**МИНОБРНАУКИ РОССИИ**

**САНКТ–ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра вычислительной техники**

**ОТЧЕТ**

**по лабораторным работам №3**

**по дисциплине «Программирование интернет-приложений на языке Java»**

**Тема: ИНТЕРНАЦИОНАЛИЗАЦИЯ**

**Web-ПРИЛОЖЕНИЙ**

|  |  |  |
| --- | --- | --- |
| Студентка гр. 7308 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Фомченкова Н.А. |
| Преподаватель | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Павловский М.Г. |

Санкт–Петербург

2021

**Цель работы**

Знакомство со способами отображения данных на различных языках при использовании файлов ресурсов.

**Задание**

* Проанализируйте разработанное в лабораторной работе № 2 приложение и определите, для каких элементов пользовательского интерфейса можно

провести локализацию.

* Создайте файлы ресурсов для английского и русского языков.
* Внесите изменения в сервлет приложения для возможности переключения интерфейса на английский и русский языки.
* Запустите приложения в двух режимах и снимите скриншоты интерфейсов на английском и русском языках.

**Ход работы**

После анализа интерфейса разработанного приложения были составлены следующие файлы, содержащие значения элементов пользовательского интерфейса для английского и русского языка:

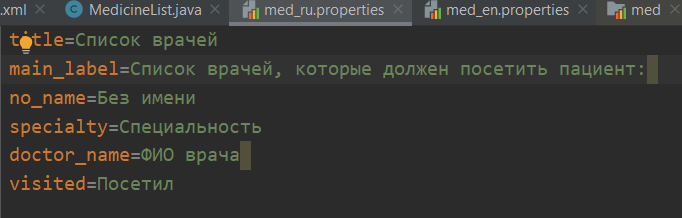


Рис.1 для английской локализации

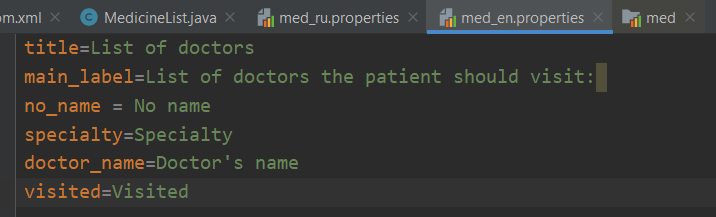


Рис.2 для русской локализации

Были внесены изменения в сервлет приложения для возможности переключения интерфейса на английский и русский языки.

//интернационализация  
String lang = request.getParameter("lang");  
if(lang == null){  
 response.sendError(HttpServletResponse.*SC\_NOT\_ACCEPTABLE*,  
 "Parameter lang doesn't found");  
 return;  
}  
//Установка локали  
ResourceBundle res = null;  
if("en".equalsIgnoreCase(lang) || "ru".equalsIgnoreCase(lang)) {  
 res = ResourceBundle.*getBundle*("med", "en".equalsIgnoreCase(lang) ? Locale.*ENGLISH* : Locale.*getDefault*());  
}else  
{  
 response.sendError(HttpServletResponse.*SC\_NOT\_ACCEPTABLE*, "Параметр lang может принимать значения en или ru");  
 return;  
}  
  
// Получение потока для вывода ответа  
PrintWriter out = response.getWriter();  
try {  
 // Создание HTML-страницы  
 out.println("<html>");  
 out.println("<head><title>" + res.getString("title") + "</title></head>");  
 out.println("<body>");  
 out.println("<h1>" + res.getString("main\_label") + (name != null ? name :  
 res.getString("no\_name")) + "</h1>");  
 out.println("<table border='1'>");  
 out.println("<tr><td><b>"+res.getString("specialty")+"</b></td><td><b>"+res.getString("doctor\_name") + "</b></td><td><b>"+res.getString("visited")+"</b></td></tr>");  
 out.println("<tr><td>Окулист </td><td>Иванов А.А. </td><td>Да</td></tr>");  
 out.println("<tr><td>Невролог </td><td>Журавлева Э.Ю. </td><td>Нет</td></tr>");  
 out.println("<tr><td>Хирург </td><td>Новиков С.Ю. </td><td>Да</td></tr>");  
 out.println("</table>");  
 out.println("</body>");  
 out.println("</html>");  
} finally {  
 // Закрытие потока вывода  
 out.close();  
}

Как видно из приведенного куска кода, если пользователь не задаст leng, программа работать не будет. Так же теперь у нас имеется два динамических параметра: имя(name), язык (lang).

