Lecture 6
Java Internationalization
Part 1

Reference:

Java-Tutorials/tutorial-2015/index.html

Objectives

- How to create a Java program which can be easily customized to run it all over the world?

Contents

1- Internationalization

2- Localization

3- Which contents of a program can be internationalized?

4- A way to Internationalize a Program

5- Basic Java APIs for Internationalization

6- Steps for Internationalizing a Program

7- A Demonstration

1- Internationalization?

- *Internationalization* is the process of designing an application so that it can be adapted to various languages and regions without engineering changes. Sometimes the term internationalization is abbreviated as i18n, because there are 18 letters between the first "i" and the last "n.“
- An internationalized program has the following characteristics:
  - With the addition of localized data, the same executable can run worldwide.
  - Textual elements, such as status messages and the GUI component labels, are not hardcoded in the program. Instead they are stored outside the source code and retrieved dynamically.
  - Support for new languages does not require recompilation.
  - Culturally-dependent data, such as dates and currencies, appear in formats that conform to the end user's region and language.
  - It can be localized quickly.

2- Localization?

- *Localization* is the process of adapting software for a specific region or language by adding locale-specific components and translating text. The term localization is often abbreviated as l10n, because there are 10 letters between the "l" and the "n."
- The primary task of localization is translating the user interface elements and documentation. Localization involves not only changing the language interaction, but also other relevant changes such as display of numbers, dates, currency, and so on. Other types of data, such as sounds and images, may require localization if they are culturally sensitive. The better internationalized an application is, the easier it is to localize it for a particular language and character encoding scheme.

3- Which contents of a program can be internationlized?

GUI Program

GUI

Labels

Inputs from user

Output

Messages in dialogs

Console Program

Guidelines for user

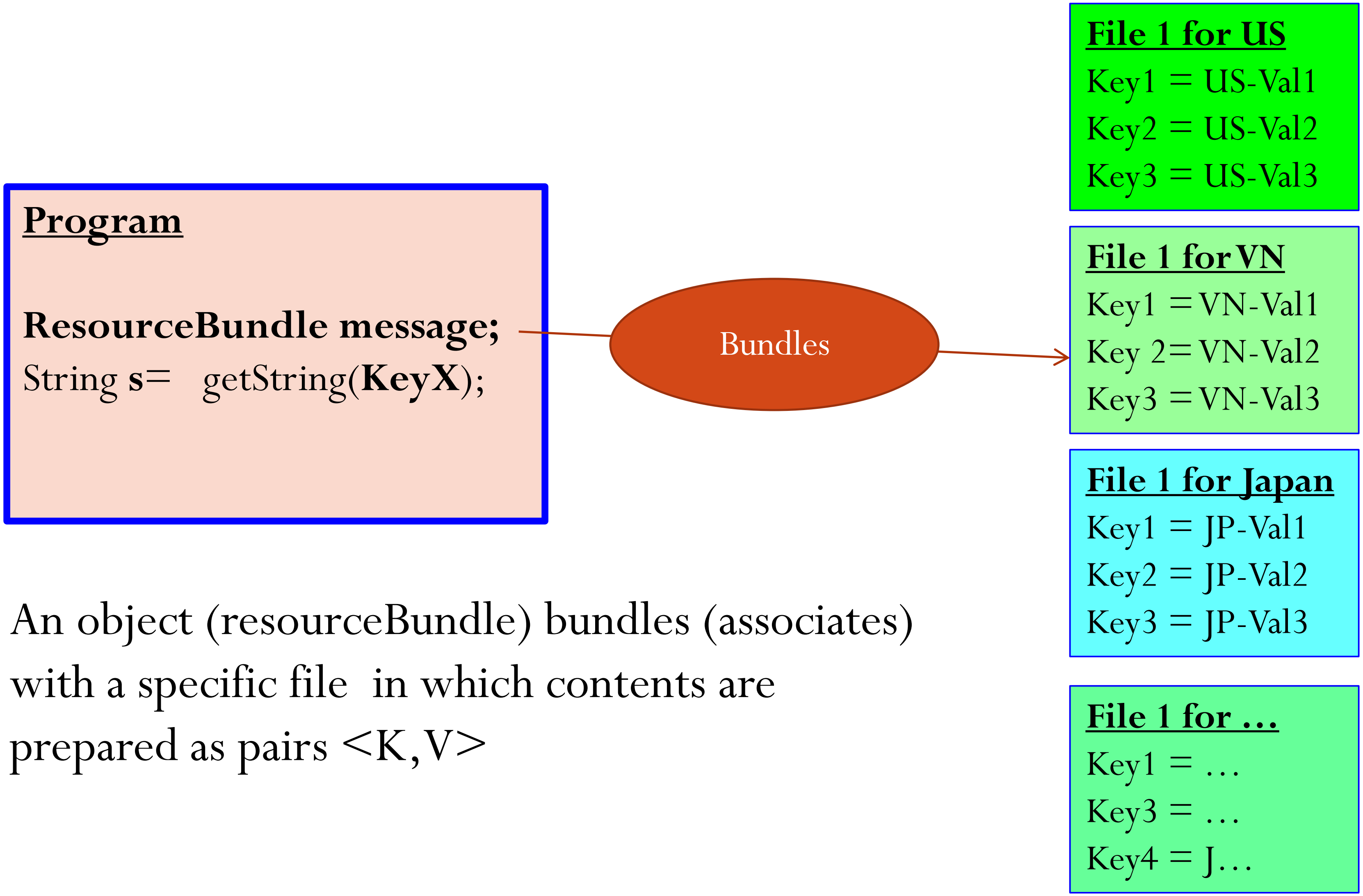
Input

Output

Contents	Objectives	Characteristic
Labels, guideline	Helping user easily using the program	It can be modified to make conformity with user
Inputs	Getting data	Language keyboard language is used by users
Messages	Communication between the program and users	It can be modified to make conformity with user

Unicode makes your wish come true

4- A way to Internationalize a Program



Program

ResourceBundle message;

String **s**= getString(**KeyX**);

File 1 for US
Key1 = US-Val1
Key2 = US-Val2
Key3 = US-Val3

File 1 for VN
Key1 =VN-Val1
Key 2=VN-Val2
Key3 =VN-Val3

File 1 for Japan
Key1 = JP-Val1
Key2 = JP-Val2
Key3 = JP-Val3

File 1 for …
Key1 = …
Key3 = …
Key4 = J…

An object (resourceBundle) bundles (associates) with a specific file in which contents are prepared as pairs <K,V>

