**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ**

**УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3

по дисциплине: «Введение в разработку программного обеспечения»

на тему: **«**Моделирование и алгоритмизация как средства проектирования программного обеспечения»

Выполнил: студент гр. ИТП-21

Сеноженский В. В.

Принял: преподаватель

Башаримов Ю. С.

Гомель 2024

**Цель работы**: Разработать алгоритм сортировка пузырьком и разработка модульных тестов для проверки работоспособности алгоритма.

На рисунке 1 приведён фрагмент таблицы с индивидуальным заданием и

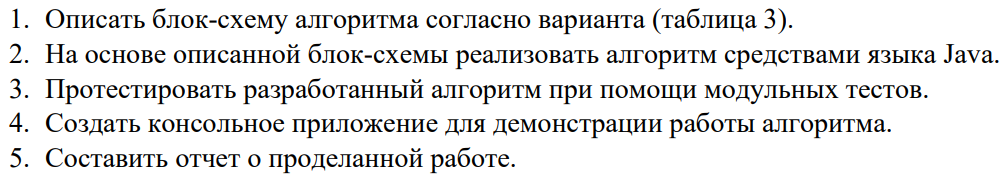




Рисунок 1 ­– Фрагмент таблицы с индивидуальными заданиями

Ход работы (Вариант 7):

Задание

Условия задания, приведённом выше. Выполнение представлено на рисунке 1.1.

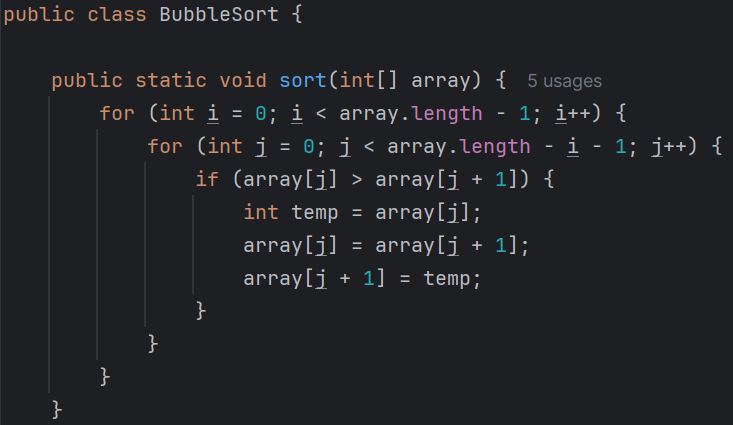


Рисунок 1.1 ­– Выполнение задания варианта

Условия задания 2- 5 на рисунке 1, приведённом выше. Код программы представлен в Приложениях А-Б.

В ходе тестирования программы можно прийти к выводу, что алгоритм работает верно. Результат выполнения программы представлен на рисунке 1.2

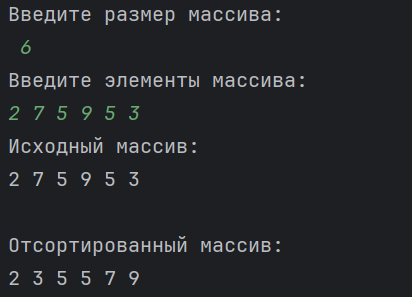


Рисунок 1.2 – Результат выполнения программы

Результат выполнения тестов для проверки сортировки методом пузырька.

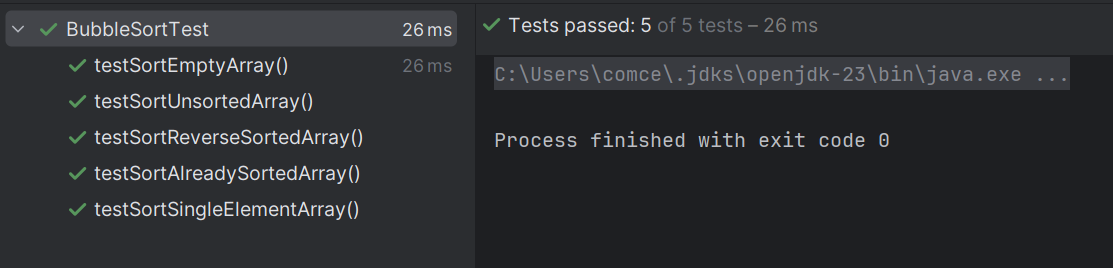


Рисунок 1.3 – выполнение тестов.

