**HW 10 for Beginners**

**Part 1**

* Все задания в первой части нужно выполнять в проекте Java for beginners (без фреймворка Maven + JUnit 5.7.2).
* Создайте класс HW10.
* Смотрите видео по каждому методу и решайте задание с помощью метода из видео.

**Видео1**: <https://youtu.be/78ZAzmNbUYQ?t=173>  
**Методы isEmpty(), length(), trim(), replace()**

1. Написать метод, который принимает на вход строку.   
   Если строка не пустая (проверить методом из видео), то примените метод по удалению пробелов в начале строки и в конце строки.   
   Догадаться, каким методом из видео можно проверить, были ли пробелы.  
   Если пробелы были, то метод должен вернуть сообщение: “Лишние пробелы удалены”.

Если пробелов не было, вернуть сообщение “Пробелов не было”.

Если строка пустая, вернуть сообщение “Строка пустая”.

Test Data:

“ QA4Everyone “ → “Лишние пробелы удалены”

“QA4Everyone“ → “Пробелов не было”  
“” → “Строка пустая”

1. Написать алгоритм RemoveAlla.  
   С помощью методов из видео1, написать алгоритм, который принимает на вход строку. Если строка валидная, то метод удаляет все буквы **a** из строки, если таковые имеются. Метод возвращает обработанную строку.

*Test Data:*

“ QA4Everyone “ → “QA4Everyone“

“panda “ → “pnd”

1. Написать алгоритм RemoveAllZeros.  
   С помощью методов из видео1, написать алгоритм, который принимает на вход строку, состоящую из цифр. Если строка валидная, то метод удаляет все пробелы из строки, если таковые имеются. Метод возвращает обработанную строку, в которой нет нулей.

*Test Data:*

“ 3 5 0 4 2 0 9 7 0 6 0 4 0 0 0 0 “ → “35429764“

“ 0000000111“ → “111”

1. Написать алгоритм RemoveAllSpaces.  
   С помощью методов из видео1, написать алгоритм, который принимает на вход строку. Если строка валидная, то метод удаляет все пробелы из строки, если таковые имеются. Метод возвращает обработанную строку.

*Test Data:*

“ QA 4 Everyone “ → “QA4Everyone“

“p a n d a “ → “panda”

**Видео2:** <https://www.youtube.com/watch?v=thzfsPEYXfI>  
**Методы charAt(), concat(), contains(), substring()**

**toLowerCase(), toUpperCase()**

1. Напишите метод, который принимает на вход строку и считает, сколько букв а или А содержится в строке.

*Test Data:*“Abracadabra” → 5  
“Homenum Revelio” → 0

1. Напишите метод, который принимает на вход текст и проверяет, содержится ли в тексте хотя бы одно слово Java.  
   *Test Data:  
   “As of March 2022,* [*Java 18*](https://en.wikipedia.org/wiki/Java_version_history) *is the latest version, while Java 17, 11 and 8 are the current* [*long-term support*](https://en.wikipedia.org/wiki/Long-term_support) *(LTS) versions. Oracle released the last zero-cost public update for the* [*legacy*](https://en.wikipedia.org/wiki/Legacy_system) *version* [*Java 8*](https://en.wikipedia.org/wiki/Java_8) *LTS in January 2019 for commercial use, although it will otherwise still support Java 8 with public updates for personal use indefinitely. Other vendors have begun to offer* [*zero-cost builds*](https://en.wikipedia.org/wiki/OpenJDK#OpenJDK_builds) *of OpenJDK 8 and 11 that are still receiving security and other upgrades.”* → true  
     
   “As a decrepit father takes delight  
   To see his active child do deeds of youth,  
   So I, made lame by fortune’s dearest spite,  
   Take all my comfort of thy worth and truth.  
   For whether beauty, birth, or wealth, or wit,  
   Or any of these all, or all, or more,  
   Entitled in thy parts do crownèd sit,  
   I make my love engrafted to this store.  
   So then I am not lame, poor, nor despised,  
   Whilst that this shadow doth such substance give  
   That I in thy abundance am sufficed,  
   And by a part of all thy glory live.  
   Look what is best, that best I wish in thee.  
   This wish I have; then ten times happy me.” → false
2. Напишите метод, который принимает на вход строку, и добавляет Кавычки в начале строки, точку и кавычки в конце строки с помощью метода concat()  
   *Test Data:*“One” → ““One.””  
   “ TWO “ → ““TWO.””
3. Напишите метод, кторый принимает на вход название города и исправляет написание:

*Test Data:*  
“ташкент” → “Ташкент”  
“ЧикаГО” → “Чикаго”

Видео3: <https://www.youtube.com/watch?v=6_RDHZfygGo>  
**indexOf(), lastIndexOf()**

