



savchenkojtr

| [log out](#)[Jenkins](#)[All](#)[task-jtr-practice3](#)[ENABLE AUTO REFRESH](#)[Back to Dashboard](#)[Status](#)[Changes](#)**Build History****(trend)**[RSS for all](#)[RSS for failures](#)

Project task-jtr-practice3

Практическое занятие №3

ВНИМАНИЕ!

- (1) указаны имена классов
- (2) указаны имена методов
- (3) указаны имена файлов со входной информацией;
- (4) во входной информации может быть кириллица (буквы русского и украинского языков);

Замечание.

Если приложение считывает информацию из файла, то необходимо указать кодировку, в которой она (информация) записана.

См. пример "Как получить входную информацию" в конце текста.

В корневом пакете создать класс Демо, который демонстрирует работу всего написанного функционала.

Задание 1

Название класса: ua.nure.your_last_name.Practice3.Part1
Входную информацию загружать из файла part1.txt

Определить класс со статическими методами, которые преобразовывают входную информацию в выходную.

В качестве входной информации (input data) использовать текст следующей структуры (значения Login/Name/Email в общем случае могут быть любыми):

Input data (part1.txt)

Login;Name;Email
ivanov;Ivan Ivanov;ivanov@mail.ru
петров;Петр Петров;petrov@google.com
obama;Barack Obama;obama@google.com
bush;Джордж Буш;bush@mail.ru

Методы, которые нужно написать, имеют следующий вид (N - цифра: 1, 2, 3, 4):

public static String convertN(String input) {
 ...
}

1.1. Метод convert1

Должен преобразовывать input data в строку следующего вида:

Output of convert1

ivanov ==> ivanov@mail.ru
петров ==> petrov@google.com
obama ==> obama@google.com
bush ==> bush@mail.ru

1.2. Метод convert2

Должен преобразовывать input data в строку следующего вида:

Jenkins	All	task-jtr-practice3	ENABLE AUTO REFRESH
---------	-----	--------------------	---------------------

```
-----
Ivanov Ivan (email: ivanov@mail.ru)
Петров Петр (email: petrov@google.com)
Obama Barack (email: obama@google.com)
Буш Джордж (email: bush@mail.ru)
-----
```

1.3. Метод convert3

Должен преобразовывать input data в строку следующего вида (почтовый домен ==> список логинов через запятую тех пользователей, чьи почтовые ящики зарегистрированы в данном домене):

Output of convert3

```
-----
mail.ru ==> ivanov, bush
google.com ==> петров, obama
-----
```

1.4. Метод convert4

Должен преобразовывать input data в строку следующего вида (должна быть добавлена колонка Password, сам пароль должен состоять ровно из 4 цифр, которые генерируются случайным образом):

Output of convert4

```
-----
Login;Name;Email;Password
ivanov;Ivan Ivanov;ivanov@mail.ru;1707
петров;Петр Петров;petrov@google.com;9321
obama;Barack Obama;obama@google.com;4623
bush;Джордж Буш;bush@mail.ru;7514
-----
```

Задание 2

```
-----
Название класса: ua.nure.your_last_name.Practice3.Part2
Входную информацию загружать из файла part2.txt
-----
```

Вход: текст (может состоять из латиницы и кириллицы).

Выход: слова максимальной и минимальной длины в формате, который дан ниже (в единственном числе и в порядке их появления в тексте).
Словом считать последовательность содержащую только буквы (все остальные символы в состав слова не входят).

Создать статический метод convert, который преобразовывает вход в выход.

Заглушка метода

```
-----
public static String convert(String input) {
    ...
}
-----
```

Пример Input data (part2.txt)

```
-----
When I was younger, so much younger than today
I never needed anybody's help in any way
But now these days are gone, I'm not so self-assured
Now I find I've changed my mind
I've opened up the doors
-----
```

Пример Output

```
-----
Min: I, s, m
Max: younger, anybody, assured, changed
-----
```

