**Министерство цифрового развития, связи и массовых коммуникаций**

**Российской Федерации Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский технический университет связи и информатики»**

Кафедра математической кибернетики и информационных технологий

Отчет по лабораторной работе №1

на тему «Методы сортировки»

по дисциплине «Структуры и алгоритмы обработки данных»

Выполнила: студентка группы БВТ1904

Кремер К.С.

# 

Москва 2021

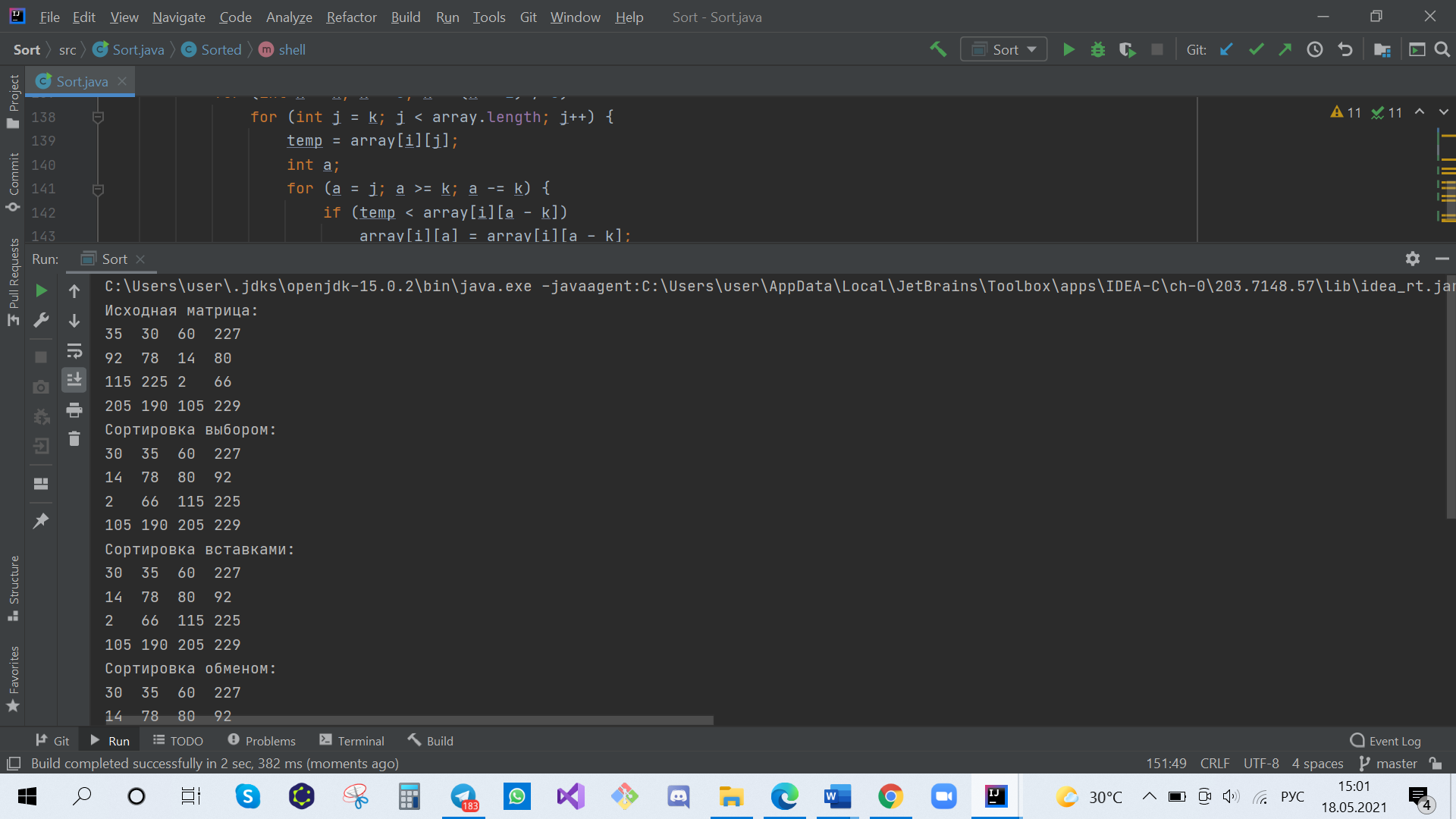
**Цель работы**

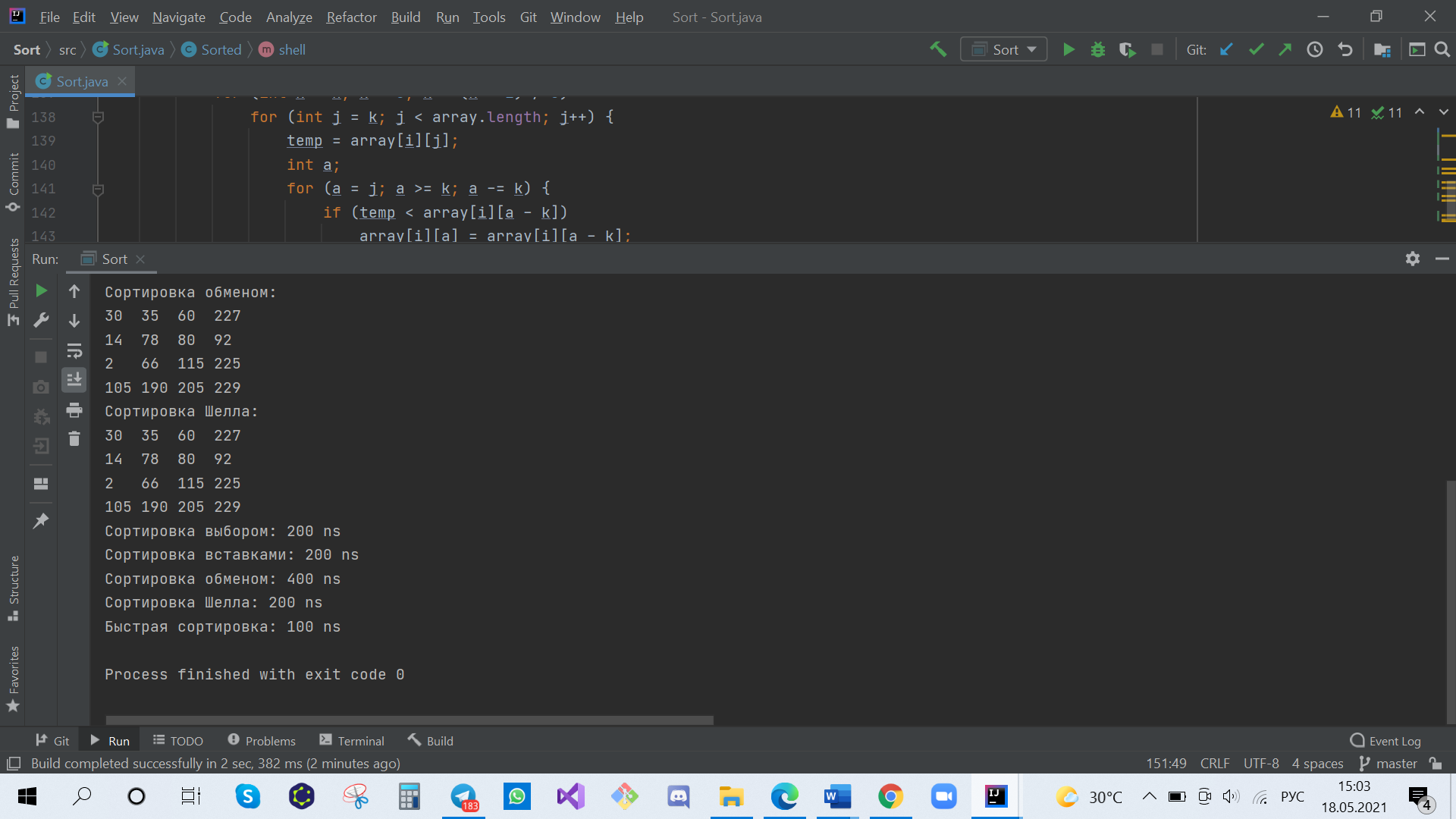
Реализовать заданный метод сортировки строк числовой матрицы в соответствии с индивидуальным заданием. Для всех вариантов добавить реализацию быстрой сортировки (quicksort). Оценить время работы каждого алгоритма сортировки и сравнить его со временем стандартной функции сортировки, используемой в выбранном языке программирования.

**Код программы**

import java.util.Random;  
  
public class Sort {  
 public static void main(String[] args) {  
 Random rand = new Random();  
 int[][] array = new int[4][4];  
 for (int i = 0; i < array.length; i++) {  
 for (int j = 0; j < array.length; j++) {  
 int upperbound = 250;  