1. Напишите метод, который принимает на вход строку и букву-параметр, и возвращает все, что находится между первой и последней буквой-параметром:  
   *Test Data:*  
   “Abracadabra”, “b” → “bracadab”  
   “Whippersnapper”, “p” → “ppersnapp”
2. Напишите метод, который принимает на вход слово, и возвращает true, если слово начинается и заканчивается на одинаковую букву, и faulse иначе

*Test Data:*  
 “Abracadabra” → true  
 “Whippersnapper” → false

Видео4: <https://youtu.be/SBByYcrsXu0?t=42>  
  
**split()**

1. Напишите метод, который принимает на вход предложение и возвращает слова из этого предложения в виде массива слов  
   *Test Data:*  
   “QA for Everyone” → {“QA”, “for”, “Everyone”}

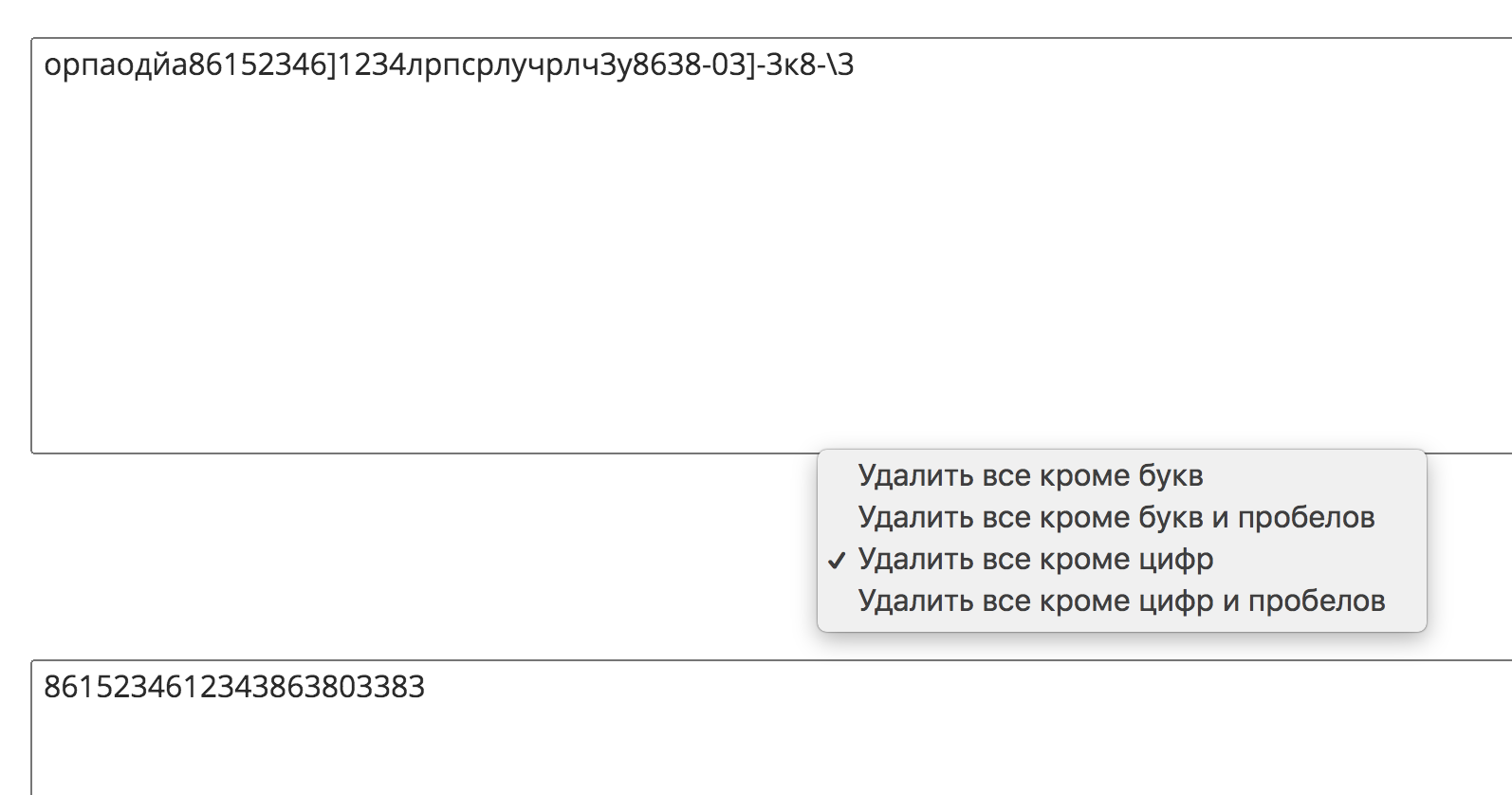
“Александр Сергеевич Пушкин” → {“Александр”, “Сергеевич”, “Пушкин”}

