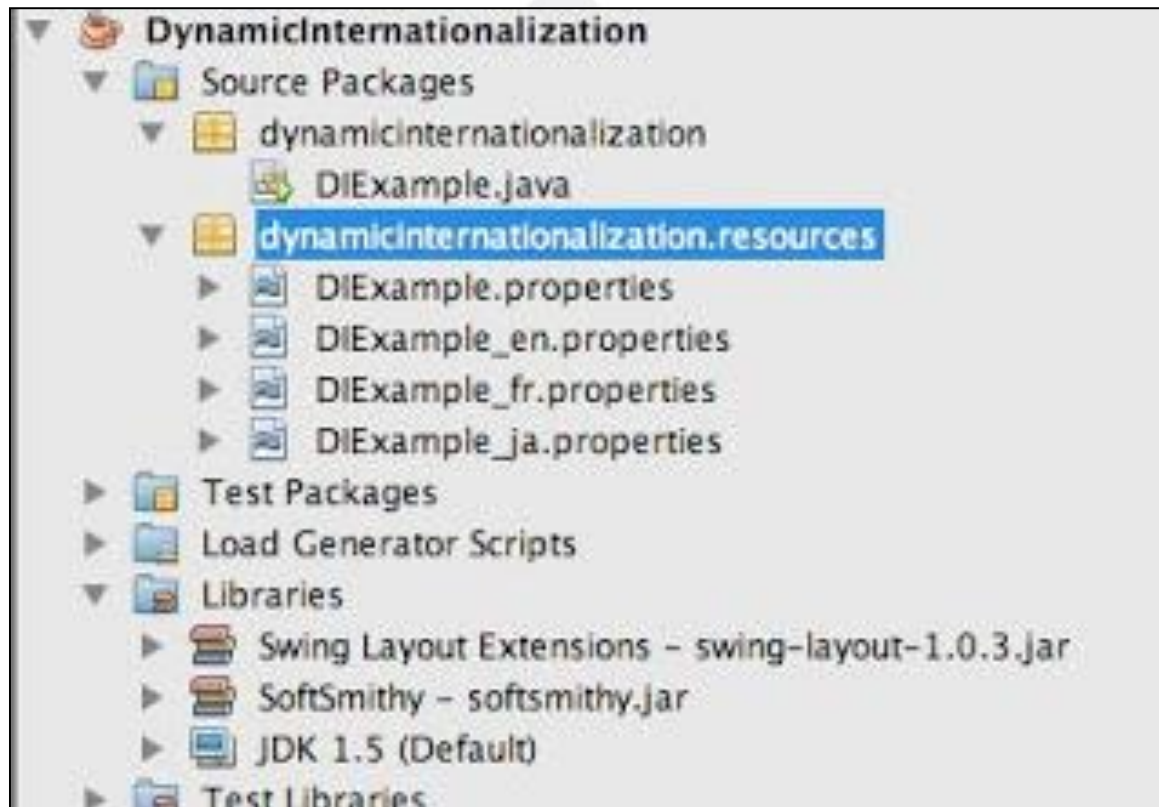
- Phát triển toàn bộ sản phẩm bằng ngôn ngữ mong muốn -> **dẫn tới phát sinh nhân công, thời gian...gia tăng chi phí...**
- Dịch toàn bộ sản phẩm ra ngôn ngữ mong muốn, cung cấp khả năng tương thích với quốc gia, ngôn ngữ khác nhau -> **có thể giảm thiểu các thành phần hard-code trong mã nguồn, chi phí nhân công, tăng tính tiện lợi khi phân phối sản phẩm.**

Do vậy, phần mềm phát triển độc lập với ngôn ngữ của người dùng, sau đó được nội địa hóa theo từng quốc gia, khu vực. Phần mềm có thể điều chỉnh mà không cần bất kỳ sự thay đổi mã nguồn.



# I18N

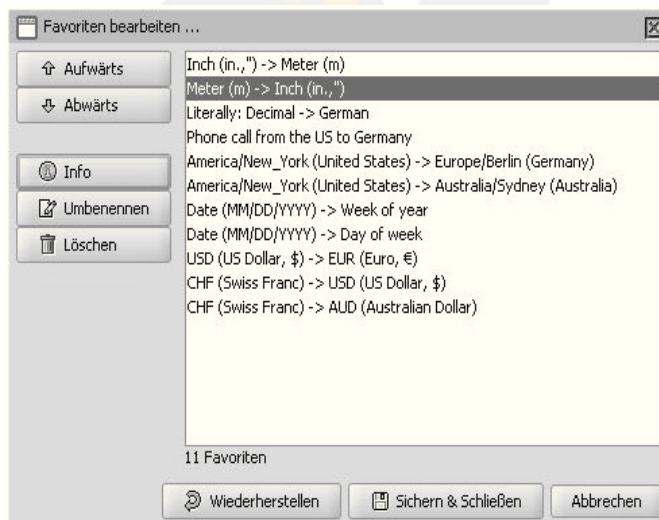
Quốc tế hóa sử dụng kỹ thuật file properties trong Java Swing.



