

Задание 1. Исследование List.

```
1 package ru.jrp.generator;
2
3 import java.io.FileWriter;
4 import java.io.IOException;
5
6 import static ru.jrp.Main.rand;
7
8 public class RandomNumbers { 2 usages new *
9
10     private RandomNumbers() {} no usages new *
11
12     public static void generate() { 1 usage new *
13         StringBuilder stringBuilder = new StringBuilder();
14
15         for (int i = 0; i < 100; i++) {
16             stringBuilder.append(rand.nextInt(bound: 100) + 1);
17             if (i < 99) {
18                 stringBuilder.append(", ");
19             }
20         }
21
22         try (FileWriter fileWriter = new FileWriter(fileName: "inputDataOfNumbers.txt")) {
23             fileWriter.write(stringBuilder.toString());
24             System.out.println("Файл успешно создан");
25         } catch (IOException e) {
26             System.out.println(e.getMessage());
27         }
28     }
29 }
30
```

```
11 public class Main { new *
12
13     public static void main(String[] args) { new *
14
15         //firstExercise
16         //-----
17         RandomNumbers.generate();
18
19         //Создайте массив из N чисел
20         Integer[] array = createArrayOfIntegers(fileName: "inputDataOfNumbers.txt");
21
22         //На основе массива создайте список list.
23         List<Integer> list = new ArrayList<>(Arrays.asList(array));
24         System.out.println(list);
25
26         //Отсортируйте список в натуральном порядке.
27         Collections.sort(list);
28
29         //Отсортируйте список в обратном порядке.
30         list.sort(Collections.reverseOrder());
31
32         //Перемешайте список.
33         Collections.shuffle(list);
34
35         //Выполните циклический сдвиг на 1 элемент.
36         //cyclicLeftShift(list);
37         //cyclicRightShift(list);
38
39         //Оставьте в списке только уникальные элементы.
40         //System.out.println(returnUniqueList(list));
41
42         //Оставьте в списке только дублирующиеся элементы.
43         System.out.println(returnDuplicate(list));
44
45         //Из списка получите массив.
46         //replaceInArray(list);
47
48     }
49 }
```

Задание 2. Исследование HashSet.

```
1 package ru.jrp.generator;
2
3 import java.io.FileWriter;
4 import java.io.IOException;
5 import java.util.Random;
6
7 import static ru.jrp.Main.rand;
8
9 public class RandomWords { 2 usages new *
10
11     private RandomWords() {} no usages new *
12
13     public static void generate() { 1 usage new *
14         // Количество слов
15         int numberOfWords = 20;
16
17         // Длина каждого слова
18         int wordLength = 5;
19
20         // Генерация случайных слов, разделенных запятой
21         String randomWords = generateRandomWords(numberOfWords, wordLength);
22
23         // Запись строки в файл
24         try (FileWriter writer = new FileWriter("inputDataOfWords.txt")) {
25             writer.write(randomWords);
26             System.out.println("Файл успешно создан");
27         } catch (IOException e) {
28             System.out.println(e.getMessage());
29         }
30     }
```

```
1
2 @ private static String generateRandomWords(int numberOfWords, int wordLength) { 1 usage new *
3     StringBuilder sb = new StringBuilder();
4
5     for (int i = 0; i < numberOfWords; i++) {
6         // Генерация случайного слова
7         String randomWord = generateRandomWord(rand, wordLength);
8         sb.append(randomWord);
9
10        // Добавление запятой, если это последнее слово
11        if (i < numberOfWords - 1) {
12            sb.append(",\n");
13        }
14    }
15    return sb.toString();
16 }
17
18 @ private static String generateRandomWord(Random random, int wordLength) { 1 usage new *
19     String characters = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz"; // Буквы для генерации слов
20     StringBuilder sb = new StringBuilder();
21     for (int i = 0; i < wordLength; i++) {
22         // Выбираем случайный символ из строки characters
23         int index = random.nextInt(characters.length());
24         sb.append(characters.charAt(index));
25     }
26
27     return sb.toString();
28 }
29
30 }
```

```

Main.java x
11 public class Main { new *
13     public static void main(String[] args) { new *
14     ,
50         //secondExercise
51         //-----
52         RandomWords.generate();
53
54         //Создать коллекцию HashSet с типом элементов String.
55         Set<String> words = new HashSet<>();
56
57         //Добавить в неё 10 строк.
58         //read10Words("inputDataOfWords.txt", words);
59
60         //Добавить в множество минимум пять объектов, которые являются
61         //совместимыми с типом данных коллекции.
62         /*addCorrectInstances(
63             Set.of(
64                 "gelsa",
65                 new Object(),
66                 new ArrayList<>(),
67                 "asc",
68                 "sdc",
69                 new Random(),
70                 new StringBuilder(),
71                 "sdcs",
72                 "sdvs"), words);*/
73
74         //Вывести на экран элементы множества используя цикл for.
75         /*for (String word : readWords("inputDataOfWords.txt")) {
76             System.out.println(word);
77         }*/
78
79         //Добавить новый элемент, который уже присутствует в множестве.
80         /*words = readWords("inputDataOfWords.txt");
81         words.add(words.iterator().next());*/
82     }

```

```

//Определить, содержит ли коллекция определенный объект.
words = readWords( fileName: "inputDataOfWords.txt");
System.out.println(words.contains(words.iterator().next()));
System.out.println(words.contains("kjfhbvsk"));

//Удалить из коллекции любой объект.
//words = readWords("inputDataOfWords.txt");
System.out.println(words);
System.out.println(words.remove(words.iterator().next()));
System.out.println(words);

//Получить количество элементов, содержащихся в коллекции на данный момент.
/*words = readWords("inputDataOfWords.txt");
System.out.println(words.size());*/

//Удалить все элементы множества.
//words = readWords("inputDataOfWords.txt");
words.clear();

//Проверить, является ли коллекция HashSet пустой.
System.out.println(words.isEmpty());
}

```

Задание3. Добавить в файл docker-compose.yml настройки для запуска приложения.

