**МОЛДАВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Факультет Математики и Информатики**

**Департамент Информатики**

Лабораторная работа № 3

Программирование на JAVA

Обработка строк. Коллекции. Регулярные выражения.

Проверил: профессор, др. Gheorghe Latu

Выполнил: Mamaliga Artur grupa I2302-1

Кишинев, 2024

# Задание

При решении задач использовать: класс String, StringBuffer, коллекции, настраиваемые коллекции, регулярные выражения. Избегать, по возможности, “мелкой” работы на уровне отдельных символов.  
Дана строка символов. Написать программу выборки из строки i-го слова. Между словами может быть произвольное число пробелов и знаков препинания. Найти слова, начинающиеся на букву ‘к’ (при помощи регулярных выражений). Из слов создать коллекцию и отсортировать её.

# Решение

Программа написана на языке Java и использует классы для работы со строками, регулярные выражения для нахождения слов, начинающихся на букву "к", и коллекции для хранения и сортировки слов.

## Program.java

import java.util.Scanner;  
  
public class Program {  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.in);  
 System.out.println("[-]ENTER THE STRING: ");  
 String text = sc.nextLine();  
  
 System.out.println("[-]TYPE INDEX OF WORD YOU WANT TO GET FROM ZERO: ");  
 int index = sc.nextInt();  
  
 System.out.println("[+]WORD BY INDEX: "+index);  
 System.out.println(TextUtils.getWordByIndex(text,index));  
 System.out.println("[+]NUMBER OF WORDS IN TEXT: ");  
 System.out.println(TextUtils.countWords(text));  
 System.out.println("[+]WORDS ON LETTER K: ");  
 String[] wordsOnK = TextUtils.findWordsOnK(text);  
 for (String word : wordsOnK) {  
 System.out.println(word);  
 }  
  
 System.out.println("[+]WORDS SORTED: ");  
 String[] words\_sorted = TextUtils.sortWordsFromText(text);  
 for (String word : words\_sorted) {  
 System.out.println(word);  
 }  
 }  
}

Класс TextUtils содержит утилитарные методы для работы со строками текста. Эти методы включают функции для разбиения текста на слова, извлечения слова по индексу, подсчета количества слов, поиска слов, начинающихся на букву 'K' или 'k', и сортировки слов.

## Описание методов

### Метод getWords(String str)

Метод getWords принимает строку str и разбивает её на массив слов. Разделителями являются пробелы, запятые, точки, восклицательные и вопросительные знаки, двоеточия и точки с запятой.  
  
Параметры:  
- String str — строка, которая будет разбита на слова.  
Возвращаемое значение:  
- String[] — массив слов, полученных из строки.  
  
Код метода:  
public static String[] getWords(String str) {  
 return str.split("[\\s,\\.!?:;]+");  
}

### Метод getWordByIndex(String str, int index)

Этот метод возвращает слово по указанному индексу из строки str. Он использует метод getWords, чтобы сначала получить массив слов, а затем возвращает слово по заданному индексу.  
  
Параметры:  
- String str — строка, из которой извлекается слово.  
- int index — индекс слова, которое необходимо вернуть.  
Возвращаемое значение:  
- String — слово, находящееся на позиции index.  
  
Код метода:  
public static String getWordByIndex(String str, int index) {  
 String[] words = getWords(str);  
 return words[index];  
}

### Метод countWords(String str)

Метод countWords возвращает количество слов в строке str. Он использует метод getWords для разбиения строки на слова и затем возвращает длину полученного массива.  
  
Параметры:  
- String str — строка, в которой подсчитываются слова.  
Возвращаемое значение:  
- int — количество слов в строке.  
  
Код метода:  
public static int countWords(String str) {  
 String[] words = getWords(str);  
 return words.length;  
}

### Метод findWordsOnK(String str)

Метод findWordsOnK ищет все слова в строке str, которые начинаются с буквы 'K' или 'k'. Он использует регулярное выражение \b[Kk]\w\* для поиска слов и сохраняет найденные слова в списке ArrayList, который затем преобразуется в массив строк.  
  
Параметры:  
- String str — строка, в которой ищутся слова на 'K' или 'k'.  
Возвращаемое значение:  
- String[] — массив слов, начинающихся с 'K' или 'k'.  
  
Код метода:  
public static String[] findWordsOnK(String str) {  
 ArrayList<String> words = new ArrayList<>();  
 String regex = "\\b[Kk]\\w\*";  
 Pattern pattern = Pattern.compile(regex);  
 Matcher matcher = pattern.matcher(str);  
 while (matcher.find()) {  
 words.add(matcher.group());  
 }  
 return words.toArray(new String[0]);  
}

### Метод sortWordsFromText(String str)

Метод sortWordsFromText принимает строку str, разбивает её на слова, сортирует их в лексикографическом порядке и возвращает отсортированный массив слов. Метод использует Collections.sort для упорядочивания и преобразует отсортированный список обратно в массив строк.  
  
Параметры:  
- String str — строка, слова в которой будут отсортированы.  
Возвращаемое значение:  
- String[] — массив отсортированных слов.  
  
Код метода:  
public static String[] sortWordsFromText(String str) {  
 String[] wordsArray = getWords(str);  
 ArrayList<String> words = new ArrayList<>(Arrays.asList(wordsArray));  
 Collections.sort(words);  
 return words.toArray(new String[0]);  
}