|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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 09.04.01/07 **Интеллектуальные системы анализа, обработки и интерпретации больших данных**

**Отчет**

|  |
| --- |
| **по лабораторной работе № 2** |

**Название:**

Арифметические операции

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

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

Москва, 2022

Вариант 1:

1. Ввести n строк с консоли, найти самую короткую и самую длинную строки. Вывести найденные строки и их длину.
2. Ввести n строк с консоли. Упорядочить и вывести строки в порядке возрастания (убывания) значений их длины.

Решение:

|  |
| --- |
| /\* Малкина Т.А. Вариант 11 Вариант 1 В приведенных ниже заданиях необходимо вывести внизу фамилию разработчика, дату и время получения задания, а также дату и время сдачи задания. Для получения последней даты и времени следует использовать класс Date.  1. Ввести n строк с консоли, найти самую короткую и самую длинную строки. Вывести найденные строки и их длину.  \*/ Scanner in = new Scanner(System.in); System.out.println("------------------------------"); System.out.println("Вариант 1"); System.out.println("Введите n строк через Enter"); String str = null; ArrayList<String> list = new ArrayList<>(); while(!(str = in.nextLine()).isEmpty()) {  list.add(str); } System.out.println("Вариант 1, задание 1"); int minstr = 0; int maxstr = 0; for(int i = 0 ; i < list.size() ; i++){  if (list.get(i).length() <= list.get(minstr).length()){  minstr = i;  }  if (list.get(i).length() > list.get(maxstr).length()) {  maxstr = i;  } } System.out.println("Самая короткая строка - "+list.get(minstr)+ ". Её длина - "+list.get(minstr).length()); System.out.println("Самая длинная строка - "+list.get(maxstr)+ ". Её длина - "+list.get(maxstr).length()); /\* Вариант 1 В приведенных ниже заданиях необходимо вывести внизу фамилию разработчика, дату и время получения задания, а также дату и время сдачи задания. Для получения последней даты и времени следует использовать класс Date.  2. Ввести n строк с консоли. Упорядочить и вывести строки в порядке возрастания (убывания) значений их длины.  \*/ System.out.println("------------------------------"); System.out.println("Вариант 1, задание 2"); System.out.println("Изначальная строка - "+list); Comparator<String> stringLengthComparator = new StringLengthSort(); Collections.sort(list, stringLengthComparator); System.out.println("Сортировка по возрастанию - "+list); Collections.sort(list,Collections.reverseOrder(stringLengthComparator)); System.out.println("Сортировка по убыванию - "+list); |

Вариант 2:

1. Упорядочить строки (столбцы) матрицы в порядке возрастания значений элементов k-го столбца (строки).
2. Выполнить циклический сдвиг заданной матрицы на k позиций вправо (влево, вверх, вниз).

Решение:

|  |
| --- |
| /\* Вариант 2 Ввести с консоли n – размерность матрицы a[n][n]. Задать значения элементов матрицы в интервале значений от -n до n с помощью датчика случайных чисел.  1. Упорядочить строки (столбцы) матрицы в порядке возрастания значений элементов k-го столбца (строки).  \*/ System.out.println("------------------------------"); System.out.println("Вариант 2"); System.out.print("Введите размерность матрицы n - "); int n = in.nextInt(); int[][] matrix = new int[n][n]; int[][] matrixnum2 = new int[n][n]; int a=-n; int b=n; System.out.println("Исходная матрица: "); for(int i = 0 ; i < n ; i++){  for(int j = 0 ; j < n ; j++){  matrix[i][j] = a + (int)(Math.random() \* ((b - a) + 1));  matrixnum2[i][j] = matrix[i][j];  System.out.print(" " +matrix[i][j]+" ");  }  System.out.println(); } System.out.println("------------------------------"); System.out.println("Вариант 2, задание 1"); System.out.print("Введите номер необходимой сртировки (0 - столбцы; 1 - строки) - "); int sort = in.nextInt(); System.out.print("Введите k - "); int k= in.nextInt(); if (k > n){  System.out.print("Сотрировка невозможна"); } else {  boolean sortable = true;  while (sortable) {  sortable = false;  switch (sort){  case 0:  for (int i = 1; i < matrix.length; i++) {  if (matrix[i][k] < matrix[i - 1][k]) {  int[] tmp = matrix[i];  matrix[i] = matrix[i - 1];  matrix[i - 1] = tmp;  sortable = true;  }  }  break;  case 1:  for (int i = 0; i < matrix.length - 1; i++) {  if (matrix[k][i] > matrix[k][i + 1]) {  for (int j = 0; j < matrix.length; j++) {  sortable = true;  int tmp = matrix[j][i];  matrix[j][i] = matrix[j][i + 1];  matrix[j][i + 1] = tmp;  }  }  }  break;  }  } } System.out.println("Измененная матрица :"); for(int i = 0 ; i < n ; i++){  for(int j = 0 ; j < n ; j++){  System.out.print(" " +matrix[i][j]+" ");  }  System.out.println(); } /\* Вариант 2 Ввести с консоли n – размерность матрицы a[n][n]. Задать значения элементов матрицы в интервале значений от -n до n с помощью датчика случайных чисел.  2. Выполнить циклический сдвиг заданной матрицы на k позиций вправо (влево, вверх, вниз).  \*/ System.out.println("------------------------------"); System.out.println("Вариант 2, задание 2"); System.out.print("Введите номер необходимого сдвига (0 - лево; 1 - вправо; 2 - вверх; 3 - вниз) - "); int sdvig = in.nextInt(); System.out.print("Введите сдвиг - "); int step = in.nextInt(); int buff, z, x; if (step > matrixnum2.length || step < 1) {  System.out.println("Сдвиг невозможен"); } else {  switch (sdvig) {  case 0:  for (int r = 0; r < step; r++) {  for (z = 0; z < matrixnum2.length; z++) {  buff = matrixnum2[z][0];  for (x = 0; x < matrixnum2.length - 1; x++) {  matrixnum2[z][x] = matrixnum2[z][x + 1];  }  matrixnum2[z][x] = buff;  }  }  break;  case 1:  for (int r = 0; r < step; r++) {  for (z = 0; z < matrixnum2.length; z++) {  buff = matrixnum2[z][matrixnum2.length - 1];  for (x = matrixnum2.length - 1; x > 0; x--) {  matrixnum2[z][x] = matrixnum2[z][x - 1];  }  matrixnum2[z][x] = buff;  }  }  break;  case 2:  for (int r = 0; r < step; r++) {  for (z = 0; z < matrixnum2.length; z++) {  buff = matrixnum2[0][z];  for (x = 0; x < matrixnum2.length - 1; x++) {  matrixnum2[x][z] = matrixnum2[x + 1][z];  }  matrixnum2[x][z] = buff;  }  }  break;  case 3:  for (int r = 0; r < step; r++) {  for (z = 0; z < matrixnum2.length; z++) {  buff = matrixnum2[matrixnum2.length - 1][z];  for (x = matrixnum2.length - 1; x > 0; x--) {  matrixnum2[x][z] = matrixnum2[x - 1][z];  }  matrixnum2[x][z] = buff;  }  }  break;  } } System.out.println("Измененная матрица :"); for(int i = 0 ; i < n ; i++){  for(int j = 0 ; j < n ; j++){  System.out.print(" " +matrixnum2[i][j]+" ");  }  System.out.println(); } |

Ссылка на репозиторий:

https://github.com/Carpediem2026/BigData