 array[i][j] = rand.nextInt(upperbound);  
 }  
 }  
  
 System.*out*.println("Исходная матрица:");  
 for (int i = 0; i < array.length; i++) {  
 for (int j = 0; j < array.length; j++) {  
 System.*out*.print(array[i][j] + "\t");  
 }  
 System.*out*.println();  
 }  
  
 int low = 0;  
 int high = array.length - 1;  
  
 Sorted sort = new Sorted();  
 sort.choice(array);  
 sort.insert(array);  
 sort.exchange(array);  
 sort.shell(array);  
 sort.quickSort(array, low, high);  
  
 long startchoice = System.*nanoTime*();  
 long endchoice = System.*nanoTime*();  
 System.*out*.println("Сортировка выбором: " + (endchoice-startchoice) + " ns");  
  
 long startinsert = System.*nanoTime*();  
 long endinsert = System.*nanoTime*();  
 System.*out*.println("Сортировка вставками: " + (endinsert-startinsert) + " ns");  
  
 long startexchange = System.*nanoTime*();  
 long endexchange = System.*nanoTime*();  
 System.*out*.println("Сортировка обменом: " + (endexchange-startexchange) + " ns");  
  
 long startshell = System.*nanoTime*();  
 long endshell = System.*nanoTime*();  
 System.*out*.println("Сортировка Шелла: " + (endshell-startshell) + " ns");  
  
