Лабораторна робота 3

ОБ'ЄДНАННЯ ТА ІНДЕКСУВАННЯ НОВИН З RSS-КАНАЛІВ

Мета: навчитися застосовувати технології обробки xml-документів на мові Java для аналізу вмісту RSS-каналів; опанувати техніки роботи з бібліотекою Mockito для побудови тестового середовища для модульного тестування

Оцінювання роботи: 7 (завдання 1) + 6 (завдання 2) + 7 (завдання 3) = 20
Термін здачі роботи без штрафних балів: 02.05.2017 – 02.06.2017

Завдання до лабораторної роботи. Постановка задачі

Новини зчитуються з RSS-каналів. Список каналів формується самостійно, виходячи із предметної області в індивідуальному завданні (табл.1). До списку новин можна дати запит, який містить ключове слово: термін, прізвище, назву країни тощо. За цим запитом новинні повідомлення індексуються: здійснюється пошук кількості повторення ключового слова в описі кожного повідомлення. Результат запиту видає топ-10 новин, упорядкованих за зростанням кількості повторень.

Завдання 1. Отримання інформації у форматі RSS з сайтів інформаційних агентств

- записати у текстовий файл список RSS-каналів за заданою предметною областю (табл. 1). Приблизний список постачальників інформації у форматі RSS (список можна доповнити):
  - CNN RSS: http://edition.cnn.com/services/rss/
  - theGuardiansRSS: https://www.theguardian.com/world/rss
  - Telegraph RSS Feeds: http://www.telegraph.co.uk/finance/rssfeeds/
  - BBC world Feed: https://www.repeatsoftware.com/Help/RSSFeedList.htm
  - NY times: http://www.nytimes.com/services/xml/rss/index.html
  - Metro RSS Feeds: http://www.metronews.ca/about/rssfeeds.html
  - The Star RSS: http://www.thestar.com.my/rss/
  - Independent RSS feeds: http://www.independent.co.uk/service/rss-feeds-775086.html
  - Reuters News RSS Feeds: http://uk.reuters.com/tools/rss
  - European Union Newsroom: http://www.liga.net/rss/
  - Unian: https://rss.unian.net/site/news_eng.rss

- побудувати синтаксичний аналізатор, який аналізує структуру документа у форматі RSS, отриманого у постачальника інформації, і створює базу новинних повідомлень
- підтримувати збереження новинних повідомлень між запусками програми, оновлення та видалення тих, які зберігаються більше встановленого терміну.

Корисні посилання:

1. Class java.net.URL: https://docs.oracle.com/javase/7/docs/api/java/net/URL.html
2. The Java™ Tutorials. Lesson: Working with URLs: https://docs.oracle.com/javase/tutorial/networking/urls/
3. Вікіпедія: сторінка про RSS: https://uk.wikipedia.org/wiki/RSS
4. XML RSS: https://www.w3schools.com/xml/xml_rss.asp

5. RSS specifications: http://www.rss-specifications.com/rss-specifications.htm
6. Пакет javax.xml.parsers: https://docs.oracle.com/javase/7/docs/api/javax/xml/parsers/package-summary.html
7. The Java™ Tutorials. Trail: Java API for XML Processing (JAXP): http://docs.oracle.com/javase/tutorial/jaxp/index.html
8. The Java™ Tutorials. Object Streams: http://docs.oracle.com/javase/tutorial/essential/io/objectstreams.html
9. Interface Serializable: https://docs.oracle.com/javase/8/docs/api/java/io/Serializable.html

Рекомендації до реалізації:

