Лабораторна робота 1

РЕГУЛЯРНІ ВИРАЗИ. АЛГОРИТМИ ІДЕНТИФІКАЦІЇ ТЕКСТОВИХ ОБРАЗІВ

Мета: дослідити техніки роботи з регулярними виразами, набути практичних навичок використання засобів пакету `java.util.regex` мови Java для пошуку текстових образів, навитися реалізовувати скінченний автомат їх ідентифікації.

Оцінювання роботи: 6 (завдання 1) + 7 (завдання 2) + 7 (завдання 3) = 20
Термін здачі роботи без штрафних балів: 27.02.2017 – 19.03.2017

Завдання 1. Робота з класами пакету `java.util.regex`

- за описом, наданим в індивідуальному завданні (табл. 1), побудувати регулярний вираз;
- створити текстовий файл за допомогою стандартних засобів операційної системи (наприклад, Notepad) і записати до файлу слова, кожне з яких розташоване на окремому рядку;
- використовуючи класи і методи з пакету `java.util.regex`, визначити слова у файлі, які відповідають регулярному виразу;
- створити текстовий файл за допомогою стандартних засобів операційної системи (наприклад, Notepad) і записати до файлу довільний текст;
- використовуючи класи і методи з пакету `java.util.regex`, визначити, на яких позиціях розташовані слова, які відповідають регулярному виразу.

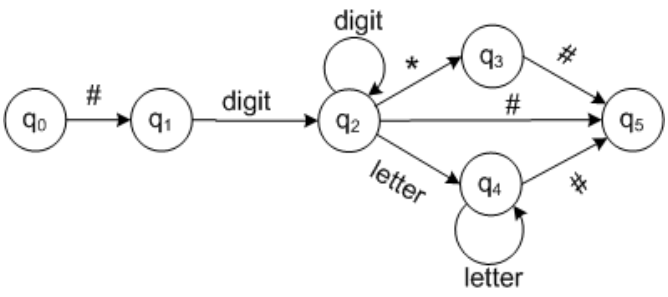
Рекомендована література:

1. Package java.util.regex
   https://docs.oracle.com/javase/8/docs/api/java/util/regex/package-summary.html
2. Guide to Regular Expressions in Java (Part 1)
   http://www.ocpsoft.org/opensource/guide-to-regular-expressions-in-java-part-1/
3. Jan Goyvaerts, Steven Levithan. Regular Expressions Cookbook, Second Edition. – 2012.
   https://drive.google.com/drive/folders/0B-9zxkXP7PZMMlgxcTNLb3M3Mzg

Завдання 2. Побудова скінченного автомату для ідентифікації текстового образу

- побудувати у вигляді графа скінченний автомат, який розпізнає текстовий образ, заданий регулярним виразом (табл. 2);

Приклад скінченного автомату для регулярного виразу ***#[4-8]+(*|[a-f]+)?#***



$Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$

$q = \{ q_0 \}$
$F = \{ q_5 \}$
$B = \{ 4, 5, 6, 7, 8 \}$
$C = \{ a, b, c, d, e, f \}$
$D1 = \{ \# \}$
$D2 = \{ * \}$
$\Sigma = P(B) \cup P(C) \cup P(D1) \cup P(D2)$

