



# 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(补充)

华东师范大学 陈良育



# ResourceBundle类(1)

- ResourceBundle
  - 根据Locale要求, 加载语言文件(Properties文件), 存储K-V对
    - `Locale myLocale = new Locale("zh", "CN");`
    - `ResourceBundle.getBundle("msg", myLocale);` // 加载msg\_zh\_CN.properties
  - `getString(String key)` 返回所对应的value
    - 如果找不到key, 则报异常



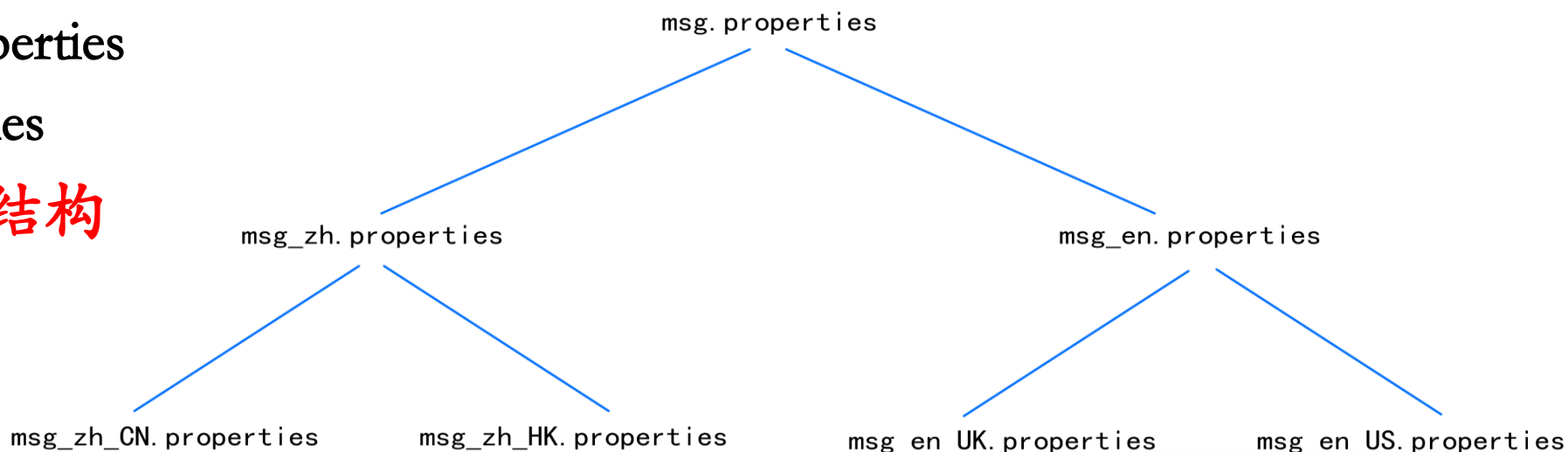
# ResourceBundle类(2)

- 语言文件层级结构

- 文件: **包名\_当前Locale语言\_当前Locale国家地区\_当前Locale变量**  
(variant)

- msg\_zh\_CN.properties (注意大小写)
    - msg\_zh.properties
    - msg.properties

- 文件形成**树结构**





# ResourceBundle类(3)

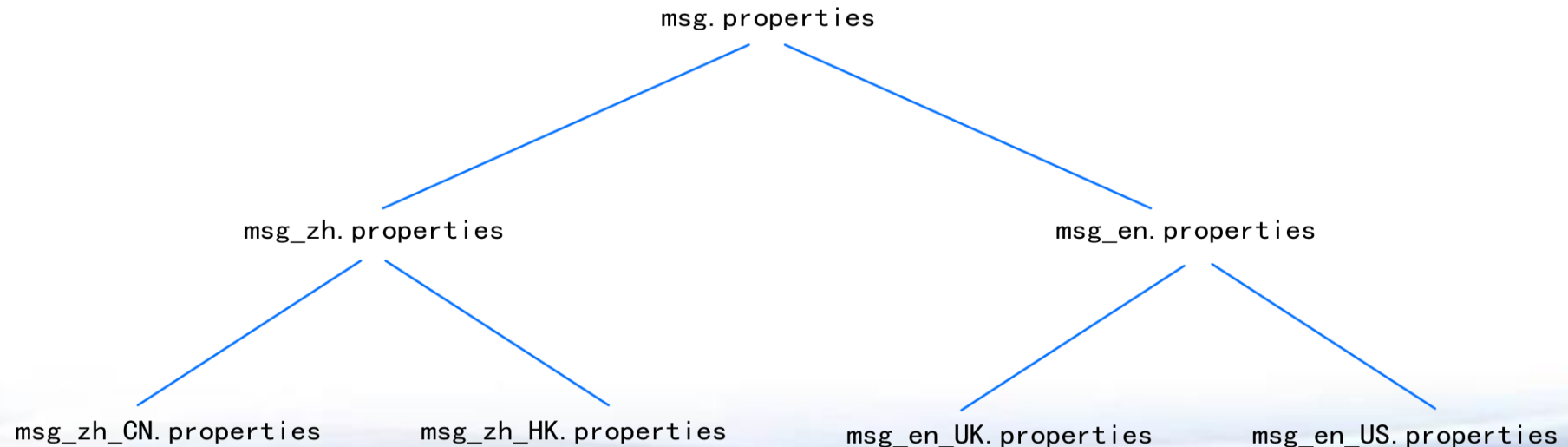
- ResourceBundle 加载文件(根据给定的Locale)
  - 首先, 按如下顺序找到第一个Properties文件
    - 包名\_当前Locale语言\_当前Locale国家地区\_当前Locale变量(variant)
    - 包名\_当前Locale语言\_当前Locale国家地区
    - 包名\_当前Locale语言
    - 包名\_默认Locale语言\_默认Locale国家地区\_默认Locale变量(variant)
    - 包名\_默认Locale语言\_默认Locale国家地区
    - 包名\_默认Locale语言
    - 包名
  - 根据Properties树结构, 把第一步的Properties文件及父级文件全部加载, 作为后续的K-V集合





# ResourceBundle类(4)

- ResourceBundle 加载文件
  - ("zh", "CN")加载msg\_zh\_CN, msg\_zh, msg
  - ("zh", "TW")加载msg\_zh, msg
  - ("ja", "JP") 加载msg\_zh\_CN, msg\_zh, msg(假设系统默认是zh\_CN)





# ResourceBundle类(5)

- ResourceBundle 加载文件
  - ("zh", "CN")加载msg\_zh\_CN, msg\_zh
  - ("zh", "TW")加载msg\_zh
  - ("ja", "JP")加载msg\_zh\_CN, msg\_zh(假设系统默认是zh\_CN)
  - ("en", "NZ")加载msg\_en



# 总结



- ResourceBundle加载文件总结
  - Locale: 是由语言和国家/地区组合起来的
  - 所有的Properties文件形成一棵树
  - 指定Locale后, ResourceBundle就形成一个树
    - 先找到第一个节点文件(搜索路径)
    - 然后将该节点的所有父节点文件全部加载
  - 根据所找到的文件内容和key, 返回相应的值



谢谢!