

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧЕРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
Кафедра ИИТ

Лабораторная работа №1

Выполнил:
студента группы АС-50
Гиголян А. О.
Проверил:
Крощенко А. А.

Брест 2020

Задание 1:

3) Вывод моды последовательности. Модой ряда чисел называется число, которое встречается в данном ряду чаще других. Последовательность может иметь более одной моды, а может не иметь ни одной.

Задание 2:

3) Написать метод `shiftRight(double[] array, int shift)`, который сдвигает элементы массива `array` на заданное число позиций `shift` вправо.

Задание 3:

3) Напишите метод `boolean pangramEng(String str)`, проверяющий, является ли строка панграммой или нет. Панграмма – это такая строка, которая содержит все или почти все буквы алфавита, по возможности не повторяя их.

```
package com.company;

import java.util.*;
import java.util.stream.Collectors;
import java.util.stream.IntStream;
import java.util.stream.Stream;

public class Main {

    public static void main(String[] args) {

        calculateMode(Arrays.stream(args).flatMapToInt(num ->
            IntStream.of(Integer.parseInt(num))).toArray());
        double[] array = new double[]{1, 2, 3, 4, 5};
        shiftRight(array, 2);
        System.out.println(Arrays.toString(array));
        System.out.println(check("Аэрофотосъёмка ландшафта уже выявила земли
        богачей и процветающих крестьян"));
    }

    private static void calculateMode(int[] numbers) {
        Map<Integer, Integer> map = new HashMap<>();
        for (int element : numbers) {
            if (!map.containsKey(element))
                map.put(element, 1);
            else
                map.put(element, map.get(element) + 1);
        }

        Set<Integer> results;
        int maxCount = map
            .entrySet()
            .stream()
            .max(Map.Entry.comparingByValue())
            .get()
            .getValue();

        if (maxCount <= 1)
            System.out.println("Row doesn't have mode");
    }
}
```

```

        else {
            results = keys(map, maxCount).collect(Collectors.toSet());
            System.out.println(results);
        }
    }

    private static Stream<Integer> keys(Map<Integer, Integer> map, Integer
value) {
        return map
            .entrySet()
            .stream()
            .filter(entry -> value.equals(entry.getValue()))
            .map(Map.Entry::getKey);
    }

    //task 2
    public static void shiftRight(double[] array, int shift) {
        double[] resultArray = new double[array.length];
        for (int i = 0, length = array.length; i < length; i++) {
            if (i + shift < length)
                resultArray[i + shift] = array[i];
            else
                resultArray[i + shift - length] = array[i];
        }
        System.arraycopy(resultArray, 0, array, 0, resultArray.length);
    }

    //task 3
    public static boolean check(String str) {
        for(char i = 'a'; i <= 'я'; i++) {
            if(!str.contains(String.valueOf(i).toLowerCase())) {
                return false;
            }
        }
        return true;
    }
}

```

Program arguments: 1 2 3 4 2 4

```

C:\Users\Lenovo\.jdk\openjdk-15.0.1\bin\jav
[2, 4]
[4.0, 5.0, 1.0, 2.0, 3.0]
true

Process finished with exit code 0

```