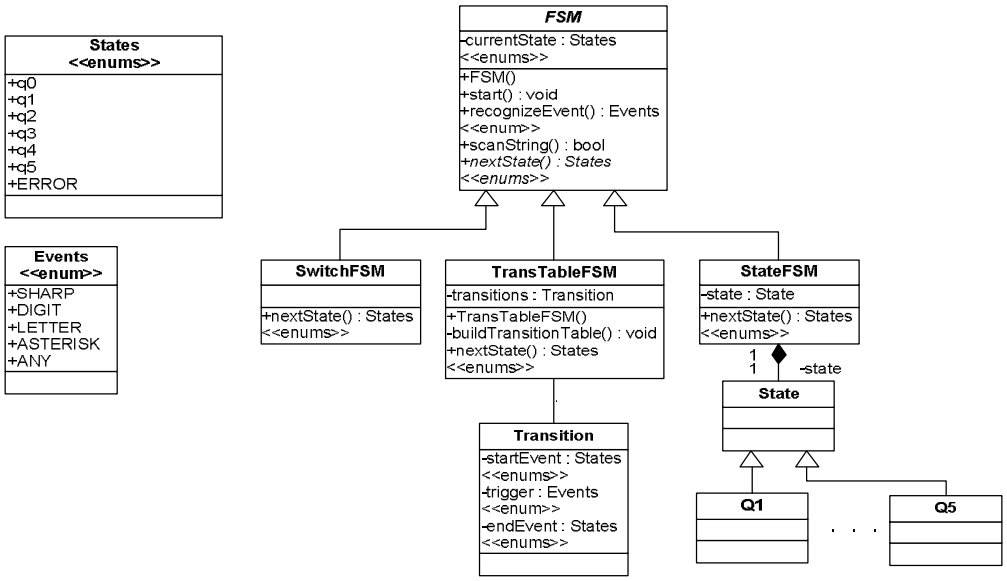
– описати таблицю переходів для побудованого скінченного автомату;

Приклад функцій переходів і таблиці переходів

δ:
$(q_0, D1) \rightarrow q_1$
$(q_1, B) \rightarrow q_2$
$(q_2, B) \rightarrow q_2$
$(q_2, D2) \rightarrow q_3$
$(q_2, C) \rightarrow q_4$
$(q_2, D1) \rightarrow q_5$
$(q_3, D1) \rightarrow q_5$
$(q_4, C) \rightarrow q_4$
$(q_4, D1) \rightarrow q_5$

Q **Σ**	**Q_0**	**Q_1**	**Q_2**	**Q_3**	**Q_4**
D1	q_1	Error	q_5	q_5	q_5
D2	Error	Error	q_3	Error	Error
B	Error	q_2	q_2	Error	Error
C	Error	Error	q_4	Error	q_4

– описати три програмні реалізації синтаксичного аналізатору, який перевіряє рядок на відповідність заданому регулярному виразу. Аналізатор реалізує скінченний автомат трьома способами: за допомогою оператору вибору switch, таблицю переходів і зразок проектування «State». Приблизна структура програми у вигляді діаграми класів UML наведена нижче:



Завдання 3. Модульне тестування реалізацій синтаксичного аналізатора засобами тестового середовища JUnit4

- описати тестові класи і методи для перевірки методів класів, які реалізують скінчений автомат;
- створити набір тестових даних для перевірки правильності роботи синтаксичного аналізатору;
- запустити тести на виконання через параметризацію конструктора (Parameterized) і параметризацію тестового метода (Theories).

Рекомендована література:

1. Сторінка проекту JUnit4: http://junit.org/junit4/
2. P. Tahchiev, F. Leme, V. Massol, and Gary Gregory. JUnit in Action https://drive.google.com/drive/folders/0B-9zxkXP7PZMMlgxcTNLb3M3Mzg

Індівідуальні завдання

Таблиця 1

Варіант	Синтаксичний устрій слова
1	**2**
1	Слово складається з двох частин, кожна з яких містить послідовність із символів «0÷9, що починаються з символу «+» та розділяються символами «$%».
2	Слово обов'язково починається символом «%», потім обов'язково йде послідовність із символів «0÷9», за якою обов'язково слідують символи «##» або «#%», потім може йти послідовність із символів «A÷Z». Закінчується слово обов'язковим символом «%».
3	Слово обов'язково починається символом «+» або «-», потім обов'язково йде послідовність із символів «5÷9», за якою може йти послідовність із символів «5÷9» або «A÷G». Закінчується слово обов'язковим символом «-».
4	Слово обов'язково починається символом «<», потім обов'язково йде символ «+» або «-», за якими слідує послідовність із символів «0÷5» або «P÷Z», наприкінці розміщується символ «>».
5	Слово обов'язково починається символом «+», потім обов'язково йде послідовність із символів «0÷9», за якою слідує послідовність із символів «0÷9» або «A÷Z».
6	Слово складається з двох частин, кожна з яких містить послідовності із символів «0÷9». Частини слова розподіляються символами «*E».
7	Слово обов'язково починається символом «{», потім можуть йти послідовність із символів «0÷9» або «A÷Z», закінчується слово символом «}».
8	Слово обов'язково починається з двох символів «/», потім може йти послідовність із символів «a÷z», закінчується слово послідовністю із символів «F÷K».
9	Слово складається з двох частин, що розподілені символом «@». Кожна частина обов'язково починається символом «#», а далі може йти послідовність із символів «0÷9» або «A÷F».
10	Слово обов'язково починається з послідовності символів «A÷Z», потім може йти символ «_», а далі послідовність із символів «A÷Z» або «0÷9».
11	Слово обов'язково починається символом «#», потім може йти послідовність із символів «0÷9», а далі обов'язково слідують символи або «%», або «*», або послідовність із символів «A÷Z». Закінчується слово символом «#».

