**Лабораторная работа 10**

**Сортировка массива**

*Сортировка, как и поиск, является распространенной задачей программирования. Для сортировки разработано множество различных алгоритмов. В этом подразделе представлен интуитивно понятный алгоритм сортировки методом выбора.*

Предположим, что требуется упорядочить массив по возрастанию. При *сортировке методом выбора* в массиве находится наименьшее значение и переставляется с первым элементом. Затем в оставшемся массиве находится наименьшее значение и переставляется со вторым элементом и так далее, пока в массиве не останется только одно значение. На следующем рисунке показано, как упорядочить массив **{2, 9, 5, 4, 8, 1, 6}** с помощью сортировки методом выбора.

|  |  |  |
| --- | --- | --- |
| Выберите 1 (наименьший элемент) и переставьте его с 2 (первый элемент) в массиве. | https://cs.sberuniversity.online/image/1000/auto/upsize/51e60654-6791-11eb-8f88-0242ac180002 |  |
| 1 теперь находится в правильной позиции и, таким образом, больше не нуждается в рассмотрении. | https://cs.sberuniversity.online/image/1000/auto/upsize/88b918a6-6791-11eb-ade3-0242ac180005 | Выберите 2 (наименьший элемент) и переставьте его с 9 (первый элемент) в оставшемся массиве. |
| 2 теперь находится в правильной позиции и, таким образом, больше не нуждается в рассмотрении. | https://cs.sberuniversity.online/image/1000/auto/upsize/9164e192-6791-11eb-87e3-0242ac1b0004 | Выберите 4 (наименьший элемент) и переставьте его с 5 (первый элемент) в оставшемся массиве. |
| 4 теперь находится в правильной позиции и, таким образом, больше не нуждается в рассмотрении. | https://cs.sberuniversity.online/image/1000/auto/upsize/99317f52-6791-11eb-8851-0242ac1b0004 | 5 - наименьший элемент в правильной позиции. Перестановка не требуется. |
| 5 теперь находится в правильной позиции и, таким образом, больше не нуждается в рассмотрении. | https://cs.sberuniversity.online/image/1000/auto/upsize/23b463b0-6792-11eb-b914-0242ac1b0004 | Выберите 6 (наименьший элемент) и переставьте его с 8 (первый элемент) в оставшемся массиве. |
| 6 теперь находится в правильной позиции и, таким образом, больше не нуждается в рассмотрении. | https://cs.sberuniversity.online/image/1000/auto/upsize/2beeb8be-6792-11eb-be3f-0242ac180005 | Выберите 8 (наименьший элемент) и переставьте его с 9 (первый элемент) в оставшемся массиве. |
| 8 теперь находится в правильной позиции и, таким образом, больше не нуждается в рассмотрении. | https://cs.sberuniversity.online/image/1000/auto/upsize/341b2b62-6792-11eb-b79d-0242ac180005 | Поскольку в массиве остался только один элемент, сортировка завершена. |

 Вы узнали, как работает сортировка методом выбора. Теперь задача состоит в том, чтобы реализовать ее на Java. Начните с написания кода для первой итерации, чтобы найти в массиве наименьший элемент и переставить его с первым элементом, затем проверьте, что будет отличаться для второй итерации, третьей и так далее. Это позволит написать цикл, обобщающий все итерации.

Решение может быть описано следующим образом:

for (int i = 0; i < list.length - 1; i++) {

выбрать наименьший элемент в list[i..list.length-1];