5- Basic Java APIs for Internationalization

Class	Description
java.util.**Locale**	A Locale object represents a specific geographical, political, or cultural region. Data: language, country Some constants are defined for some locales
java.util.**ResourceBundle**	Resource bundles contain locale-specific objects. When your program needs a locale-specific resource, a String for example, your program can load it from the resource bundle that is appropriate for the current user's locale. A template file for this goal was defined including pairs <key, value> Filename of a resource bundle must be: **prefix_language_COUNTRY.properties** **(prefix: user-defined string)** ***Examples:*** **MessagesBundle_de_DE.properties** **MessagesBundle_en_US.properties** **MessagesBundle_vn_VN.properties** **(prefix = MessagesBundle)**

6- Steps for Internationalizing a Program

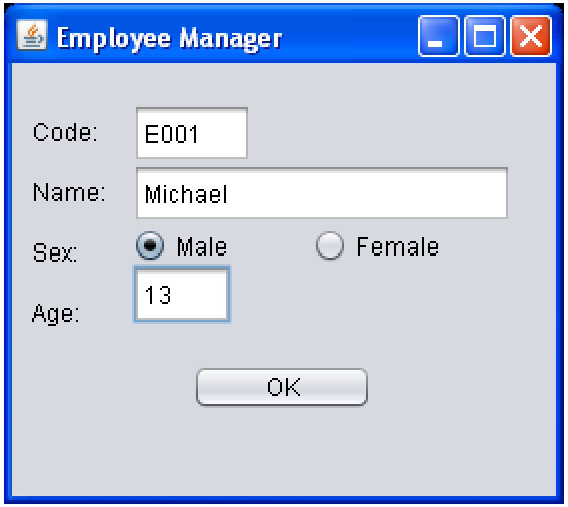
Step	Description
1	Determining means data will be presented to user

2	Creating resource file using the format:
	Key1 =Value1
	Key2 =Value2
	…

3	Develop the program. All data presented to users must be gotten from the appropriate resource file using classes **Local, ResourceBundle**
4	Making program configuration using the syntax: **java package. Class language country** *Examples* **java package.class vi VN** **java package.class en US**

Demo 1

- This demonstration depicts steps to develop an international program.



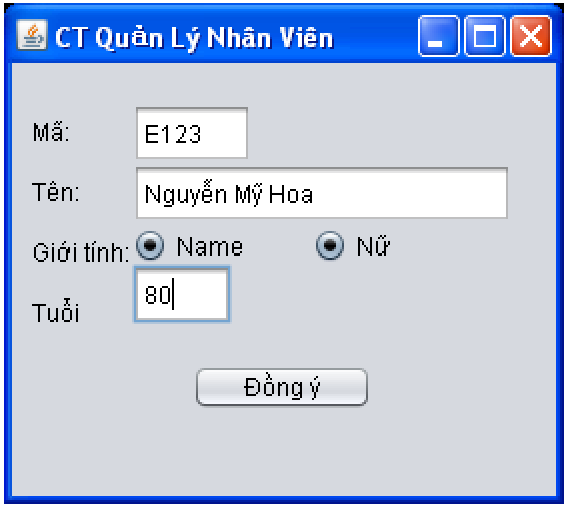
Employee Manager

Code: E001

Name: Michael

Sex: ● Male ○ Female

Age: 13



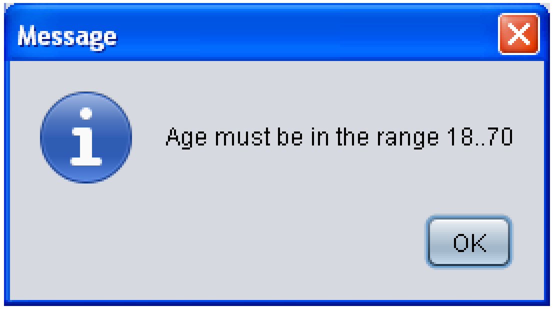
CT Quản Lý Nhân Viên

Mã: E123

Tên: Nguyễn Mỹ Hoa

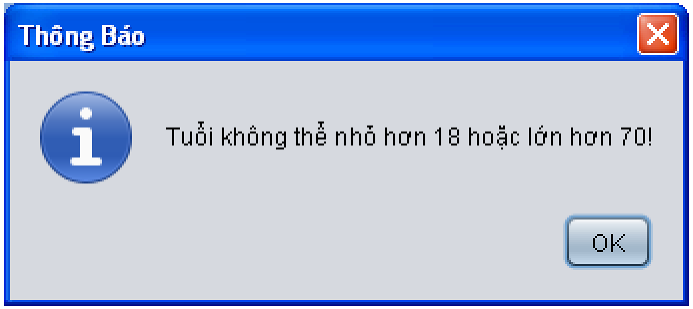
Giới tính: ● Name ● Nữ

Tuổi 80



Message

Age must be in the range 18..70



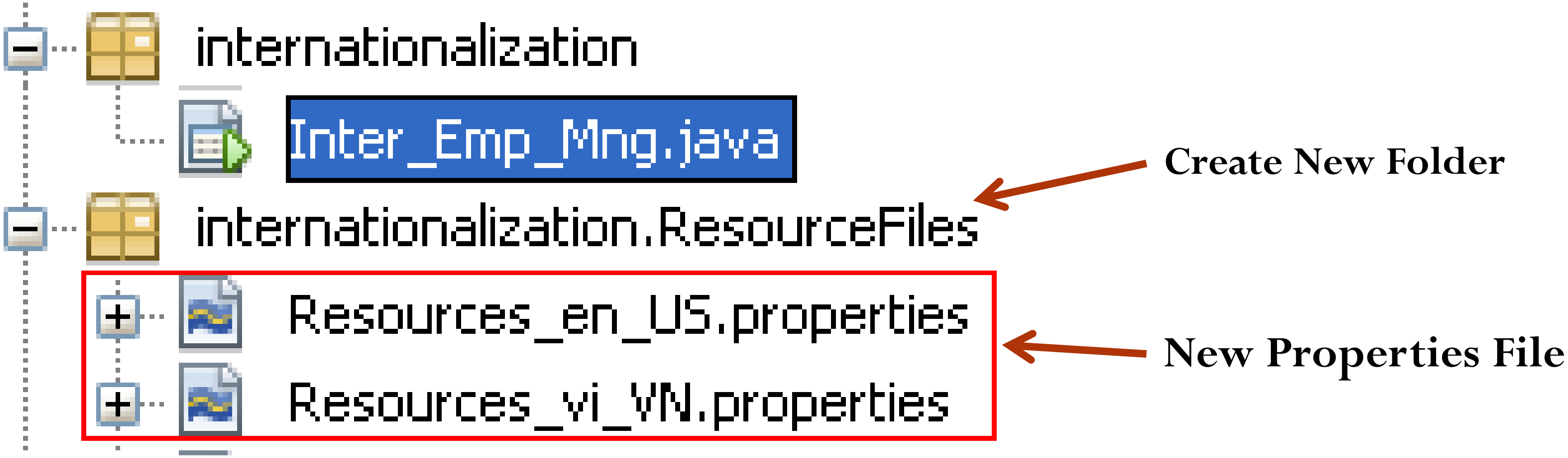
Thông Báo

Tuổi không thể nhỏ hơn 18 hoặc lớn hơn 70!