Варіант	Синтаксичний устрій слова
1	**2**
12	Слово обов'язково починається символом «_», далі обов'язково слідує символ «+» або «-», потім послідовність із символів «A÷K».
13	Слово складається з двох частин, що розподілені символом «*», кожна частина містить послідовність із символів «A÷Z», яка обов'язково починається та закінчується символом «^».
14	Слово обов'язково починається символом «(», далі може йти послідовність із символів «A÷Z», які розділені комою, закінчується слово символом «)».
15	Слово починається символом «[», далі йде символ «+» або символ «-», за якими слідує послідовність із символів «0÷9» або «A÷Z», наприкінці розміщується символ «]».
16	Слово починається символом «{» або «(», потім може йти послідовність із символів «0÷9» або «A÷Z». Закінчуватися слово може одним або декількома символами «}».
17	Слово починається символом «%», далі йде послідовність із символів «0÷9», а потім символи «~~» або «~%», за якими послідовність із символів «A÷Z». Слово може закінчуватися символом «%».
18	Слово починається символом «(», далі може йти послідовність із символів «5÷9», потім символ «%» або «*», а далі послідовність із символів «A÷Z». Закінчується слово символом «)».
19	Слово починається символом «0», закінчується символом «1». Слово містить дві частини з послідовностей символів «0÷9», які розподілені символом «!».
20	Слово починається символом «+», далі обов'язково йде послідовність із символів «0÷9», а далі послідовність із символів «A÷K» або «^» та «&».
21	Слово може починатися символами «0÷9», далі обов'язково йде крапка і символи «0÷9»
22	Слово складається з послідовностей різної довжини з символів «k÷s», які розділені «,» , «–» або «:». Остання послідовність символів «k÷s» закінчується «.»

Таблиця 2

Варіант	Регулярний вираз	Варіант	Регулярний вираз
1	**2**	**3**	**4**
1	\++([\D]?\d*	12	\w+\.(\w+)?.#
2	\+[0-9]+\+%\+[0-9]+	13	\d+(%|*|([A-Z])+)#
3	_\d+#(#|&)[A-Z]*%	14	_(\+|-)[A-K]+\d{1,3}
4	(\+|-)[5-9]+([0-4]*|[AG]*)-	15	\^[A-Z]+\^*\^[^AZ^\d]+\^
5	<(\+|-)([0-5]+)|([P-Z])+>	16	\(([A-Z]*,\s)*[A-Z]*\)
6	+\d+[A-Z]*	17	\$([A-F]+|\d)*[^\d]+
7	(\d+E)+\d	18	(\d+\!)+\d+(e!n!d)*
8	\{(\d+|[A-Z]+)\}	19	([^A-Z])+(%|*|([AZ])+)#
9	\/[a-z]*[F-K]+	20	[A-Z]*_?([A-Z]+|\d+)
10	#(\d*|[a-f]*)@(\d*|[a-f]*)	21	(\+|-)[0-4]+([5-9]*|[al]*)-
11	[A-Z]+_?([A-Z]+|\d+)	22	\\d+(%|\~|([a-t])+)#