1. Написать метод, который принимает на вход предложение, которое состоит из имени, фамилии, отчества и возвращает массив строк:  
   *Test Data:*  
   “Александр Сергеевич Пушкин” →   
   {“Имя: Александр”, “Отчество: Сергеевич”, “Фамилия: Пушкин”}
2. Написать метод, который возвращает сумму всех букв слова  
     
   “abc” → 294  
   “ABC” → 198  
   “123” → 0 (это не буквы)
3. Написать метод, который принимает на вход 2 буквы и возвращает true, если первая буква встречается раньше второй, иначе метод возвращает false

method(“a”, “m”) → true  
method(“m”, “l”) → false

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Part 2 -  **Maven, JUnit 5.7.2  
Все методы проверить тестами**

1. Написать алгоритм StringToNumbers, который принимает строку, и возвращает массив чисел:  
   *Test Data:*  
   “1, 2, 3, 4, 5” → {1, 2, 3, 4, 5}  
   (Без форматирования)
2. Написать 4 алгоритма StringToLetters, StringToLettersAndSpaces, StringToNumbers, StringToNumbersAndSpaces следуя примеру на сайте: <https://allcalc.ru/node/2030>  
   Каждый алгоритм принимает строку, и удаляет ненужные символы:  
     
     
   Без форматирования, пожалуйста!
3. Напишите алгоритм CountWords, который принимает на вход текст и слово, и считает все вариации этого слова в тексте:  
   *“As of March 2022,* [*Java 18*](https://en.wikipedia.org/wiki/Java_version_history) *is the latest version, while Java 17, 11 and 8 are the current* [*long-term support*](https://en.wikipedia.org/wiki/Long-term_support) *(LTS) versions. Oracle released the last zero-cost public update for the* [*legacy*](https://en.wikipedia.org/wiki/Legacy_system) *version* [*Java 8*](https://en.wikipedia.org/wiki/Java_8) *LTS in January 2019 for commercial java use, although it will otherwise still support Java 8 with public updates for personal use indefinitely. Other vendors have begun to offer* [*zero-cost builds*](https://en.wikipedia.org/wiki/OpenJDK#OpenJDK_builds) *of OpenJDK 8 and 11 that are still receiving security and other upgrades.”, “Java” → 5  
     
   “As of March 2022,* [*Java 18*](https://en.wikipedia.org/wiki/Java_version_history) *is the latest version, while Java 17, 11 and 8 are the current* [*long-term support*](https://en.wikipedia.org/wiki/Long-term_support) *(LTS) versions. Oracle released the last zero-cost public update for the* [*legacy*](https://en.wikipedia.org/wiki/Legacy_system) *version* [*Java 8*](https://en.wikipedia.org/wiki/Java_8) *LTS in January 2019 for commercial java use, although it will otherwise still support Java 8 with public updates for personal use indefinitely. Other vendors have begun to offer* [*zero-cost builds*](https://en.wikipedia.org/wiki/OpenJDK#OpenJDK_builds) *of OpenJDK 8 and 11 that are still receiving security and other upgrades.”, “version” → 2*
4. Написать алгоритм StartsWithLetter, который принимает на вход текст, и возвращает в виде строки все слова, которые начинаются с буквы l.   
   “As a decrepit father takes delight  
   To see his active child do deeds of youth,  
   So I, made lame by fortune’s dearest spite,  
   Take all my comfort of thy worth and truth.  
   For whether beauty, birth, or wealth, or wit,  
   Or any of these all, or all, or more,  
   Entitled in thy parts do crownèd sit,  
   I make my love engrafted to this store.  
   So then I am not lame, poor, nor despised,  
   Whilst that this shadow doth such substance give  
   That I in thy abundance am sufficed,  
   And by a part of all thy glory live.  
   Look what is best, that best I wish in thee.  
   This wish I have; then ten times happy me.” → “lame, love, lame, live, Look”
5. Написать алгоритм ReverseString, который переворачивает строку:  
   *Test Data:*  
   “Abracadabra” → “arbadacarbA”
6. Написать алгоритмRemoveDuplicates, который принимает строку и возвращает строку уникальных букв:  
   “AABBCCaabbcc” → “abc”
7. Написать алгоритм StringIntersection, который принимает на вход 2 строки и возвращает общее вхождение в этих строках  
   *Test Data:*  
   “Carrot”, “carwash” → “car”  
   “Noob”, “Schmooze” → “oo”
8. Написать алгоритм LongestCommonSubstring, который принимает 3 строки и ищет самое длинное общее пересечение:  
     
   “Smaze”, “amazing”, “amazon” → “maz”