The program is used in US

The program is used in VN

Demo 1…



internationalization

Inter_Emp_Mng.java

internationalization.ResourceFiles

Resources_en_US.properties

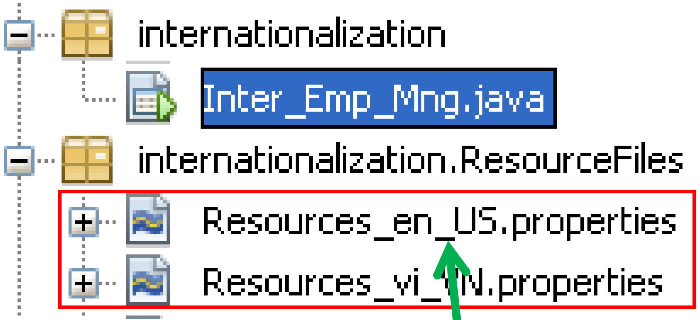
Resources_vi_VN.properties

Create New Folder

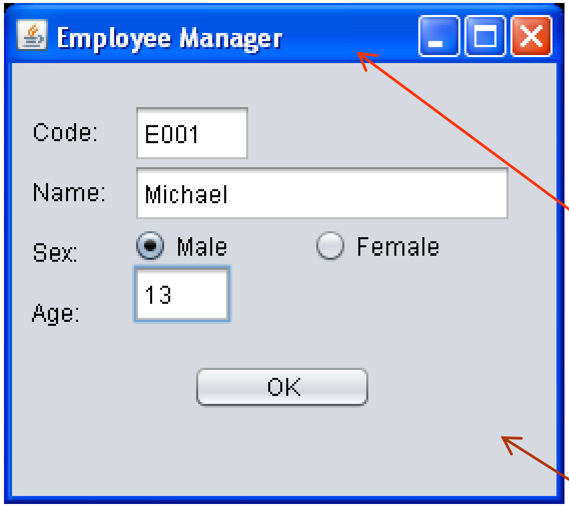
New Properties File

Demo 1…

Step 1: Determining means data will be presented to user
Step 2: Create resource files



- internationalization
  - Inter_Emp_Mng.java
- internationalization.ResourceFiles
  - Resources_en_US.properties
  - Resources_vi_VN.properties



