

## Лабораторная работа №3

### «Работа со строками»

**Цель работы:** изучить работу со строками в языке программирования Java.

**В лабораторной работе необходимо выполнить 4 (четыре) задания:**

I. К слою доступа к данным, разработанном в лабораторной работе №2 добавить:

Конфигурацию (логин/пароль для БД, строку подключения и т.д.) и запросы SQL перенести в файлы .properties. Для их чтения использовать класс ConfigurationManager построенный с использованием шаблона Singleton.

II. Добавить вывод сообщений в лог-файл с использованием логгера log4j. Во всех блоках try-catch заменить System.out.println на вывод в лог файл.

III. Написать программу для выполнения следующих действий с регулярными выражениями (по вариантам):

1. Написать регулярное выражение, определяющее является ли данная строка строкой "abcdefghijklmnpqrstuv18340" или нет.

– пример правильных выражений: abcdefghijklmnpqrstuv18340.

– пример неправильных выражений: abcdefghijklmnoasdfasdpqrstuv18340.

2. Написать регулярное выражение, определяющее является ли данная строка GUID с или без скобок. Где GUID это строчка, состоящая из 8, 4, 4, 4, 12 шестнадцатеричных цифр разделенных тире.

– пример правильных выражений: e02fd0e4-00fd-090A-ca30-0d00a0038ba0.

– пример неправильных выражений: e02fd0e400fd090Aca300d00a0038ba0.

3. Написать регулярное выражение, определяющее является ли заданная строка правильным MAC-адресом.

– пример правильных выражений: aE:dC:cA:56:76:54.

– пример неправильных выражений: 01:23:45:67:89:Az.

4. Написать регулярное выражение, определяющее является ли данная строчка валидным URL адресом. В данной задаче правильным URL считаются адреса http и https, явное указание протокола также может отсутствовать. Учитываются только адреса, состоящие из символов, т.е. IP адреса в качестве URL не присутствуют при проверке. Допускаются поддомены, указание порта доступа через двоеточие, GET запросы с передачей параметров, доступ к подпапкам на домене, допускается наличие якоря через решетку. Однобуквенные домены считаются

запрещенными. Запрещены спецсимволы, например «—» в начале и конце имени домена. Запрещен символ «\_» и пробел в имени домена. При составлении регулярного выражения ориентируйтесь на список правильных и неправильных выражений заданных ниже.

- пример правильных выражений: <http://www.example.com>, <http://example.com>.

- пример неправильных выражений: Just Text, <http://a.com>.

5. Написать регулярное выражение, определяющее является ли данная строчка шестнадцатиричным идентификатором цвета в HTML. Где #FFFFFF для белого, #000000 для черного, #FF0000 для красного и т.д.

- пример правильных выражений: #FFFFFF, #FF3421, #00ff00.

- пример неправильных выражений: 232323, f#fddee, #fd2.

6. Написать регулярное выражение, определяющее является ли данная строчка датой в формате dd/mm/уууу. Начиная с 1600 года до 9999 года.

- пример правильных выражений: 29/02/2000, 30/04/2003, 01/01/2003.

- пример неправильных выражений: 29/02/2001, 30-04-2003, 1/1/1899.

7. Написать регулярное выражение, определяющее является ли данная строчка валидным E-mail адресом согласно RFC под номером 2822.

- пример правильных выражений: user@example.com, root@localhost

- пример неправильных выражений: bug@@@com.ru, @val.ru, Just Text2.

8. Составить регулярное выражение, определяющее является ли заданная строка IP адресом, записанным в десятичном виде.

- пример правильных выражений: 127.0.0.1, 255.255.255.0.

- пример неправильных выражений: 1300.6.7.8, abc.def.gha.bcd.

9. Проверить, надежно ли составлен пароль. Пароль считается надежным, если он состоит из 8 или более символов. Где символом может быть английская буква, цифра и знак подчеркивания. Пароль должен содержать хотя бы одну заглавную букву, одну маленькую букву и одну цифру.

- пример правильных выражений: C00l\_Pass, SupperPas1.

- пример неправильных выражений: Cool\_pass, C00l.

10. Проверить является ли заданная строка шестизначным числом, записанным в десятичной системе счисления без нулей в старших разрядах.

- пример правильных выражений: 123456, 234567.

- пример неправильных выражений: 1234567, 12345.

11. Есть текст со списками цен. Извлечь из него цены в USD, RUR, EU.

- пример правильных выражений: 23.78 USD.
  - пример неправильных выражений: 22 UDD, 0.002 USD.
12. Проверить существуют ли в тексте цифры, за которыми не стоит «+».
- пример правильных выражений:  $(3 + 5) - 9 \times 4$ .
  - пример неправильных выражений:  $2 * 9 - 6 \times 5$ .
13. Создать запрос для вывода только правильно написанных выражений со скобками (количество открытых и закрытых скобок должно быть одинаково).
- пример правильных выражений:  $(3 + 5) - 9 \times 4$ .
  - пример неправильных выражений:  $((3 + 5) - 9 \times 4$ .

IV. Создать программу обработки текста учебника по программированию с использованием классов: **Символ, Слово, Предложение, Знак препинания и др.** Во всех задачах с формированием текста заменять табуляции и последовательности пробелов одним пробелом. Для выделения нужных элементов из текста использовать регулярные выражения.

1. Найти наибольшее количество предложений текста, в которых есть одинаковые слова.
2. Вывести все предложения заданного текста в порядке возрастания количества слов в каждом из них.
3. Найти такое слово в первом предложении, которого нет ни в одном из остальных предложений.
4. Во всех вопросительных предложениях текста найти и напечатать без повторений слова заданной длины.
5. В каждом предложении текста поменять местами первое слово с последним, не изменяя длины предложения.
6. Напечатать слова текста в алфавитном порядке по первой букве. Слова, начинающиеся с новой буквы, печатать с красной строки.
7. Рассортировать слова текста по возрастанию доли гласных букв (отношение количества гласных к общему количеству букв в слове).
8. Слова текста, начинающиеся с гласных букв, рассортировать в алфавитном порядке по первой согласной букве слова.
9. Все слова текста рассортировать по возрастанию количества заданной буквы в слове. Слова с одинаковым количеством расположить в алфавитном порядке.
10. Существует текст и список слов. Для каждого слова из заданного списка найти, сколько раз оно встречается в каждом предложении, и рассортировать слова по убыванию общего количества вхождений.
11. В каждом предложении текста исключить подстроку максимальной длины, начинающуюся и заканчивающуюся заданными символами.

12. Из текста удалить все слова заданной длины, начинающиеся на согласную букву.
13. Отсортировать слова в тексте по убыванию количества вхождений заданного символа, а в случае равенства – по алфавиту.
14. В заданном тексте найти подстроку максимальной длины, являющуюся палиндромом, т.е. читающуюся слева направо и справа налево одинаково.
15. Преобразовать каждое слово в тексте, удалив из него все последующие (предыдущие) вхождения первой (последней) буквы этого слова.
16. В некотором предложении текста слова заданной длины заменить указанной подстрокой, длина которой может не совпадать с длиной слова.