



Java 核心技术(进阶)

第三章 高级文本处理

第二节 Java国际化编程

华东师范大学 陈良育



国际化编程

- Internationalization, 缩写为i18n.
- 多语言版本的软件
 - 一套软件，多个语言包
 - 根据语言设定，可以切换显示文本





Java国际化编程

- Java是第一个设计成支持国际化的编程语言
 - `java.util.ResourceBundle` 用于加载一个语言_国家语言包
 - `java.util.Locale` 定义一个语言_国家
 - `java.text.MessageFormat` 用于格式化带占位符的字符串
 - `java.text.NumberFormat` 用于格式化数字/金额
 - `java.text.DateFormat` 用于格式化日期时间
 - `java.time.format.DateTimeFormatter` 用于格式化日期时间
- (后4个Format参见《Java核心技术》第8章)

国际化HelloWorld



```
public class HelloWorld {  
  
    public static void main(String[] args) {  
        System.out.println("Hello World");  
        //System.out.println("你好，世界");  
    }  
}  
  
import java.util.Locale;  
import java.util.ResourceBundle;  
  
public class NewHelloWorld {  
  
    public static void main(String[] args) {  
        // 取得系统默认的国家/语言环境  
        Locale myLocale = Locale.getDefault();  
  
        System.out.println(myLocale); //zh_CN //en_US  
  
        // 根据指定语言_国家环境加载资源文件  
        ResourceBundle bundle = ResourceBundle.getBundle("message", myLocale);  
  
        // 从资源文件中取得的消息  
        System.out.println(bundle.getString("hello"));  
    }  
}
```


Locale类



- Locale(zh_CN, en_US,···)
 - 语言, zh, en等
 - 国家/地区, CN, US等
 - 其他变量(variant)(几乎不用)
- Locale方法
 - getAvailableLocales()返回所有的可用Locale
 - getDefault()返回默认的Locale



语言文件(1)

- 语言文件

- 一个Properties文件 (参见《Java核心技术》第十章)
- 包含K-V对, 每行一个K-V, 例如: age=20
- 命名规则
 - 包名+语言+国家地区.properties, (语言和国家地区可选)
 - message.properties
 - message_zh.properties
 - message_zh_CN.properties



语言文件(2)

- 语言文件
 - 存储文件必须是ASCII码文件
 - 如果是ASCII以外的文字，必须用Unicode的表示\uxxxx
 - 可以采用native2ascii.exe (%JAVA_HOME%\bin目录下)进行转码

a.properties - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

hello=你好, 世界

message_zh_CN.properties - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

hello=\u4f60\u597d, \u4e16\u754c

```
c:\temp>native2ascii a.properties message_zh_CN.properties
```

ResourceBundle类(1)



- ResourceBundle
 - 根据Locale要求, 加载语言文件(Properties文件)
 - 存储语言集合中所有的K-V对
 - getString(String key) 返回所对应的value



ResourceBundle类(2)

- ResourceBundle 根据key找value的查找路径
 - 包名_当前Locale语言_当前Locale国家地区_当前Locale变量(variant)
 - 包名_当前Locale语言_当前Locale国家地区
 - 包名_当前Locale语言
 - 包名_默认Locale语言_默认Locale国家地区_默认Locale变量(variant)
 - 包名_默认Locale语言_默认Locale国家地区
 - 包名_默认Locale语言
 - 包名

其他国际化



- 日期/时间国际化
 - DateTimeFormatter和Locale的结合
- 数字/金额国际化
 - NumberFormat和Locale结合

总结



- Java国际化总结
 - ResourceBundle和Locale类
 - Properties文件的制作和native2ascii的转化

代码(1) LocaleTest.java



```
public class LocaleTest {  
  
    public static void main(String[] args) {  
        // 返回Java所支持的全部国家和语言的数组  
  
        Locale[] localeList = Locale.getAvailableLocales();  
  
        for (Locale locale : localeList)  
        {  
            System.out.println(locale.getLanguage() + "_" + locale.getCountry());  
            System.out.println(locale.getDisplayLanguage() + "_" + locale.getDisplayCountry());  
        }  
  
        System.out.println("=====");  
        Locale myLocale = Locale.getDefault();  
        System.out.println(myLocale); //zh_CN  
        System.out.println(Locale.CHINA); //zh_CN  
  
        myLocale = new Locale("en", "US"); //语言 国家, 强制换成en_US  
        System.out.println(myLocale); //en_US  
    }  
}
```


代码(2) HelloWorld.java



```
public class HelloWorld {  
  
    public static void main(String[] args) {  
        System.out.println("Hello World");  
        //System.out.println("你好, 世界");  
    }  
  
}
```

代码(3) NewHelloWorld.java



```
public class NewHelloWorld {  
  
    public static void main(String[] args) {  
        // 取得系统默认的国家/语言环境  
        Locale myLocale = Locale.getDefault();  
  
        System.out.println(myLocale); //zh_CN  
  
        // 根据指定语言_国家环境加载资源文件  
        ResourceBundle bundle = ResourceBundle.getBundle("message", myLocale);  
  
        // 从资源文件中取得的消息  
        System.out.println(bundle.getString("hello")); //你好, 世界  
  
        myLocale = new Locale("en", "US"); //语言 国家, 强制换成en_US  
        bundle = ResourceBundle.getBundle("message", myLocale);  
        System.out.println(bundle.getString("hello")); //Hello World  
  
    }  
}
```



代码(4) NameTest.java

```
public class NameTest {  
  
    public static void main(String[] args) {  
        Locale myLocale = Locale.getDefault();  
  
        System.out.println(myLocale); //zh_CN  
  
        // 根据指定语言_国家环境加载资源文件  
        ResourceBundle bundle = ResourceBundle.getBundle("msg", myLocale);  
  
        // 从资源文件中取得的消息  
        System.out.println(bundle.getString("name")); //陈良育  
  
    }  
  
}
```

代码(5) message_en_US.properties



```
hello=Hello World
```


代码(6) message_zh_CN.properties



```
hello=\u4f60\u597d, \u4e16\u754c
```

代码(7) msg_zh_CN.properties



```
name=\u9648\u826f\u80b2
```



谢谢!