Блок-схема алгоритма, сортировка методом пузырька представлена на рисунке 1.4.

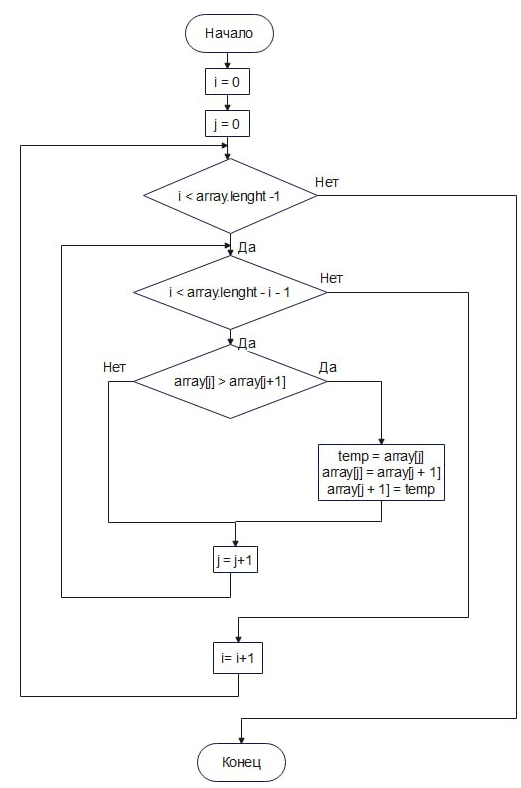


Рисунок 1.4 – схема сортировки.

Вывод: Освоил написание алгоритма сортировки пузырьком, выполнил необходимые тесты для проверки алгоритма. Разработал консольное окно.

Приложение А

(Консольное приложение)

import java.util.Scanner;  
  
public class Main {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.in);  
  
 System.out.println("Введите размер массива:");  
 int size = scanner.nextInt();  
  
 int[] array = new int[size];  
 System.out.println("Введите элементы массива:");  
 for (int i = 0; i < size; i++) {  
 array[i] = scanner.nextInt();  
 }  
  
 System.out.println("Исходный массив:");  
 printArray(array);  
  
 BubbleSort.sort(array);  
  
 System.out.println("\nОтсортированный массив:");  
 printArray(array);  
 }  
  
 private static void printArray(int[] array) {  
 for (int item : array) {  
 System.out.print(item + " ");  
 }  
 System.out.println();  
 }  
}  
  
class BubbleSort {  
 public static void sort(int[] array) {  
 for (int i = 0; i < array.length - 1; i++) {  
 for (int j = 0; j < array.length - i - 1; j++) {  
 if (array[j] > array[j + 1]) {  
 int temp = array[j];  
 array[j] = array[j + 1];  
 array[j + 1] = temp;  
 }  
 }  
 }  
 }  
}

Приложение Б

(алгоритм с тестами)

import static org.junit.jupiter.api.Assertions.*assertArrayEquals*;  
import org.junit.jupiter.api.Test;  
import org.example.BubbleSort;  
  
public class BubbleSortTest {  
  
 @Test  
 public void testSortEmptyArray() {  
 int[] array = {};  
 BubbleSort.*sort*(array);  
 int[] expected = {};  
 *assertArrayEquals*(expected, array);  
 }  
  
 @Test  
 public void testSortSingleElementArray() {  
 int[] array = {1};  
 BubbleSort.*sort*(array);  
 int[] expected = {1};  
 *assertArrayEquals*(expected, array);  
 }  
  
 @Test  
 public void testSortAlreadySortedArray() {  
 int[] array = {1, 2, 3, 4, 5};  
 BubbleSort.*sort*(array);  
 int[] expected = {1, 2, 3, 4, 5};  
 *assertArrayEquals*(expected, array);  
 }  
  
 @Test  
 public void testSortReverseSortedArray() {  
 int[] array = {5, 4, 3, 2, 1};  
 BubbleSort.*sort*(array);  
 int[] expected = {1, 2, 3, 4, 5};  
 *assertArrayEquals*(expected, array);  
 }  
  
 @Test  
 public void testSortUnsortedArray() {  
 int[] array = {3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5};  
 BubbleSort.*sort*(array);  
 int[] expected = {1, 1, 2, 3, 3, 4, 5, 5, 5, 6, 9};  
 *assertArrayEquals*(expected, array);  
 }  
}