# L10N

## Nội địa hóa là viết tắt của Localization?

- Nội địa hóa đề cập đến **vùng**, **ngôn ngữ** cụ thể.
- Trong nội địa hóa, phần mềm được điều chỉnh tương thích với vùng, ngôn ngữ cụ thể.
- Trong nội địa hóa, các yếu tố giao diện và văn bản được dịch và chuyển đổi tương ứng.



# Locale

- Đối tượng lớp **java.util.Locale** là một định danh cho sự kết hợp giữa **ngôn ngữ** và **khu vực**, nó giúp thể hiện một khu vực địa lý, chính trị, văn hóa cụ thể.
- Sử dụng Locale có thể định kiểu định dạng các thành phần như **số**, **tiền tệ**, **kiểu mẫu câu**...phù hợp với từng địa phương.
- Các constructor của Locale:

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

```
public Locale(String language)
```





# Locale

## Các phương thức trong lớp Locale:

**public static Locale getDefault() :** lấy về Locale mặc định trong Java Virtual Machine.

**Example:** `Locale currentLocale = Locale.getDefault();`

**public final String getDisplayCountry() :** Trả về tên quốc gia trong Locale hiện tại, vd: Locale `fr_FR`, phương thức trả về **France**.

**Example:** `Locale currentLocale = Locale.getDefault();`

`String strCountry = currentLocale.getDisplayCountry();`

**public final String getDisplayLanguage() :** Trả về tên ngôn ngữ. VD: `fr_FR`, phương thức trả về French.

**Example:** `Locale currentLocale = Locale.getDefault();`

`String strLanguage = currentLocale.getDisplayLanguage();`



# Locale

Mã ngôn ngữ - mã quốc gia (vd):

Language	Code
English	En
Arabic	Ar
German	De

Country	Code
United States	US
Canada	CA
France	FR
Japan	JP



# ResourceBundle

- Lớp ResourceBundle sử dụng để lấy thông tin dữ liệu khu vực cụ thể dựa trên file properties.
- Thông tin này cho phép viết ứng dụng cụ thể:

1

- Nội địa hóa hoặc dịch sang ngôn ngữ khác.

2

- Quản lý đa vùng miền – khu vực cùng lúc.

3

- Hỗ trợ nhiều ngôn ngữ sau này.



# ResourceBundle

Code: nội dung các file properties

```
MessagesBundle.properties  
greetings = Hello.  
farewell = Goodbye.  
inquiry = How are you?
```

```
MessagesBundle_de_DE.properties  
greetings = Hallo.  
farewell = Tschüß.  
inquiry = Wiegeht's?
```

```
MessagesBundle_en_US.properties  
greetings = Hello.  
farewell = Goodbye.  
inquiry = How are you?
```

```
MessagesBundle_fr_FR.properties  
greetings = Bonjour.  
farewell = Au revoir.  
inquiry = Comment allez-vous?
```

```
MessagesBundle_ja_JP.properties  
greetings = Ohayōgozaimasu.  
farewell = sayonara/sayounara.  
inquiry = O genkidesuka?
```



# ResourceBundle

Code:

## Code Snippet

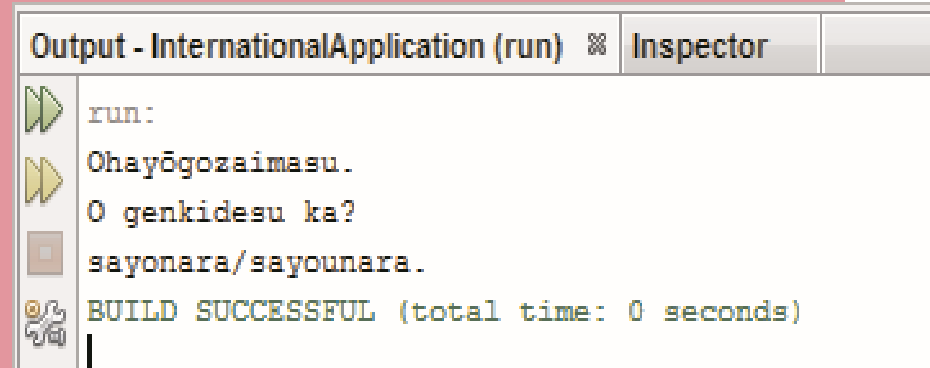
```
import java.util.Locale;
import java.util.ResourceBundle;
public class InternationalApplication {
/**
 * @param args the command line arguments
 */
public static void main(String[] args) {
    // TODO code application logic here
    String language;
    String country;
```



