|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **Компьютерные системы и сети (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.04.01 Информатика и вычислительная техника**

МАГИСТЕРСКАЯ ПРОГРАММА **09.04.01/12 Интеллектуальный анализ больших**

**данных в системах поддержки принятия решений.**

**Отчет**

**по лабораторной работе № 2**

**Вариант № 2**

**Название:** Арифметические операции

**Дисциплина:** языки программирования для работы с большими данными

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-23М |  |  | А.С. Бояринов |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | П.В. Степанов |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2023

**Цель:** Освоить арифметические операции в Java.

**Задание 1:**

Ввести n строк с консоли. Упорядочить и вывести строки в порядке возрастания (убывания) значений их длины.

println("Input strings with separate - comma")

val strings: String = readln()

var newStrings :List<String> = strings.split(",")

newStrings = newStrings.sortedWith(compareBy { it.length }) // оказывается, val - не изменяемое значение, а

// вот var - изменяемое

**Задание 2:**

Ввести n строк с консоли. Вывести на консоль те строки, длина которых меньше (больше) средней, а также длину.

var totalSize: Int = 0

for (i in newStrings) {

totalSize += i.length

}

var averageSize: Float = 0F

averageSize += totalSize / newStrings.size

val average: MutableList<String> = mutableListOf() // Добавление в писок разве не изменяет переменную?

for (i in newStrings) {

if (i.length < averageSize) {

average.add(i)

}

}

println("Strings sorted by length: $newStrings, strings whose length is less average size: $average " +

"(average- $averageSize)")

**Задание 3:**

Выполнить циклический сдвиг заданной матрицы на k позиций вправо (влево, вверх, вниз).

val rows: Int = (3..5).random()

val cols: Int = (3..5).random()

// Для вывода значений массива используйте класс Arrays с методом toString(), который вернёт результат в удобном и

// читаемом виде. Сейчас в Kotlin появилась функция contentToString(), которая является предпочтительным вариантом.

// собственно, с toString все падает

val matrixA = Array(rows) { IntArray(cols) { (0..10).random() } }

println("Матрица: ")

for (row in matrixA) {

println(row.toList())

}

println("Сдвиг влево на 2: ")

for (row in matrixA) {

val list = row.toList()

Collections.rotate(list, -2) // "+" - сдвиг вправо

println(list)

}

val transposeMatrix1 = Array(rows) { IntArray(cols) { (0..10).random() } }

println("Сдвиг вверх: ")

for(i in transposeMatrix1.indices){

for(j in transposeMatrix1[i].indices){

val buff = matrixA[0][j]

if (i != transposeMatrix1.size-1){

transposeMatrix1[i][j]=matrixA[i+1][j]

}

else {

transposeMatrix1[i][j] = buff

}

}

println("${transposeMatrix1[i].contentToString()} ")

}

println("Сдвиг вниз: ")

for(i in transposeMatrix1.indices.reversed()){

matrixA[transposeMatrix1.size - 1]

if (i == transposeMatrix1.size - 1)

{

transposeMatrix1[0]=matrixA[transposeMatrix1.size - 1]

}

if (i != 0 && i != transposeMatrix1.size - 1) {

transposeMatrix1[i]=matrixA[i-1]

}

else {

transposeMatrix1[transposeMatrix1.size - 1] = matrixA[transposeMatrix1.size - 2]

}

}

for(i in transposeMatrix1.indices) {

println("${transposeMatrix1[i].contentToString()} ")

}

**Задание 4:**

Найти и вывести наибольшее число возрастающих (убывающих) элементов матрицы, идущих подряд.

var count: Int = 0

var max: Int = 0

val mutableList = mutableListOf<Int>()

for(i in matrixA.indices) {

for (j in matrixA[i].indices) {

mutableList.add(matrixA[i][j])

}

}

var temList = mutableListOf<Int>()

var result = mutableListOf<Int>()

//println(mutableList)

for (i in 0 until mutableList.size - 1){

if (mutableList[i] <= mutableList[i+1]) {

count += 1

temList.add(mutableList[i])

}

else {

count += 1

temList.add(mutableList[i])

if (count >= max) {

max = count

result = temList

temList = mutableListOf<Int>()

count = 0

} else {

temList = mutableListOf<Int>()

count = 0

}

}

}

println("Количество подряд возрастающих элементов: $max, сами элементы: $result")

**Вывод:** были освоены арифметические операции в Java.