- список постачальників інформації, наведений у завданні 1, не є вичерпним. Його треба доповнити іншими постачальниками залежно від обраної предметної області;
- зчитати інформацію у постачальника інформації як потік, отриманий через URL-з'єднання [1, 2];
- кожне новинне повідомлення зберігається як об'єкт класу `FeedMessage`. Інформація у полях об'єкту збігається з вмістом тега `item` документа у форматі RSS [3-5];
- синтаксичний аналізатор у завданні треба побудувати на основі XML-аналізаторів у пакеті javax.xml.parsers [6, 7];
- для збереження бази новинних повідомлень та їх читання слід використати сериалізацію [8]. Для цього клас `FeedMessage` та класи, об'єкти яких серіалізуються, повинні реалізовувати інтерфейс-маркет `Serializable` [9];
- кожне повідомлення характеризується датою публікації. Очищення бази новинних повідомлень здійснюється за параметром: кількість днів зберігання. Цей параметр встановлюється власноруч. Якщо дата публікації повідомлення перевищила термін зберігання, який вираховується, виходячи с поточної дати, повідомлення вилучається з бази;
- базу новинних повідомлень зберігається у структурі даних, яка більш доцільна для розв'язання поточних задач. Для цього не слід використовувати базу даних

Завдання 2. Індексування базу новин за ключовим словом

- знайти індекси (кількість повторень ключового слова) у новинах
- організувати клієнтську частину, де у діалоговому режимі клієнт вводить ключове слово (або фраза), за яким буде індексуватися база новин. Якщо слово належить списку "stop-words" (файл stop-words.txt), наприклад слова «nobody», «the», «is», то користувачу надається повідомлення про неможливість виконання індексування.
- вивести результати індексування: список перших 10 новин з найбільшими індексами

Рекомендації до реалізації:

- список stop-words слід зчитати з файлу і зберігати у хеш-таблиці
- не слід індексувати одну статтю двічі. Дві статті є однаковими, якщо вони мають один той самий URL (навіть якщо різні заголовки) або той самий заголовок і прийшли з одного сервера

Завдання 3. Модульне тестування засобами тестового середовища JUnit4 та Mockito

- описати тестові класи і методи для перевірки методів класів для додавання, видалення та оновлення існуючої бази новинних повідомлень. Для отримання і зберігання серіалізованих даних використати тестовий дублер (mock-об'єкт). Виконати тести;
- описати тестові класи і методи для перевірки методів класів для взаємодії синтаксичного аналізатора і бази повідомлень. Для отримання від синтаксичного аналізатора використати тестовий дублер (mock-об'єкт). Виконати тести;
- описати тестові класи і методи для перевірки методів класів для роботи синтаксичного аналізатора документа у форматі RSS. Для отримання даних з RSS-каналу використати тестовий дублер (mock-об'єкт). Виконати тести;
- описати тестове середовище і виконати в ньому тести для перевірки взаємодії об'єктів під час індексування і виведення результатів індексування новинних повідомлень.

Корисні посилання:

1. Mockito: http://site.mockito.org/
2. Mockito Tutorial: https://www.tutorialspoint.com/mockito/

Рекомендації до реалізації:

- тестування проекту за допомогою тестових дублерів (mock-об'єктів) слід почати до організації URL-з'єднання і десеріалізації, тобто отримання реальних даних;
- порядок опису та виконання тестів у завдання може бути змінений, виходячи з власного порядку виконання роботи;
- бажано, щоб тести були написані і застосовувались не для перевірки і підтвердження роботи вже функціонуючого проекту, а для пошуку помилок, перевірки граничних умов, отримання хибних даних, неможливості доступу до зовнішніх джерел даних.

Індівідуальні завдання

Таблиця 1

Варіант	Предметна область	Варіант	Предметна область
1	**2**	**3**	**4**
1	News. Business	12	World Sport
2	Technologies	13	Education
3	Life&Style	14	Environment&Climate
4	Art&Entertainment	15	Health
5	News.World	16	Movies
6	Economy	17	Technology & Media
7	Transport & Travel	18	News.Europe
8	Money&	19	News.UK
9	Science & Space	20	News.Ukraine
10	News.USA	21	Markets&Retail
11	Work&Careers	22	Auto