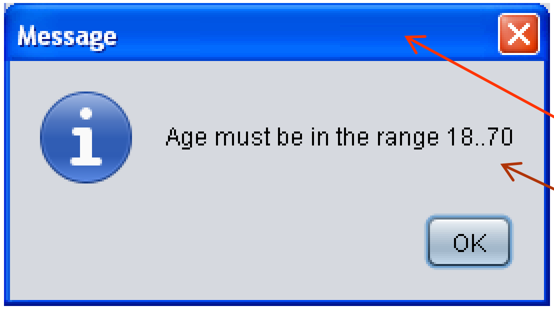
Employee Manager

Code: E001

Name: Michael

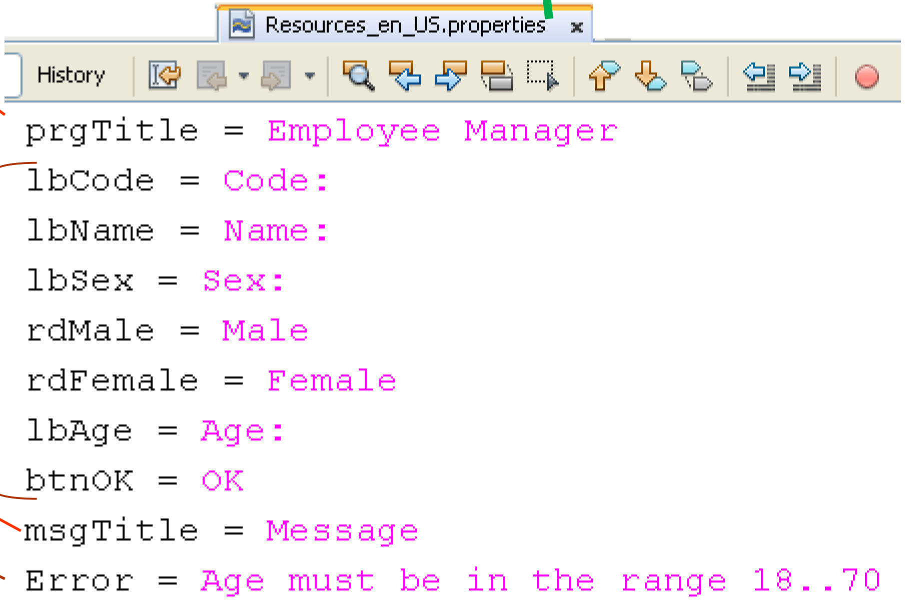
Sex: ● Male ○ Female

Age: 13



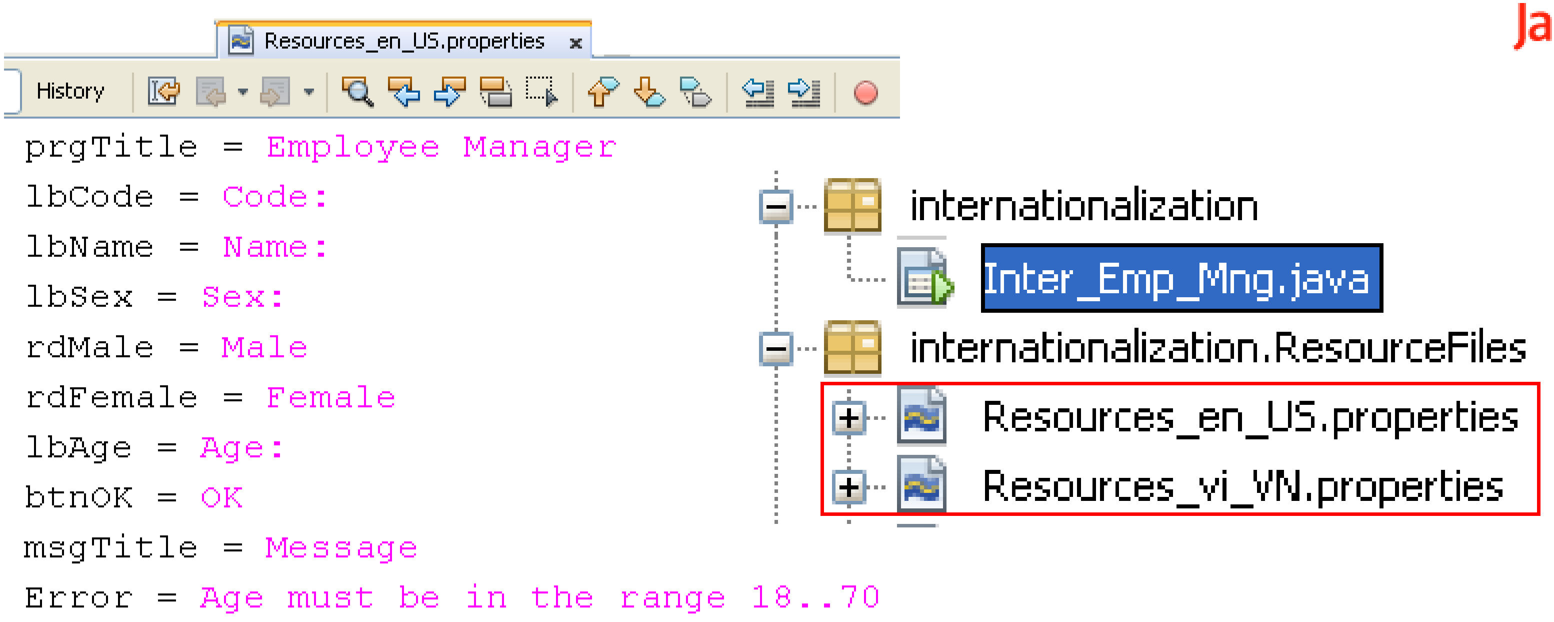
Message

Age must be in the range 18..70

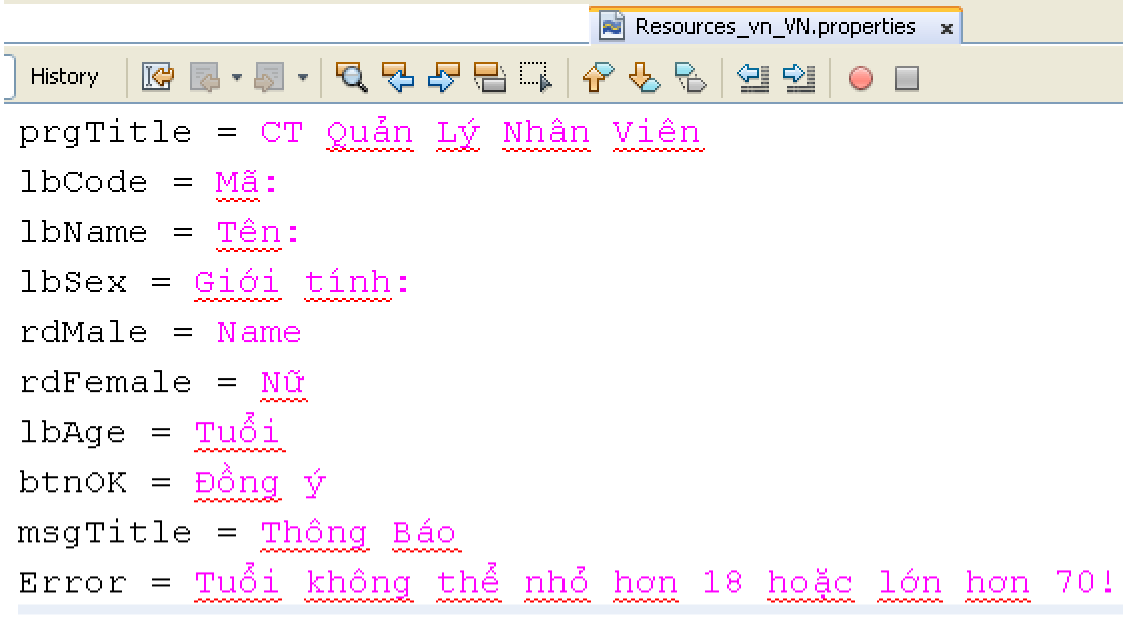


```
prgTitle = Employee Manager
lbCode = Code:
lbName = Name:
lbSex = Sex:
rdMale = Male
rdFemale = Female
lbAge = Age:
btnOK = OK
msgTitle = Message
Error = Age must be in the range 18..70
```