Скриншоты работы программы:

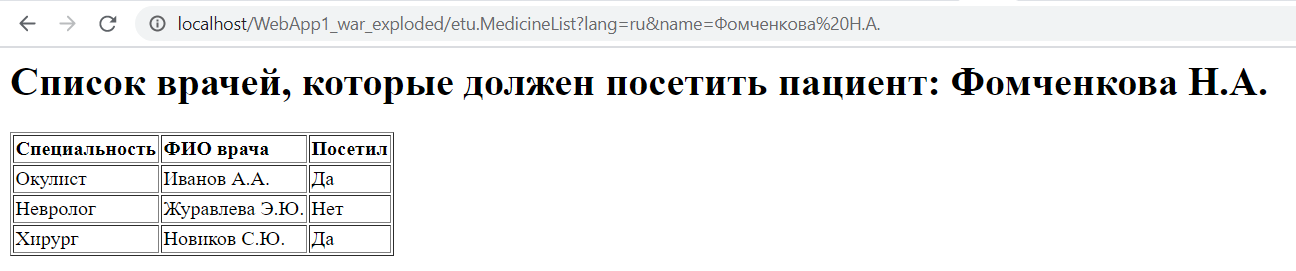


Рис.3 язык: русский, имя задано

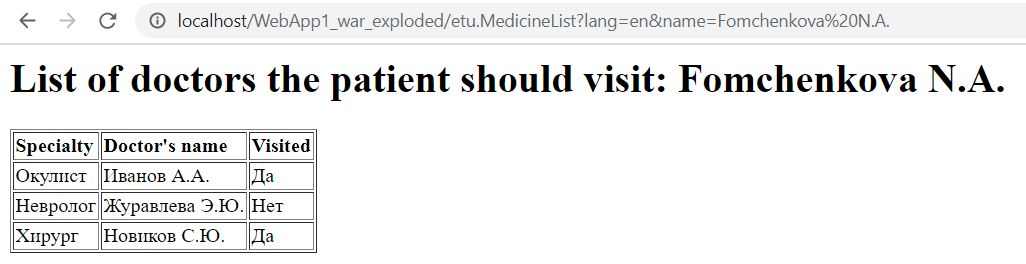


Рис.4 язык: английский, имя задано

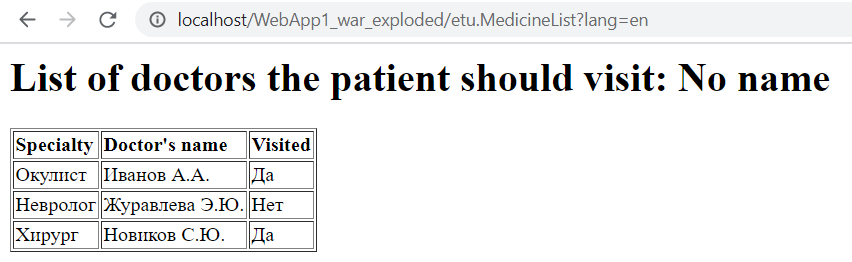


Рис.5 язык: английский, имя не задано

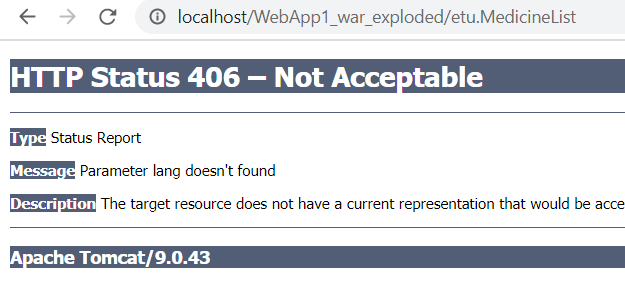


Рис.6 ошибка язык не задан

**Вывод**

В данной л/р мы познакомились с тем, что такое интернационализация (i18n). На практике убедились в ее необходимости, создав приложение, работающее на двух языках. Также познакомились с работой абстрактного класса ResourceBundle, который использовали для хранения данных, зависящих от локали.

