

## CSE 3461

### Ch 10 - Internationalization

p. 208

- Globalization ... involves externalizing any information that is culture or language dependent. Externalizing information moves it out of the code and into resources. After the information is moved into resources, **culture specialists** can localize the interface by modifying those resources rather than modifying the code.

2

### *Internationalization vs Localization vs Globalization*

- Internationalization
  - the process of designing a software application so that it can be adapted to various languages and regions without engineering changes.
- Localization
  - the process of adapting internationalized software for a specific region or language by adding locale-specific components and translating text.
- Globalization = Internationalization + Localization

3

### Culturally Dependent Data

Messages  
Labels on GUI components  
Online help  
Sounds  
Colors  
Graphics  
Icons  
Dates  
Times  
Numbers  
Currencies  
Measurements  
Phone numbers  
Honorifics and personal titles  
Postal addresses  
Page layouts

4

## Mechanisms for Globalization

1. Locale-dependent strings
  - multilingual support
  - sorting
2. Locale-independent string processing
  - string analysis
3. Numbers and Currencies
4. Dates and Times

5

## Locales

- Language + Country
- Eg.
  - French\_Belgian fr\_BE
  - French\_Canadian fr\_CA
  - French\_Swiss fr\_CH
  - French\_Luxembourg fr\_LU
  - French\_Monaco fr\_MC
  - French\_France fr\_FR
- *<locale listing app>*

6

## 1. Locale-Dependent Strings

- I18NSample
- this is an example of “canned” translation

7

## Machine Translation (MT)

- a sub-field of computational linguistics
- translate text or speech from one natural language to another (source/target)
- simplest MT performs substitution of words
- sophisticated MT performs complex translations that:
  - handles differences in linguistic typology
  - performs phrase recognition
  - translates of idioms
- textbook and other sources identify “compound strings” and “compound messages” as a challenging issue
  - at its logical endpoint, this is no less than an open research problem in MT

8

## 2. Locale-Independent String Processing

9

## 2. Locale-Independent String Processing

- use Unicode compliant tests
  - eg., in Java, the services of `Character` class
- not compliant:

```
char ch; ...  
(ch >= 'A' && ch <= 'Z')
```

recall that `char` is an unsigned int, e.g., 'A' is 65
- the `Character.isLetter` method returns true if the character is a letter in Chinese, German, Arabic, or another language.

10

## 2. Locale-Independent String Processing

- Character types
  - can be used with getter `Character.getType(ch)`
    - `Character.LOWERCASE_LETTER`
    - `Character.UPPERCASE_LETTER`
    - `Character.MATH_SYMBOL`
    - `Character.CONNECTOR_PUNCTUATION`

11

## 3. Numbers and Currencies

- culturally dependent:
  - punctuation
  - words and symbols for numbers (in specific writing system)
  - currency symbol (1 char, 3 char), currency name
- culturally independent:
  - use of (Western) Arabic numerals (1234567890) (at least in realm of commerce)
  - decimal base (previously sexagesimal; Sumerians)
- `NumberFormat` provides Locale-specific services for numbers and currencies

12

## 4. Dates and Times

- dates are implicit in time
- culturally dependent:
  - am/pm signifier
  - 12 vs 24 hr format
  - punctuation
  - names of months and days
  - definition of workweek
- culturally independent:
  - need for different formats, such as short, medium, and long formats
- `DateFormat` provides Locale-specific services
- `DateFormattingExample`
- <http://download.oracle.com/javase/tutorial/118n/format/dateintro.html>