# ResourceBundle

```
if (args.length != 2) {  
    language = new String("en");  
    country = new String("US");  
} else {  
    language = new String(args[0]);  
    country = new String(args[1]);  
}
```

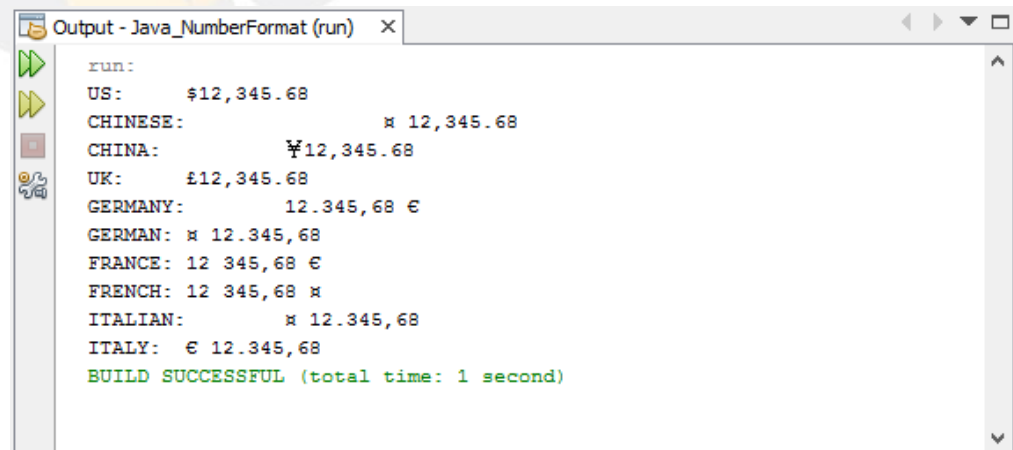
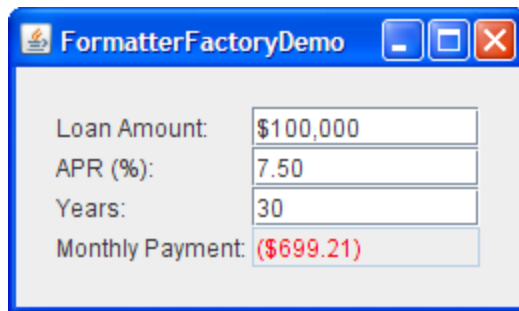
```
    Locale currentLocale;  
    ResourceBundle messages;  
    currentLocale = new Locale(language, country);  
    messages = ResourceBundle.getBundle("internationalApplication/  
MessagesBundle", currentLocale);  
    System.out.println(messages.getString("greetings"));  
    System.out.println(messages.getString("inquiry"));  
    System.out.println(messages.getString("farewell"));  
}
```



# Number – Currencies – Percentage

**NumberFormat** là lớp hỗ trợ định dạng số, tiền tệ, phần trăm. Lớp có các hàm:

- `getNumberInstance()` : định dạng số
- `getCurrencyInstance()` : định dạng tiền tệ
- `getPercentInstance()` : định dạng phần trăm



# Number – Currencies – Percentage

## Code Snippet

```
import java.text.NumberFormat;
import java.util.Locale;
import java.util.ResourceBundle;
public class InternationalApplication {

    static public void printValue(Locale currentLocale) {
        Integer value = new Integer(123456);
        Double amt = new Double(345987.246);
        NumberFormat numFormatObj;
        String valueDisplay;
        String amtDisplay;
```

```
123456      fr_FR
345987,246   fr_FR

123.456      de_DE
345.987,246   de_DE

123,456      en_US
345,987.246   en_US
```





# Number – Currencies – Percentage

```
numFormatObj = NumberFormat.getNumberInstance(currentLocale);
valueDisplay = numFormatObj.format(value);
amtDisplay = numFormatObj.format(amt);
System.out.println(valueDisplay + " " + currentLocale.toString());
System.out.println(amtDisplay + " " + currentLocale.toString());
}

/**
 * @param args the command line arguments
 */
public static void main(String[] args) {
    // TODO code application logic here
    String language;
    String country;
```