**Листинг**

package etu;  
  
import java.io.IOException;  
import javax.servlet.ServletException;  
import javax.servlet.annotation.WebServlet;  
import javax.servlet.http.HttpServlet;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
// библиотечный класс для работы с потоками вывода  
import java.io.PrintWriter;  
import java.util.Locale;  
import java.util.ResourceBundle;  
  
  
*/\*\*  
 \** ***@author*** *Fomchenkova Nataliya 7308  
 \** ***@date*** *11.02.2021  
 \*/*@WebServlet("/etu.MedicineList")  
public class MedicineList extends HttpServlet {  
 private static final long *serialVersionUID* = 1L;  
 private final static ResourceBundle *resourceBundle* = ResourceBundle.*getBundle*("med");  
public MedicineList() {  
 super();  
 // *TODO Auto-generated constructor stub* }  
  
protected void processRequest(HttpServletRequest request, HttpServletResponse  
 response)  
 throws ServletException, IOException {  
// Задание типа кодировки для параметров запроса  
 request.setCharacterEncoding("utf-8");  
 // Чтение параметра name из запроса  
 String name = request.getParameter("name");  
 // Задание типа содержимого для ответа (в том числе кодировки)  
 response.setContentType("text/html;charset=UTF-8");  
  
 //интернационализация  
 String lang = request.getParameter("lang");  
 if(lang == null){  
 response.sendError(HttpServletResponse.*SC\_NOT\_ACCEPTABLE*,  
 "Parameter lang doesn't found");  
 return;  
 }  
 //Установка локали  
 ResourceBundle res = null;  
 if("en".equalsIgnoreCase(lang) || "ru".equalsIgnoreCase(lang)) {  
 res = ResourceBundle.*getBundle*("med", "en".equalsIgnoreCase(lang) ? Locale.*ENGLISH* : Locale.*getDefault*());  
 }else  
 {  
 response.sendError(HttpServletResponse.*SC\_NOT\_ACCEPTABLE*, "Параметр lang может принимать значения en или ru");  
 return;  
 }  
  
 // Получение потока для вывода ответа  
 PrintWriter out = response.getWriter();  
 try {  
 // Создание HTML-страницы  
 out.println("<html>");  
 out.println("<head><title>" + res.getString("title") + "</title></head>");  
 out.println("<body>");  
 out.println("<h1>" + res.getString("main\_label") + (name != null ? name :  
 res.getString("no\_name")) + "</h1>");  
 out.println("<table border='1'>");  
 out.println("<tr><td><b>"+res.getString("specialty")+"</b></td><td><b>"+res.getString("doctor\_name") + "</b></td><td><b>"+res.getString("visited")+"</b></td></tr>");  
 out.println("<tr><td>Окулист </td><td>Иванов А.А. </td><td>Да</td></tr>");  
 out.println("<tr><td>Невролог </td><td>Журавлева Э.Ю. </td><td>Нет</td></tr>");  
 out.println("<tr><td>Хирург </td><td>Новиков С.Ю. </td><td>Да</td></tr>");  
 out.println("</table>");  
 out.println("</body>");  
 out.println("</html>");  
 } finally {  
 // Закрытие потока вывода  
 out.close();  
 }  
 }  
@Override  
 protected void doGet(HttpServletRequest request, HttpServletResponse response)  
 throws ServletException, IOException {  
 processRequest(request, response);  
 }@Override  
 protected void doPost(HttpServletRequest request, HttpServletResponse response)  
 throws ServletException, IOException {  
 processRequest(request, response);  
 }  
  
 @Override  
 public void destroy() {  
 super.destroy();  
 }  
}

JavaDoc

