Practices for Lesson 19: Localization

Chapter 19

Practices for Lesson 19: Overview

Practices Overview

In these practices, you create a date application that is similar to the example used in the lesson. For each practice, a NetBeans project is provided for you. Complete the project as indicated in the instructions.

Practice 19-1: Summary Level: Creating a Localized Date Application

Overview

In this practice, you create a text-based application that displays dates and times in a number of different ways. Create the resource bundles to localize the application for French, Simplified Chinese, and Russian.

Assumptions

You have attended the lecture for this lesson. You have access to the JDK8 API documentation.

Summary

Create a simple text-based date application that displays the following date information for today:

- Full date
- Long date
- Short date
- Medium date/time
- Medium time

Localize the application so that it displays this information in Simplified Chinese and Russian. The user should be able to switch between the languages.

The application output in English is shown here.

```
=== Date App ===
Full Date is: Tuesday, June 17, 2014
Long Date is: June 17, 2014
Short Date is: 6/17/14
Medium Date and Time is: Jun 17, 2014 10:51:09 AM
Medium Time is: 10:51:09 AM

--- Choose Language Option ---
1. Set to English
2. Set to French
3. Set to Chinese
4. Set to Russian
q. Enter q to quit
Enter a command:
```

Tasks

1. Open the `Localized19-01Prac` project in NetBeans.
   - Select File > Open Project.
   - Browse to `/home/oracle/labs/19-Localization/practices/practice1`.
   - Select `Localized19-01Prac` and click Open Project.
2. Edit the `DateApplication.java` file.
3. Create a message bundle for Russian and Simplified Chinese.
   - The translated text for the menus can be found in the `MessagesText.txt` file in the `practices` directory.

4. Add code to display the specified date formats (indicated with comments) and localized text.
5. Add code to change the `Locale` based on the user input.
6. Run the `DateApplication.java` file and verify that it operates as described.

Practice 19-1: Detailed Level: Creating a Localized Date Application

Overview

In this practice, you create a text-based application that displays dates and times in a number of different ways. Create the resource bundles to localize the application for French, Simplified Chinese, and Russian.

Assumptions

You have attended the lecture for this lesson. You have access to the JDK8 API documentation.

Summary

Create a simple text-based date application that displays the following date information for today:

- Full date
- Long date
- Short date
- Medium date/time
- Medium time

Localize the application so that it displays this information in Simplified Chinese and Russian. The user should be able to switch between languages.

The application output in English is shown here.

```
=== Date App ===
Full Date is: Tuesday, June 17, 2014
Long Date is: June 17, 2014
Short Date is: 6/17/14
Medium Date and Time is: Jun 17, 2014 10:51:09 AM
Medium Time is: 10:51:09 AM

--- Choose Language Option ---
1. Set to English
2. Set to French
3. Set to Chinese
4. Set to Russian
q. Enter q to quit
Enter a command:
```

Tasks

1. Open the `Localized19-01Prac` project in NetBeans.
   a. Select File > Open Project.
   b. Browse to `/home/oracle/labs/19-Localization/practices/practice1`.
   c. Select `Localized19-01Prac` and click Open Project.

2. Expand the project directories.
3. Edit the `DateApplication.java` file.

4. Open the `MessagesText.txt` file found in the `practices` directory for this practice in a text editor.
5. Create a message bundle file for Russian text named `MessagesBundle_ru_RU.properties`.
   - Right-click the project and select New > Other > Other > Properties File.
   - Click Next.
   - Enter `MessagesBundle_ru_RU` in the File Name field.
   - Click Browse.
   - Select the `src` directory.
   - Click Select Folder.
   - Click Finish.
   - Paste the localized Russian text into the file and save it.
6. Create a message bundle file for Simplified Chinese text named `MessagesBundle_zh_CN.properties`.
   - Right-click the project and select New > Other > Other > Properties File.
   - Click Next.
   - Enter `MessagesBundle_zh_CN` in the File Name field.
   - Click Finish.
   - Paste the localized Simplified Chinese text into the file and save it.
7. Update the code that sets the locale based on user input.

```
    public void setEnglish(){
        currentLocale = Locale.US;
        messages = ResourceBundle.getBundle("MessagesBundle",
currentLocale);
    }

    public void setFrench(){
        currentLocale = Locale.FRANCE;
        messages = ResourceBundle.getBundle("MessagesBundle",
currentLocale);
    }

    public void setChinese(){
        currentLocale = Locale.SIMPLIFIED_CHINESE;
        messages = ResourceBundle.getBundle("MessagesBundle",
currentLocale);
    }

    public void setRussian(){
        currentLocale = ruLocale;
        this.messages =
ResourceBundle.getBundle("MessagesBundle", currentLocale);
    }
```

8. Add the code that displays the date information to the `printMenu` method.

```
    public void printMenu(){
      pw.println("=== Date App ===");

      // Full Date
      df =
DateTimeFormatter.ofLocalizedDate(FormatStyle.FULL).withLocale(c
urrentLocale);
      pw.println(messages.getString("date1") + " " +
today.format(df));

      // Long Date
      df =
DateTimeFormatter.ofLocalizedDate(FormatStyle.LONG).withLocale(c
urrentLocale);
      pw.println(messages.getString("date2") + " " +
today.format(df));

      // Short Date
      df =
DateTimeFormatter.ofLocalizedDate(FormatStyle.SHORT).withLocale(
currentLocale);
      pw.println(messages.getString("date3") + " " +
today.format(df));

      // Medium Date/Time
      df =
DateTimeFormatter.ofLocalizedDateTime(FormatStyle.MEDIUM).withLo
cale(currentLocale);
      pw.println(messages.getString("date4") + " " +
today.format(df));

      // Medium Time
      df =
DateTimeFormatter.ofLocalizedTime(FormatStyle.MEDIUM).withLocale
(currentLocale);
      pw.println(messages.getString("date5") + " " +
today.format(df));

        pw.println("\n--- Choose Language Option ---");
        pw.println("1. " + messages.getString("menu1"));
        pw.println("2. " + messages.getString("menu2"));
        pw.println("3. " + messages.getString("menu3"));
        pw.println("4. " + messages.getString("menu4"));
        pw.println("q. " + messages.getString("menuq"));
        System.out.print(messages.getString("menucommand") + "
");
}
```

9. Run the `DateApplication.java` file and verify that it operates as described.