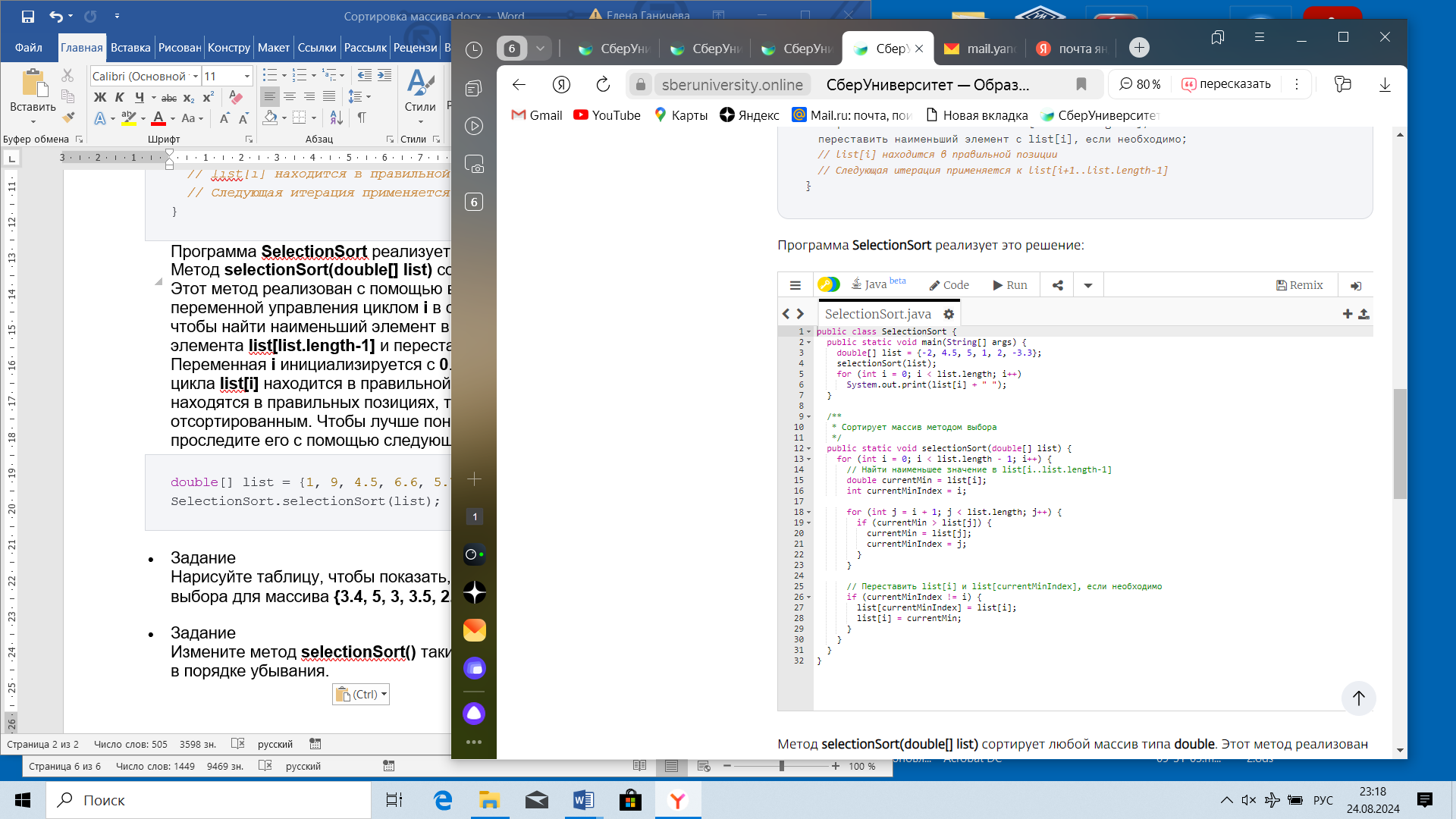
переставить наименьший элемент с list[i], если необходимо;

*// list[i] находится в правильной позиции*

*// Следующая итерация применяется к list[i+1..list.length-1]*

}

Программа **SelectionSort** реализует это решение:



Метод **selectionSort(double[] list)** сортирует любой массив типа **double**. Этот метод реализован с помощью вложенного цикла **for**. Внешний цикл (с переменной управления циклом**i** в строчке №13) повторяется для того, чтобы найти наименьший элемент в массиве от элемента **list[i]** до элемента**list[list.length-1]** и переставить его с элементом **list[i]**.

