

Лабораторная работа №1

Цель работы:

приобрести практические навыки обработки параметров командной строки, закрепить базовые знания языка программирования Java при решении практических задач

Требования к оформлению отчета

Отчет по лабораторной работе должен содержать следующие разделы (примеры оформления отчетов можно найти в папке с заданиями):

- 1) Изложение цели работы.
- 2) Задание по лабораторной работе с описанием своего варианта.
- 3) Спецификации ввода-вывода программы.
- 4) Текст программы (кратко).
- 5) Выводы по проделанной работе.

Задание 1

Для переданной в качестве параметра последовательности из N целых чисел написать утилиту с функционалом:

- 1) Для данной последовательности, выводит значение «равны» если все элементы последовательности равны и «не равны», в противном случае.
- 2) Вывод максимального и минимального значения, а также суммы и произведения элементов последовательности.
- 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.
- 4) Вывод суммы квадратов всех отрицательных чисел.
- 5) Вывод размаха последовательности (разницы между максимальным и минимальным числом).
- 6) Вывод N чисел в случайном порядке.
- 7) Вывод медианы последовательности. Медианой упорядоченного ряда чисел с нечётным числом членов называется число, записанное посередине, а медианой упорядоченного ряда чисел с чётным числом членов называется среднее арифметическое двух чисел, записанных посередине.
- 8) Вывод всех уникальных чисел.
- 9) Распределение чисел: количество одноцифровых чисел, двуцифровых, трехцифровых и т. д.
- 10) Поиск выброса в последовательности. Выброс – это элемент последовательности максимальным образом отличающийся от других элементов последовательности. Например, в последовательности 1 2 3 4 5 6 100, выбросом является значение 100.
- 11) Вывод процента чисел, которые больше среднего значения.

- 12) Напишите утилиту `ger` для генерации числовых последовательностей, которая принимает 3 параметра: `start` – начало последовательности `end` – конец последовательности (`start < end`) `step` – шаг в последовательности.
- 13) Вывод среднего значения и выборочной дисперсии последовательности.
- 14) Вывод последовательности в порядке убывания.

Задание 2

Написать функцию, выполняющую указанную операцию над массивом. Использовать только базовые возможности языка, без привлечения специализированных функций для обработки коллекций. Ввод массивов выполнять из командной строки.

- 1) Напишите метод `double[] subarray(double[] array, int startIndex, int endIndex)`, выделяющий подмассив по указанным индексам (первый включает, второй исключает).
- 2) Напишите метод `reverse(double[] array)`, который меняет порядок элементов в массиве на обратный.
- 3) Написать метод `shiftRight(double[] array, int shift)`, который сдвигает элементы массива `array` на заданное число позиций `shift` вправо.
- 4) Напишите метод `double[][] random(double a, double b, int size)`, который создает квадратную матрицу заданного размера и заполняет ее случайными числами из указанного интервала.
- 5) Напишите метод `long[] removeElement(long[] array, long element)`, который ищет и удаляет из массива указанный элемент.
- 6) Напишите метод `double[][] transpose(double[][] matrix)`, транспонирующий переданную матрицу.
- 7) Написать метод `add(double[] array, int index, double element)`, который добавляет в массив еще один элемент по указанному индексу.
- 8) Напишите метод `randomPerturbations(double[][] matrix)`, который переставляет элементы матрицы `matrix` случайным образом.
- 9) Написать метод `shiftLeft(double[] array, int shift)`, который сдвигает элементы массива `array` на заданное число позиций `shift` влево.
- 10) Напишите метод `double[] flatten(double[][] array)`, который преобразует двумерный массив в соответствующий ему одномерный, выстраивая элементы по порядку.
- 11) Напишите метод `double[][] single(int size)`, который создает единичную матрицу заданного размера.

Задание 3

Решите задачу на обработку строк. Ввод исходных строк выполнять из командной строки.

- 1) Написать метод `String shiftRight(String str, int shift)` который возвращает строку, символы которой сдвинуты на `shift` позиций вправо.

```

shiftRight(null, *) = null
shiftRight("abcd", 2) = "cdab"
shiftRight(*, -1) - сдвиг влево на 1

```

- 2) Напишите метод **boolean polindrome(String str)** проверяющий, является ли строка палиндромом или нет. Палиндром – это такая строка, которая в прямом и обратном порядке читается одинаково. Например: А лис, он умён – крыса сыр к нему носила.
- 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.
- 4) Напишите метод **String swapStringCase(String str)**, выполняющий замену регистра для каждого символа строки. Метод должен работать следующим образом:

```

swapCase(null) = null
swapCase("") = ""
swapCase("The dog has a BONE") = "tHE DOG HAS A bone"

```

- 5) Написать функцию **String randomString(int lenght, boolean asciiOnly)** для генерации случайных строк заданного размера. Функция должна принимать флаг **asciiOnly**, определяющий, должны ли в итоговой строке быть только ASCII символы.
- 6) Напишите метод **String abbreviate(String str)**, которая для введенной строки генерирует соответствующую ей аббревиатуру.
- 7) Напишите метод **String repeat(char ch, int repeat)** который строит строку из указанного символа, повторённого заданное количество раз.

```

repeat('e', 0) = ""
repeat('e', 3) = "eee"
repeat('e', -2) = ""

```

- 8) Напишите метод **boolean isAllLowerCase(String cs)**, проверяющий, все ли символы строки в нижнем регистре или нет.

```

isAllLowerCase(null) = false
isAllLowerCase("") = false
isAllLowerCase(" ") = false
isAllLowerCase("abc") = true
isAllLowerCase("abC") = false

```

- 9) Напишите метод **String xor(String str1, String str2)**, который возвращает строку получающуюся операцией XOR битового представлением для введенных строк.
- 10) Напишите метод **String stripWhitespaces(String str)**, убирающий пробелы по концам строки. Метод должен работать следующим образом:

```

stripWhitespaces(null) = null
stripWhitespaces("") = null
stripWhitespaces(" ") = null
stripWhitespaces("abc") = "abc"
stripWhitespaces(" abc") = "abc"
stripWhitespaces("abc ") = "abc"

```

```
stripWhitespaces(" abc ") = "abc"  
stripWhitespaces(" ab c ") = "ab c"
```

- 11) Напишите метод **String capitalize(String str)**, выполняющий капитализацию строки. Капитализация – это замена первых букв слов с маленьких на большие. Метод должен работать следующим образом:

```
capitalize(null) = null  
capitalize("") = ""  
capitalize("cat") = "Cat"  
capitalize("cAt") = "CAt"
```