Demo 1…

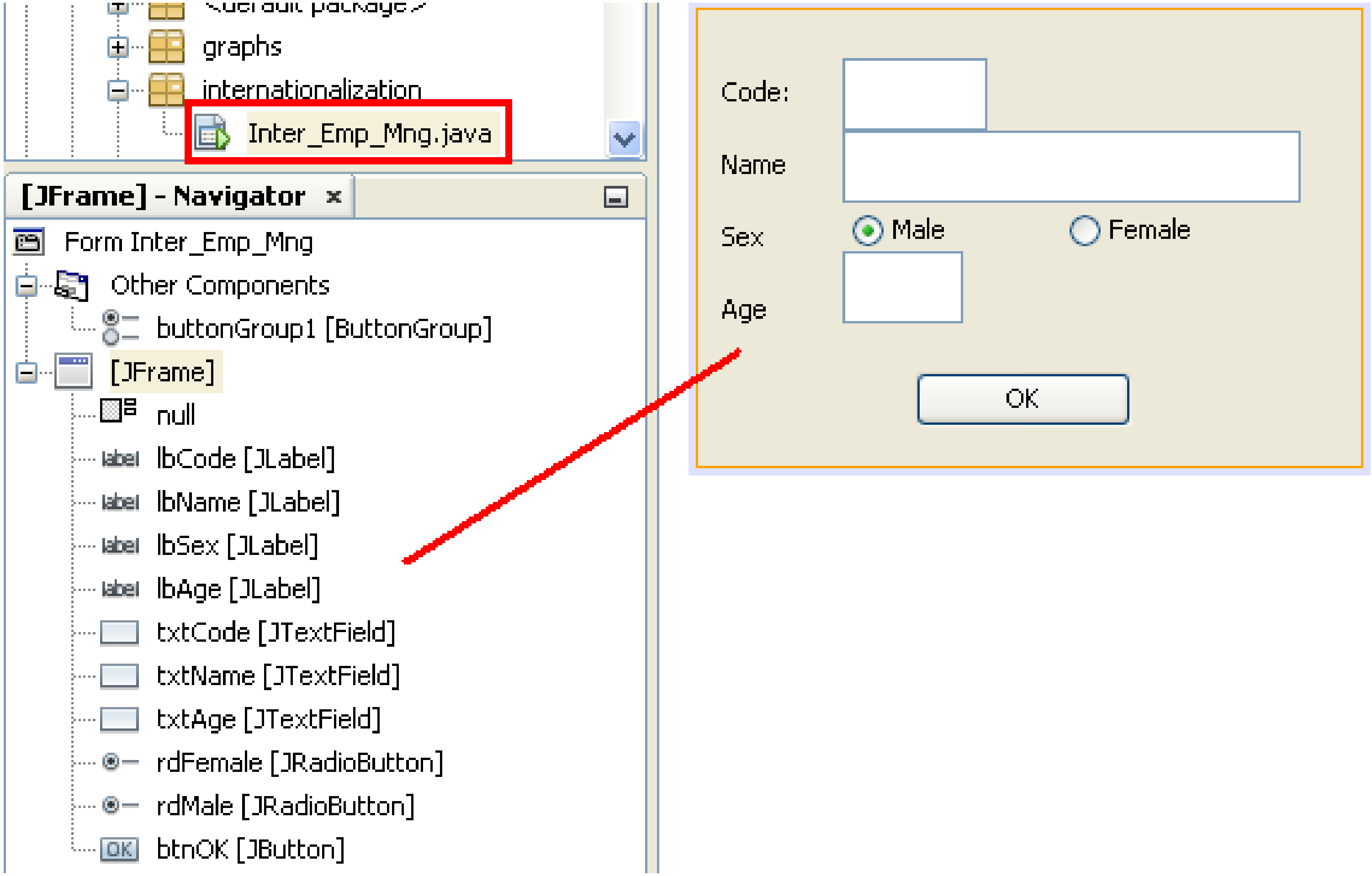

Resources_en_US.properties
History
internationalization
Inter_Emp_Mng.java
internationalization.ResourceFiles
Resources_en_US.properties
Resources_vi_VN.properties

```
prgTitle = Employee Manager
lbCode = Code:
lbName = Name:
lbSex = Sex:
rdMale = Male
rdFemale = Female
lbAge = Age:
btnOK = OK
msgTitle = Message
Error = Age must be in the range 18..70
```



```
prgTitle = CT Quản Lý Nhân Viên
lbCode = Mã:
lbName = Tên:
lbSex = Giới tính:
rdMale = Name
rdFemale = Nữ
lbAge = Tuổi
btnOK = Đồng ý
msgTitle = Thông Báo
Error = Tuổi không thể nhỏ hơn 18 hoặc lớn hơn 70!
```

Demo 1…



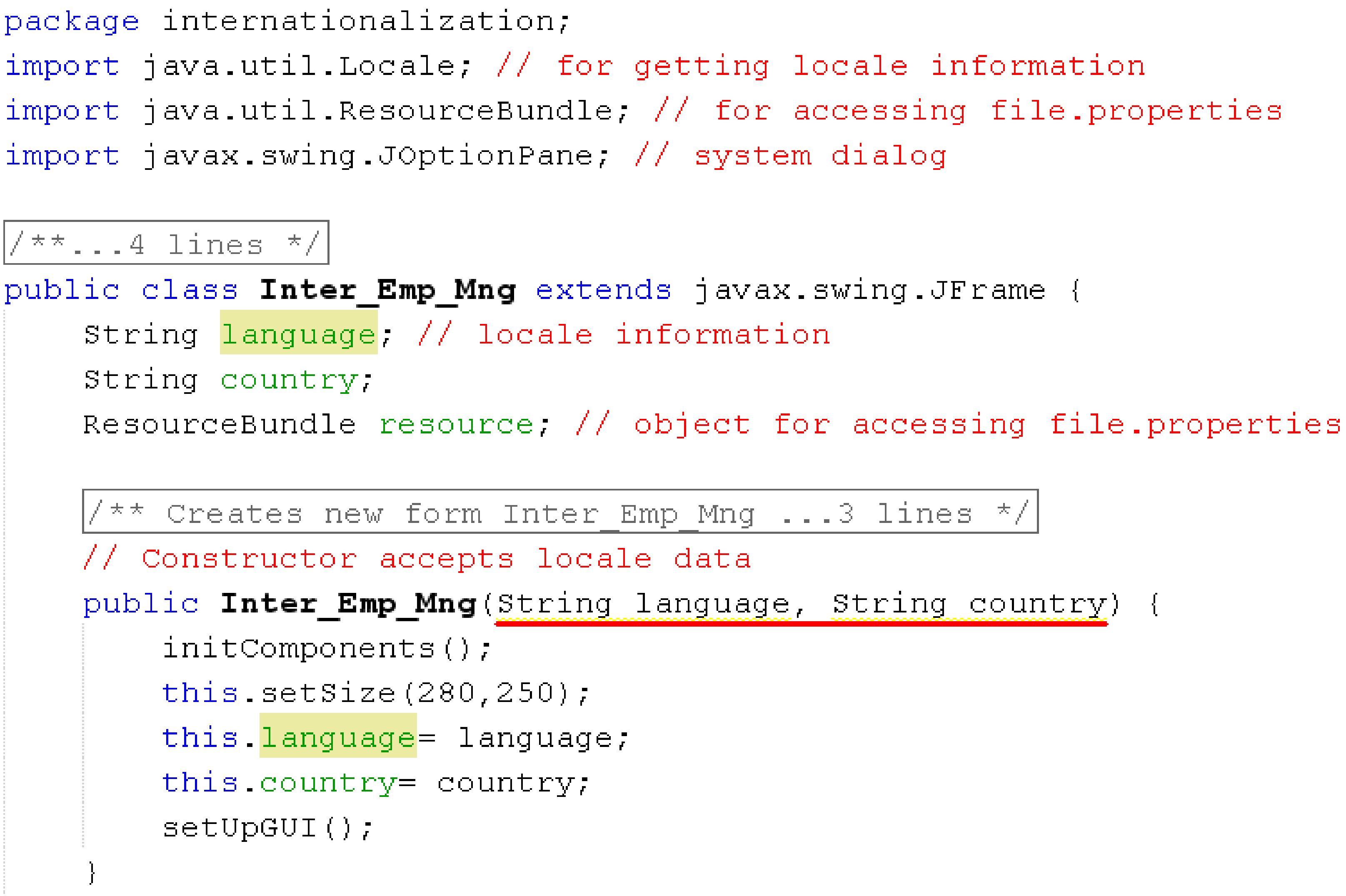
<default package>
graphs
internationalization
Inter_Emp_Mng.java

