МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Лабораторна робота №2

з курсу

«Інженерія даних та знань»

*на тему:*

**«Практична реалізація алгоритмів обробки символьних рядків»**

Виконав:

ст. гр. ITIT-12

Кобак Р.З.

Прийняла:

Рибчак З.Л.

**Львів 2017**

**Мета роботи:**

Набуття практичних навичок застосування операцій над рядками.

**Індивідуальне завдання:**

**Варіант 11**

Реалізувати додаток, який дозволяє зчитувати вихідні рядки з файлів формату CSV (Comma Separated Value) або XML (Extensible Markup Language) та зберігати результати в файл та розробити процедури та функції які забезпечують виконання операції вказаних в завданні. В контрольному прикладі передбачити всі можливі комбінації вхідних параметрів (нульова довжина, вихід за межі рядка, розмір текстового файлу більше 100 МБ і т.п.), в тому числі і неправильні.

|  |  |  |
| --- | --- | --- |
| **№** |  | **Завдання** |
| 11 | WordCmp(s1,s2) | Порівняння рядків (з ігноруванням множинних пробілів). |

**Порядок виконання:**

Було розроблено додаток для зчитування вхідних даних форматів CSV i XML та порівняння рядків з ігноруванням множинних пробілів.

Код програми:

Клас StringItem.java

**public class** StringItem {  
 **int number**;  
 String **content**;  
}

Клас Main.java

**public class** Main {  
  
 **private static** String *XML\_FILE* = **"G:\\навчання\\5 курс\\Інженерія даних та знань\\lab2\\lab2\\xmlFile.xml"**;  
 **private static** String *CSV\_FILE* = **"G:\\навчання\\5 курс\\Інженерія даних та знань\\lab2\\lab2\\csvFile.csv"**;  
  
 **public static void** main(String[] args) **throws** Exception {  
 List<String> xmlStrings = *readXMLFile*();  
 List<StringItem> csvStrings = *readCSV*(*CSV\_FILE*);  
 BufferedReader br = **new** BufferedReader(**new** InputStreamReader(System.***in***));  
 System.***out***.println(String.*format*(**"Choose xml string number (enabled %d strings)"**, xmlStrings.size()));  
 **int** xmlIndex = Integer.*parseInt*(br.readLine());  
 System.***out***.println(String.*format*(**"Choose csv string number (enabled %d strings)"**, csvStrings.size()));  
 **int** csvIndex = Integer.*parseInt*(br.readLine());  
  
 **int** result = *WordCmp*(xmlStrings.get(xmlIndex), csvStrings.get(csvIndex).**content**);  
 System.***out***.println(**"Strings:"**);  
 System.***out***.println(**"S1: "** + xmlStrings.get(xmlIndex));  
 System.***out***.println(**"S2: "** + csvStrings.get(csvIndex).**content**);  
 **if** (result == 0) {  
 System.***out***.println(**"String have equal length"**);  
 } **else if** (result == 1) {  
 System.***out***.println(**"S1 bigger"**);  
 } **else** {  
 System.***out***.println(**"S2 bigger"**);  
 }  
  
 System.***out***.println();  
 }  
  
 **private static int** WordCmp(String s1, String s2) {  
 s1 = s1.replaceAll(**"\\s{2,}"**, **" "**).trim();  
 s2 = s2.replaceAll(**"\\s{2,}"**, **" "**).trim();  
 **if** (s1.length() == s2.length()) {  
 **return** 0;  
 }  
 **return** s1.length() > s2.length() ? 1 : -1;  
 }  
  
 **private static** List<String> readXMLFile() **throws** Exception {  
 List<String> elementList = **new** ArrayList<>();  
  
 File fXmlFile = **new** File(*XML\_FILE*);  
 DocumentBuilderFactory dbFactory = DocumentBuilderFactory.*newInstance*();  
 DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();  
 Document doc = dBuilder.parse(fXmlFile);  
 doc.getDocumentElement().normalize();  
  
 System.***out***.println(**"Root element :"** + doc.getDocumentElement().getNodeName());  
 NodeList nList = doc.getElementsByTagName(**"stringItem"**);  
 **for** (**int** i = 0; i < nList.getLength(); i++) {  
 elementList.add(nList.item(i).getTextContent());  
 }  
  
 **return** elementList;  
 }  
  
 **private static** List<StringItem> readCSV(String inputFilePath) {  
 List<StringItem> inputList = **new** ArrayList<>();  
 **try**{  
 File inputF = **new** File(inputFilePath);  
 InputStream inputFS = **new** FileInputStream(inputF);  
 BufferedReader br = **new** BufferedReader(**new** InputStreamReader(inputFS));  
 inputList = br.lines().map(*mapToItem*).collect(Collectors.*toList*());  
 br.close();  
 } **catch** (Exception e) {  
 }  
 **return** inputList ;  
 }  
  
 **private static** Function<String, StringItem> *mapToItem* = (line) -> {  
 String[] p = line.split(**","**);  
 StringItem item = **new** StringItem();  
 item.**number** = Integer.*parseInt*(p[0]);  
 **if** (p[1] != **null** && p[1].trim().length() > 0) {  
 item.**content** = p[1];  
 }  
 **return** item;  
 };  
}

**Результат виконання програми**

Результати виконання програми представлений на Рис.1. Як видно з рисунку незважаючи на те що в рядку 1 багато пробілів і візуально він довший додаток всерівно повернува результат, що в рядку 2 більше симоволів.

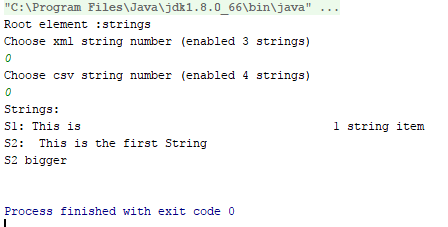


Рис.1 Результат виконання програми

**Висновок:** В результаті виконання даної лабораторної роботи було розроблено додаток для зчитування вхідних рядків з файлів формату CSV та XML та проведення порівняння даних рядків з ігноруванням множинних пробілів.