Jenkins	All	task-jtr-practice3	ENABLE AUTO REFRESH
		<div>-----</div> <div>Задание 3</div> <div>-----</div> <div>Название класса: ua.nure.your_last_name.Practice3.Part3</div> <div>Входную информацию загружать из файла part3.txt</div> <div>-----</div> <div>Вход: текст (может состоять из латиницы и кириллицы).</div> <div>Выход: исходный текст, первый символ каждого слова которое состоит из двух и более символов в верхнем регистре</div> <div>Словом считать последовательность содержащую только буквы (все остальные символы в состав слова не входят).</div> <div>Создать статический метод convert, который преобразовывает вход в выход.</div> <div>Заглушка метода</div> <div>-----</div> <div>public static String convert(String input) { ... }</div> <div>-----</div> <div>Пример Input data</div> <div>-----</div> <div>When I was younger I never needed</div> <div>-----</div> <div>Пример Output</div> <div>-----</div> <div>When I Was Younger I Never Needed</div> <div>-----</div> <div>-----</div> <div>Задание 4</div> <div>-----</div> <div>Название класса: ua.nure.your_last_name.Practice3.Part4</div> <div>-----</div> <div>Для хеширования информации (например, паролей) используют метод MessageDigest#digest, который возвращает хеш в виде массива байт.</div> <div>Пример хеширования пароля с помощью алгоритма хеширования MD5 (другие алгоритмы - SHA-256; SHA-512 и пр.)</div> <div>-----</div> <div>import java.security.*; import java.util.Arrays; public class HashExample { public static void main(String[] args) throws NoSuchAlgorithmException { MessageDigest digest = MessageDigest.getInstance("MD5"); digest.update("password to hash".getBytes()); byte[] hash = digest.digest(); System.out.println(Arrays.toString(hash)); // output: [56, 55, 83, 50, 113, -114, -54, 115, -125, 86, 79, -109, 17, -65, 107, 84] } }</div> <div>-----</div> <div>Написать статический метод hash, который на вход принимает два параметра: (1) строку, хеш которой нужно получить; (2) названия алгоритма хеширования.</div> <div>Выход должен представлять из себя строку из шестнадцатеричных цифр: каждому байту соответствует две шестнадцатеричные цифры.</div>	

Например, если некоторый элемент массива байт равен -29, то в двоичном разложении он имеет вид 1110_0011 и ему соответствует пара Е3.

Jenkins

All

task-jtr-practice3

[ENABLE AUTO REFRESH](#)

```
import java.security.*;

public class Part4 {
    public static String hash(String input, String algorithm) throws
    NoSuchAlgorithmException {
        // place yhour code here
        return null;
    }
    public static void main(String[] args) throws NoSuchAlgorithmException
    {
        System.out.println(hash("password", "SHA-256"));
        System.out.println(hash("passwort", "SHA-256"));
    }
}
```

Задание 5

Название класса: ua.nure.your_last_name.Practice3.Part5

Создать класс с двумя статическими методами перевода из десятичной системы счисления в римскую и обратно.

```
public static String decimal2Roman(int x) { ... }
public static int roman2Decimal(String s) { ... }
```

Рабочий диапазон методов - от 1 до 100 включительно.

Работу методов продемонстрировать так:

DECIMAL ==decimal2Roman==> ROMAN ==roman2Decimal==> DECIMAL

```
1 ==> I ==> 1
2 ==> II ==> 2
3 ==> III ==> 3
4 ==> IV ==> 4
5 ==> V ==> 5
...
94 ==> XCIV ==> 94
95 ==> XCV ==> 95
96 ==> XCVI ==> 96
97 ==> XCVII ==> 97
98 ==> XCVIII ==> 98
99 ==> XCIX ==> 99
100 ==> C ==> 100
```

Брут-форс (полный перебор) не допускается!

Продумать алгоритм и запрограммировать.

Решение которое использует массив из ста элементов исключить из рассмотрения:

```
String[] numbers = {"I", "II", "III", "IV", "V", ..., "XCV", "XCVI",
"XCVII", ..., "C"}
```

Как получить входную информацию (файл должен находится в корне проекта, кодировка файла - Cp1251)

```
package ua.nure.your_last_name.Practice3;
```

```
import java.io.IOException;
```

Jenkins

All

task-jtr-practice3

[ENABLE AUTO REFRESH](#)

```
public class Util {  
  
    private static final String ENCODING = "Cp1251";  
  
    public static String readFile(String path) {  
        String res = null;  
        try {  
            byte[] bytes = Files.readAllBytes(Paths.get(path));  
            res = new String(bytes, ENCODING);  
        } catch (IOException ex) {  
            ex.printStackTrace();  
        }  
        return res;  
    }  
  
    public static void main(String[] args) {  
        System.out.println(readFile("part1.txt"));  
    }  
}
```

Project disk usage information + trend graph



Disk Usage: Workspace 0, Builds {all=0, locked=0}, Job directory 22149



[Recent Changes](#)

Permalinks



[Help us localize this page](#)

Page generated: 21 жовт 2018 15:32:04

[REST API](#)

[Jenkins ver. 1.540](#)