 long startquick = System.*nanoTime*();  
 long endquick = System.*nanoTime*();  
 System.*out*.println("Быстрая сортировка: " + (endquick-startquick) + " ns");  
 }  
}  
  
class Sorted {  
  
 void choice(int array[][]) {  
 //сортировка выбором  
 for (int i = 0; i < array.length; i++) {// i - номер строки  
 for (int j = 0; j < array.length; j++) {  
 int pos = j;  
 int min = array[i][j]; // цикл выбора наименьшего элемента  
 for (int k = j + 1; k < array.length; k++) {  
 if (array[i][k] < min) {  
 pos = k; // pos - индекс наименьшего элемента  
 min = array[i][k];  
 }  
 }  
 array[i][pos] = array[i][j];  
 array[i][j] = min; // меняем местами наименьший с array[i]  
 }  
 }  
 System.*out*.println("Сортировка выбором:");  
 for (int i = 0; i < array.length; i++) {  
 for (int j = 0; j < array[i].length; j++) {  
 System.*out*.print(array[i][j] + "\t");  
 }  
 System.*out*.println();  
 }  
 }  
  
 void insert(int array[][]) {  
 //сортировка вставками  
 for (int i = 0; i < array.length; i++) {  
 for (int j = 0; j < array.length - 1; j++) {  
 int min = j;  
 for (int k = j + 1; k < array.length; k++) {  
 if (array[i][k] < array[i][min]) {  
 min = k;  
 }  
 }  
 int tmp = array[i][j];  
 array[i][j] = array[i][min];  
 array[i][min] = tmp;  
 }  
 }  
 System.*out*.println("Сортировка вставками:");  
 for (int i = 0; i < array.length; i++) {  
 for (int j = 0; j < array[i].length; j++) {  
 System.*out*.print(array[i][j] + "\t");  
 }  
 System.*out*.println();  
 }  
 }  
  
 void exchange(int array[][]) {  
 //сортировка обменом  
 for (int i = 0; i < array.length; i++) {  
 for (int j = array.length - 1; j >= 1; j--) { //Внешний цикл  
 for (int k = 0; k < j; k++) { //Внутренний цикл  
 if (array[i][k] > array[i][k + 1]) {  
 int min = array[i][k]; //во временную переменную помещаем первый элемент  
 array[i][k] = array[i][k + 1]; //на место первого ставим второй элемент  
 array[i][k + 1] = min;  
 }  
 }  
 }  
 }  
 System.*out*.println("Сортировка обменом:");  
 for (int i = 0; i < array.length; i++) {  
 for (int j = 0; j < array[i].length; j++) {  
 System.*out*.print(array[i][j] + "\t");  
 }  
 System.*out*.println();  
 }  
 }  
  
 void shell(int array[][]) {  
 //сортировка Шелла  
 int temp;  
 int h = 0;//величина интервала  
  
 //вычисляем исходное значение интервала  
 while (h <= array.length / 3)  
 h = 3 \* h + 1;  
  
 for (int i = 0; i < array.length; i++) {  
 for (int k = h; k > 0; k = (k - 1) / 3)  
 for (int j = k; j < array.length; j++) {  
 temp = array[i][j];  
 int a;  
 for (a = j; a >= k; a -= k) {  
 if (temp < array[i][a - k])  
 array[i][a] = array[i][a - k];  
 else  
 break;  
 }  
 array[i][a] = temp;  
 }  
 }  
  
 System.*out*.println("Сортировка Шелла:");  
 for (int i = 0; i < array.length; i++) {  
 for (int j = 0; j < array[i].length; j++) {  
 System.*out*.print(array[i][j] + "\t");  
 }  
 System.*out*.println();  
 }  
 }  
  
 void quickSort(int array[][], int low, int high) {  
 //быстрая сортировка  
 for (int l = 0; l < array.length; l++) {  
 if (array.length == 0)  
 return;//завершить выполнение, если длина массива равна 0  
 if (low >= high)  
 return;//завершить выполнение если уже нечего делить  
 int middle = low + (high - low) / 2;  
 int opora = array[l][middle];  
 int i = low, j = high;  
 while (i <= j) {  
 while (array[l][i] < opora) {  
 i++;  
 }  
  
 while (array[l][j] > opora) {  
 j--;  
 }  
 if (i <= j) {//меняем местами  
 int temp = array[l][i];  
 array[l][i] = array[l][j];  
 array[l][j] = temp;  
 i++;  
 j--;  
 }  
 }  
 // вызов рекурсии для сортировки левой и правой части  
 if (low < j)  
 quickSort(array, low, j);  
  
 if (high > i)  
 quickSort(array, i, high);  
 }  
 }  
}

**Результат работы программы**





**Вывод**

В ходе данной лабораторной работы я реализовала заданные методы сортировки строк числовой матрицы, оценила время работы каждого алгоритма и сравнила его со временем стандартной функции сортировки, используемой в выбранном языке программирования. В результате проделанной работы я сделала вывод, что метод быстрой сортировки занимает наименее количество времени.