Переменная **i** инициализируется с **0**. После каждой итерации внешнего цикла **list[i]** находится в правильной позиции. В конце концов, все элементы находятся в правильных позициях, таким образом, весь массив становится отсортированным. Чтобы лучше понять, как работает этот метод, проследите его с помощью следующих предложений:

double[] list = {1, 9, 4.5, 6.6, 5.7, -4.5};

SelectionSort.selectionSort(list);

**Задание 1.**

Измените метод **selectionSort()** таким образом, чтобы отсортировать числа в порядке убывания.

**Ваш вариант решения задания 1:**

**public class SelectionSort {**

**public static void main(String[] args) {**

**double[] list = {-2, 4.5, 5, 1, 2, -3.3};**

**selectionSort(list);**

**for (int i = 0; i < list.length; i++) {**

**System.out.print(list[i] + " ");**

**}**

**}**

**public static void selectionSort(double[] list) {**

**for (int i = 0; i < list.length - 1; i++) {**

**double currentMax = list[i];**

**int currentMaxIndex = i;**

**for (int j = i + 1; j < list.length; j++) {**

**if (currentMax < list[j]) {**

**currentMax = list[j];**

**currentMaxIndex = j;**

**}**

**}**

**if (currentMaxIndex != i) {**

**list[currentMaxIndex] = list[i];**

**list[i] = currentMax;**

**}**

**}**

**}**

**}**

Задания 2 – 5 по указанию преподавателя.

**Задание 2.**

**Напишите программу сортировки массива методом простого обмена для массива из 15 элементов.**

**Ваш вариант решения задания 2:**

**import java.util.Scanner;**

**public class Main {**

**public static void main(String[] args) {**

**Scanner scanner = new Scanner(System.in);**

**int[] numbers = new int[15];**

**System.out.println("Введите 15 целых чисел для массива:");**

**for (int i = 0; i < numbers.length; i++) {**

**numbers[i] = scanner.nextInt();**

**}**

**int n = numbers.length;**

**boolean swapped;**

**for (int i = 0; i < n - 1; i++) {**

**swapped = false;**

**for (int j = 0; j < n - i - 1; j++) {**

**if (numbers[j] > numbers[j + 1]) {**

**int temp = numbers[j];**

**numbers[j] = numbers[j + 1];**

**numbers[j + 1] = temp;**

**swapped = true;**

**}**

**}**

**if (!swapped) {**

**break;**

**}**

**}**

**System.out.println("итоговый массив:");**

**for (int value : numbers) {**

**System.out.print(value + " ");**

**}**

**}**

**}**

**Задание 3.**

**Напишите программу сортировки массива простыми вставками для массива из 15 элементов..**

**Ваш вариант решения задания 3:**

**import java.util.Scanner;**

**public class Main {**

**public static void main(String[] args) {**

**Scanner scanner = new Scanner(System.in);**

**int[] numbers = new int[15];**

**System.out.println("Введите 15 целых чисел для массива:");**

**for (int i = 0; i < numbers.length; i++) {**

**numbers[i] = scanner.nextInt();**

**}**

**for (int i = 1; i < numbers.length; i++) {**

**int k = numbers[i];**

**int j = i - 1;**

**while (j >= 0 && numbers[j] > k) {**

**numbers[j + 1] = numbers[j];**

**j--;**

**}**

**numbers[j + 1] = k;**

**}**

**System.out.println("итоговый массив:");**

**for (int value : numbers) {**

**System.out.print(value + " ");**

**}**

**}**

**}**

**Задание 4.**

**Напишите программу сортировки массива методом быстрой сортировки для массива из 15 элементов..**

**Ваш вариант решения задания 4:**

**Задание 5.**

**Напишите программу сортировки массива методом Шелла для массива из 15 элементов..**

**Ваш вариант решения задания 5:**