[JFrame] - Navigator ×

Form Inter_Emp_Mng
Other Components
buttonGroup1 [ButtonGroup]
[JFrame]
null
lbCode [JLabel]
lbName [JLabel]
lbSex [JLabel]
lbAge [JLabel]
txtCode [JTextField]
txtName [JTextField]
txtAge [JTextField]
rdFemale [JRadioButton]
rdMale [JRadioButton]
btnOK [JButton]

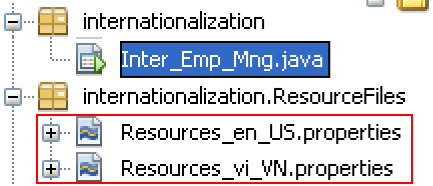
Code:
Name
Sex ● Male ○ Female
Age
OK

Demo 1…

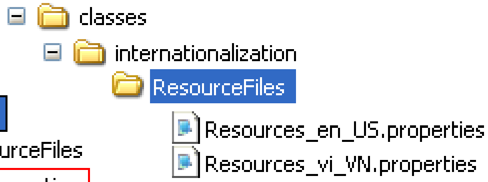


```
package internationalization;
import java.util.Locale; // for getting locale information
import java.util.ResourceBundle; // for accessing file.properties
import javax.swing.JOptionPane; // system dialog

/**...4 lines */
public class Inter_Emp_Mng extends javax.swing.JFrame {
    String language; // locale information
    String country;
    ResourceBundle resource; // object for accessing file.properties

    /** Creates new form Inter_Emp_Mng ...3 lines */
    // Constructor accepts locale data
    public Inter_Emp_Mng(String language, String country) {
        initComponents();
        this.setSize(280,250);
        this.language= language;
        this.country= country;
        setUpGUI();
    }
```

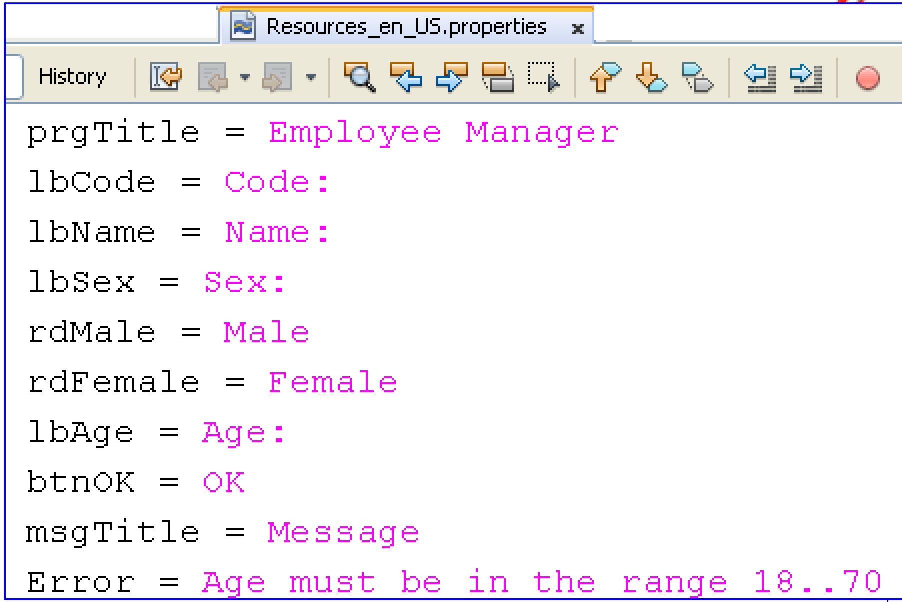
Demo 1…



- internalization
  - Inter_Emp_Mng.java
- internationalization.ResourceFiles
  - Resources_en_US.properties
  - Resources_vi_VN.properties

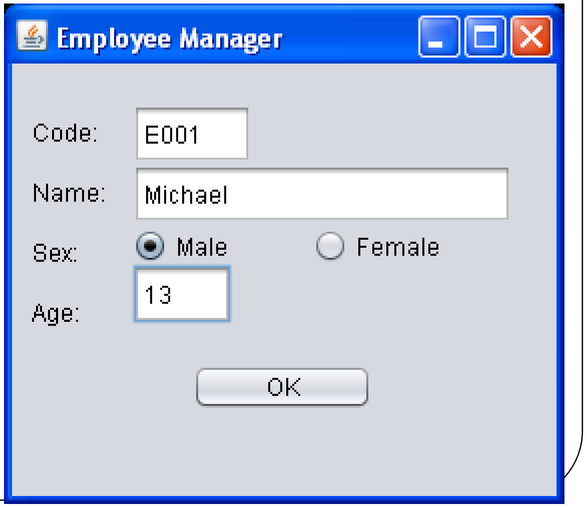


- classes
  - internationalization
    - ResourceFiles
      - Resources_en_US.properties
      - Resources_vi_VN.properties



```
prgTitle = Employee Manager
lbCode = Code:
lbName = Name:
lbSex = Sex:
rdMale = Male
rdFemale = Female
lbAge = Age:
btnOK = OK
msgTitle = Message
Error = Age must be in the range 18..70
```

```java
private void setUpGUI(){
    Locale currentLocale;
    currentLocale = new Locale(language, country);
    // Resource file name: Resources_language_country
    // Example: Resources_en_US
    // Example: Resources_vi_VN
    String base = "internationalization.ResourceFiles/Resources";
    resource = ResourceBundle.getBundle(base, currentLocale);
    this.setTitle(resource.getString("prgTitle"));
    lbCode.setText(resource.getString("lbCode"));
    lbName.setText(resource.getString("lbName"));
    lbSex.setText(resource.getString("lbSex"));
    rdMale.setText(resource.getString("rdMale"));
    rdFemale.setText(resource.getString("rdFemale"));
    lbAge.setText(resource.getString("lbAge"));
    btnOK.setText(resource.getString("btnOK"));
}
```