# Number – Currencies – Percentage

```
if (args.length != 2) {  
    language = new String("en");  
    country = new String("US");  
} else {  
    language = new String(args[0]);  
    country = new String(args[1]);  
}  
    Locale currentLocale;  
    ResourceBundle messages;  
currentLocale = new Locale(language, country);  
messages =  
ResourceBundle.getBundle("internationalApplication/MessagesBundle",  
currentLocale);
```



# Number – Currencies – Percentage

```
System.out.println(messages.getString("greetings"));
System.out.println(messages.getString("inquiry"));
System.out.println(messages.getString("farewell"));
printValue(currentLocale);
}
```

## Code Snippet

```
NumberFormat currencyFormatter;
String strCurrency;
// Creates a Locale object with language as French and country
// as France
Locale locale = new Locale("fr", "FR");
// Creates an object of a wrapper class Double
Double currency = new Double(123456.78);
```



# Number – Currencies – Percentage

```
// Retrieves the CurrencyFormatterinstance
currencyFormatter = NumberFormat. getInstance(locale);
// Formats the currency
strCurrency = currencyFormatter.format(currency);
messages =
ResourceBundle.getBundle("internationalApplication/MessagesBundle",
currentLocale);
```



# Number – Currencies – Percentage

## Code Snippet

```
NumberFormatpercentFormatter;  
String strPercent;  
// Creates a Localeobject with language as French and country  
// as France  
Locale locale = new Locale("fr", "FR");  
// Creates an object of a wrapper class Double  
Double percent = new Double(123456.78);  
// Retrieves the percentFormatter instance  
percentFormatter = NumberFormat. getPercentInstance(locale);  
// Formats the percent figure  
strPercent = percentFormatter.format(percent);
```



# Date & Time

**DateFormat** là lớp hỗ trợ định dạng ngày tháng tùy thuộc khu vực, ngôn ngữ. Có các định dạng:

- `DateFormat.LONG`,
- `DateFormat.SHORT`, và
- `DateFormat.MEDIUM`

## Code Snippet

```
public static final DateFormat getInstance(int style, Locale locale)
```



# Date & Time

## Demo:

```
import java.text.DateFormat;
import java.util.Date;
import java.util.Locale;

public class DateInternationalApplication {

    public static void main(String[] args) {
        Date today;
        String strDate;
```



# Date & Time

```
DateFormatdateFormatter;  
    Locale locale = new Locale("ja", "JP");  
dateFormatter = DateFormat.getDateInstance(DateFormat.MEDIUM,  
locale);  
today = new Date();  
strDate = dateFormatter.format(today);  
System.out.println(strDate);  
}  
}
```

```
<terminated> sniffet40 [Java Application] C:\Program Files\Java  
Date : Fri Jan 26 00:06:00 IST 2007
```





# MessageFormat

**MessageFormat** là lớp hỗ trợ định dạng mẫu văn bản tùy thuộc khu vực, ngôn ngữ. Demo:

```
import java.text.MessageFormat;
import java.util.Date;
import java.util.Locale;
import java.util.ResourceBundle;
public class MessageFormatterInternationalApplication {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        String template = "At {2,time,short} on {2,date,long}, we detected {1,number,integer} virus on the disk {0}";
```

# MessageFormat

```
MessageFormat formatter = new MessageFormat("");  
String language;  
String country;
```

```
if (args.length != 2) {  
    language = new String("en");  
    country = new String("US");  
} else {  
    language = new String(args[0]);  
    country = new String(args[1]);  
}
```

```
Locale currentLocale;  
currentLocale = new Locale(language, country);  
formatter.setLocale(currentLocale);
```



# MessageFormat

```
ResourceBundle messages =  
ResourceBundle.getBundle("messageformatterin  
ternationalapplication/MessageFormatBundle", currentLocale);  
Object[] messageArguments = {messages.getString("disk"), new Integer(7),  
new Date()};  
formatter.applyPattern(messages.getString("template"));  
String output = formatter.format(messageArguments);  
System.out.println(output);  
  
}  
}
```

On 06/03/2007 we detected 10 viruses

Sur 06/03/2007 nous avons détecté le virus 10

Auf 06/03/2007 ermittelten wir Virus 10



# Tóm tắt bài học

```
Output - JavaApplication (run) #2
run:
1. Japan
2. Vietnam
3. Exit program
Please choose your region: 1
¥10,500 for the locale ja_JP
1. Japan
2. Vietnam
3. Exit program
Please choose your region: 2
10.500 đ for the locale vi_VN
1. Japan
2. Vietnam
3. Exit program
```

Định dạng theo Nhật Bản

Định dạng theo Việt Nam



HẾT  
XIN CẢM ƠN!