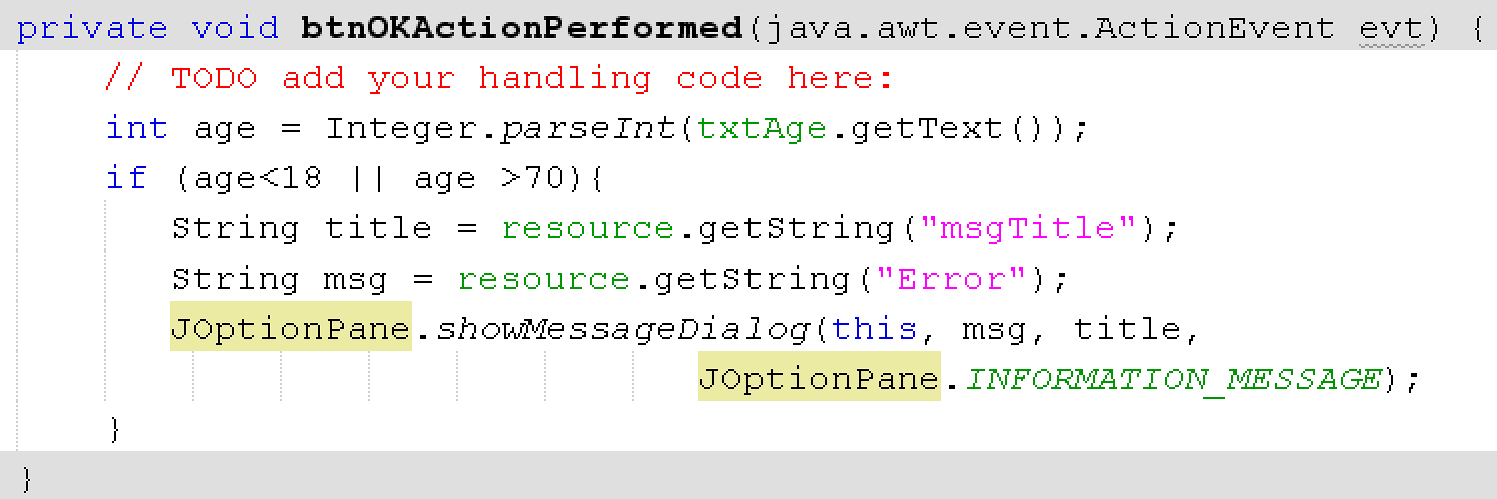
Code: E001

Name: Michael

Sex: ● Male ○ Female

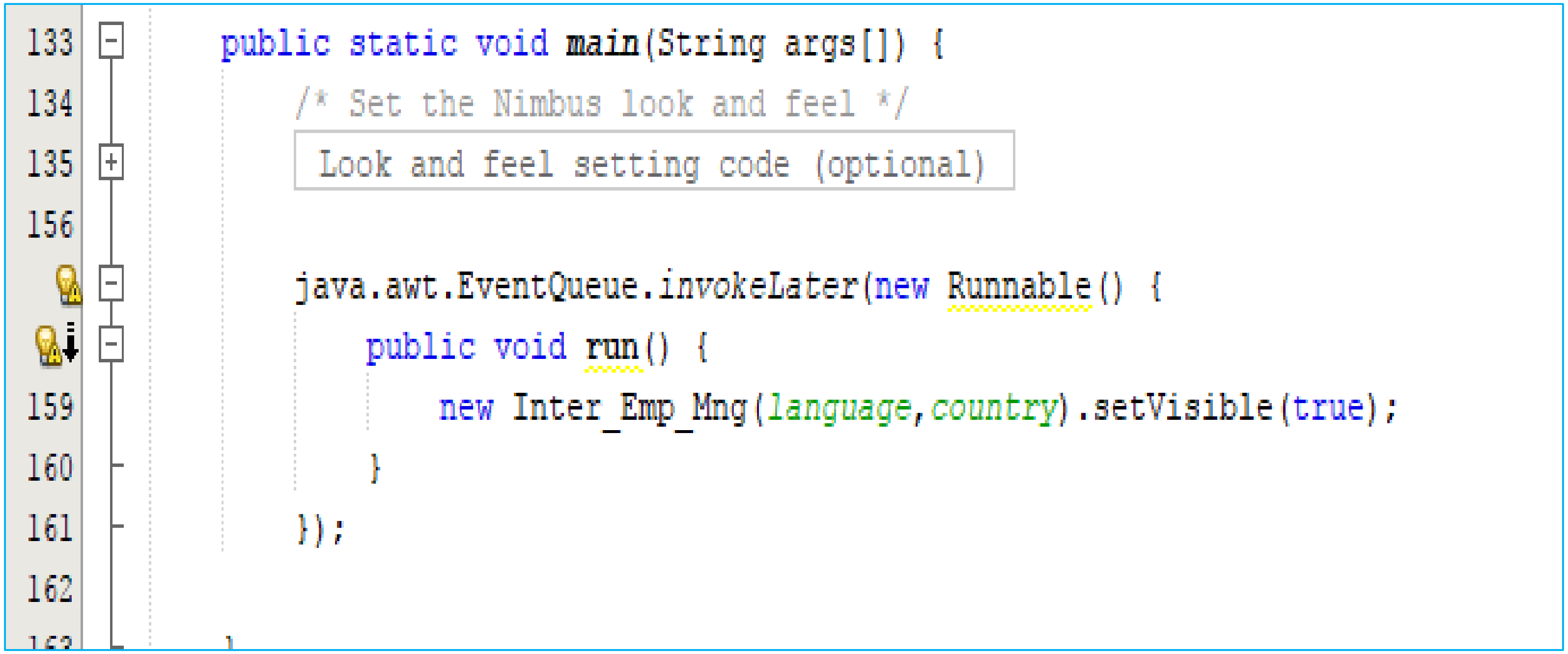
Age: 13

Demo 1…



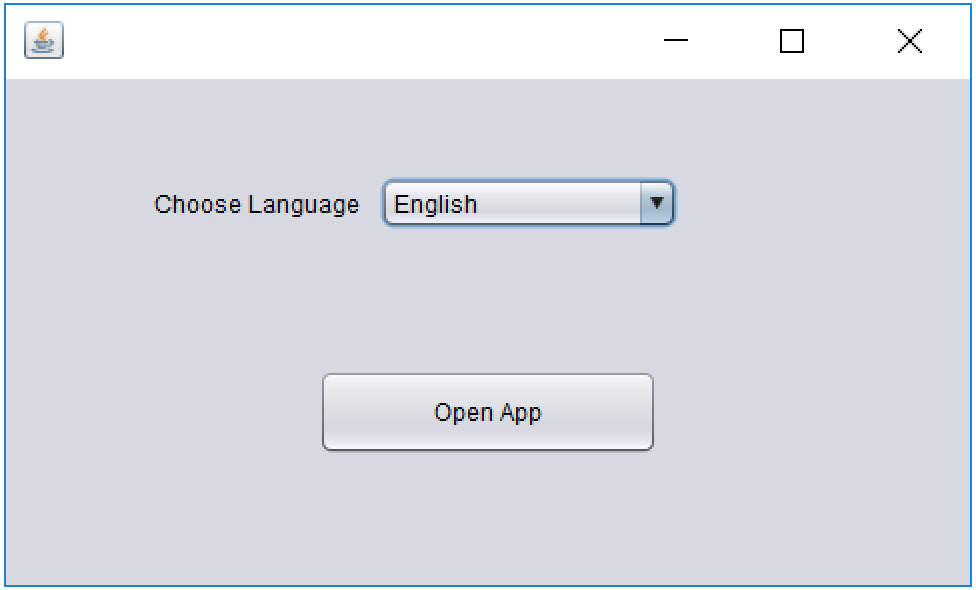
```
private void btnOKActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    int age = Integer.parseInt(txtAge.getText());
    if (age<18 || age >70){
        String title = resource.getString("msgTitle");
        String msg = resource.getString("Error");
        JOptionPane.showMessageDialog(this, msg, title,
                                JOptionPane.INFORMATION_MESSAGE);
    }
}
```

Demo 1…



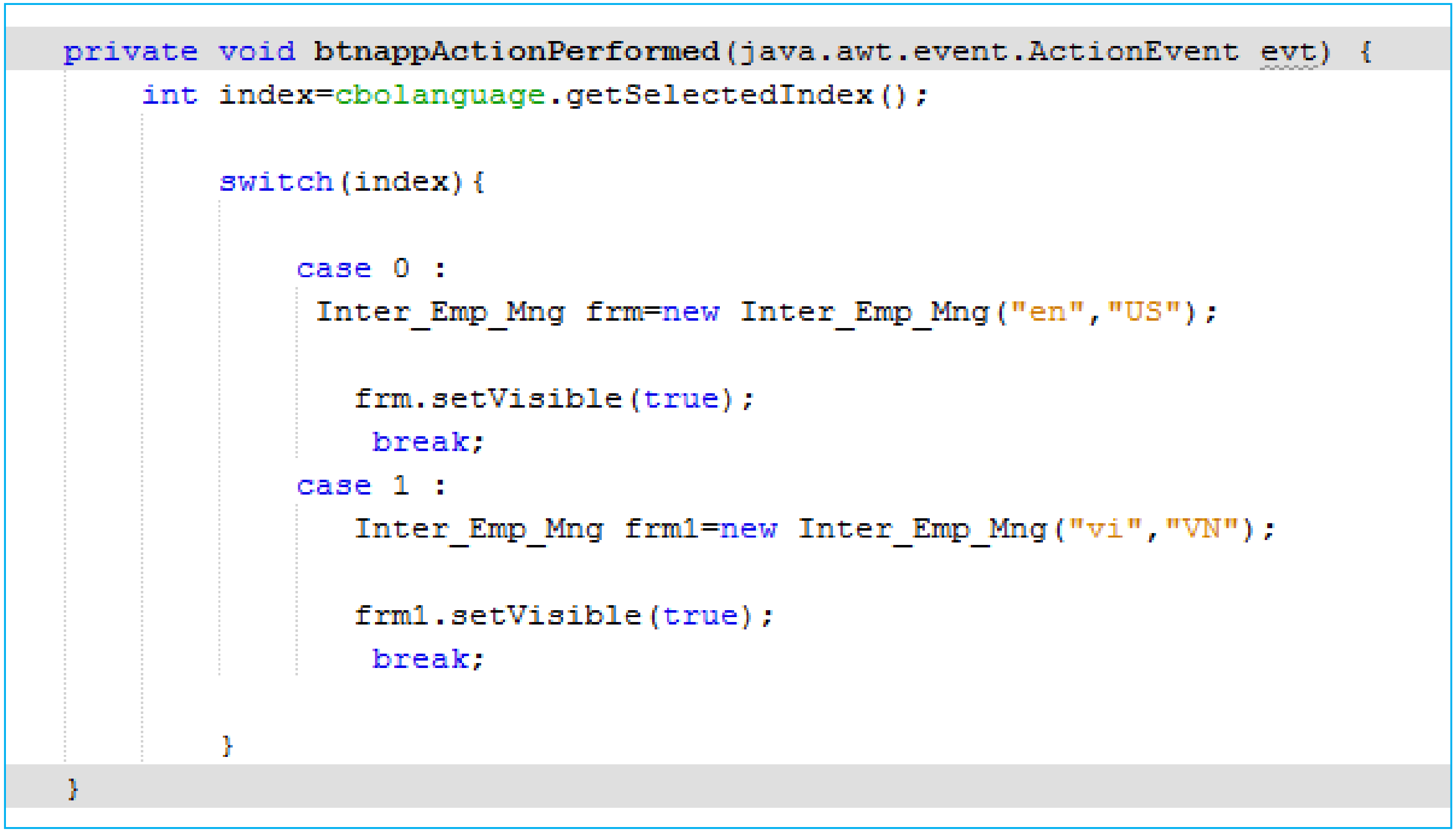
```
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        Look and feel setting code (optional)

        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Inter_Emp_Mng(language,country).setVisible(true);
            }
        });

    }
```

Demo 1…



Choose Language: English

Demo 1…



```
private void btnappActionPerformed(java.awt.event.ActionEvent evt) {
    int index=cbolanguage.getSelectedIndex();

        switch(index){

            case 0 :
             Inter_Emp_Mng frm=new Inter_Emp_Mng("en","US");

               frm.setVisible(true);
                break;
            case 1 :
               Inter_Emp_Mng frm1=new Inter_Emp_Mng("vi","VN");

               frm1.setVisible(true);
                break;

        }
}
```

Summary

1- Internationalization

2- Localization

3- Which contents of a program can be internationalized?

4- A way to Internationalize a Program

5- Basic Java APIs for Internationalization

6- Steps for Internationalizing a Program

7- A Demonstration

Thank You