**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: «Технологии разработки программного обеспечения»

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

Выполнил: студент гр. ИТИ-21

Тимошенко И.Ю.

Принял: преподаватель

Стефановский И. Л.

Гомель 2024

**Цель работы:** изучить моделирование и алгоритмизацию как средства проектирования программного обеспечения

**Задание:**

1. Описать блок-схему алгоритма согласно варианта (поразрядная сортировка)
2. На основе описанной блок-схемы реализовать алгоритм средствами языка Java.
3. Протестировать разработанный алгоритм при помощи модульных тестов.
4. Создать консольное приложение для демонстрации работы алгоритма.
5. Составить отчет о проделанной работе.

**Ход работы:**

Разработаем блок-схемы алгоритмов, на основе которых напишем код для класса сортировки. Они представлены на рисунках 1 – 4.

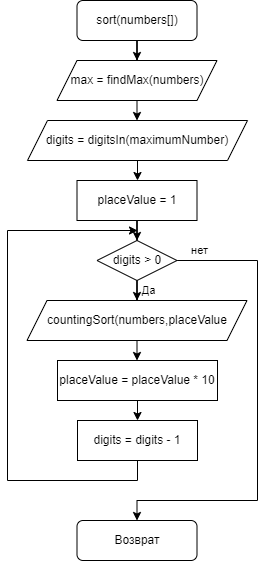


Рисунок 1 – Блок-схема алгоритма *sort*

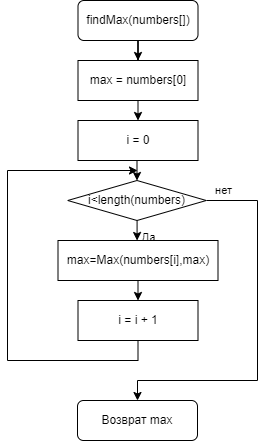


Рисунок 2 – Блок-схема алгоритма *findMax*

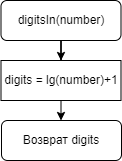


Рисунок 3 – Блок-схема алгоритма *digitsIn*

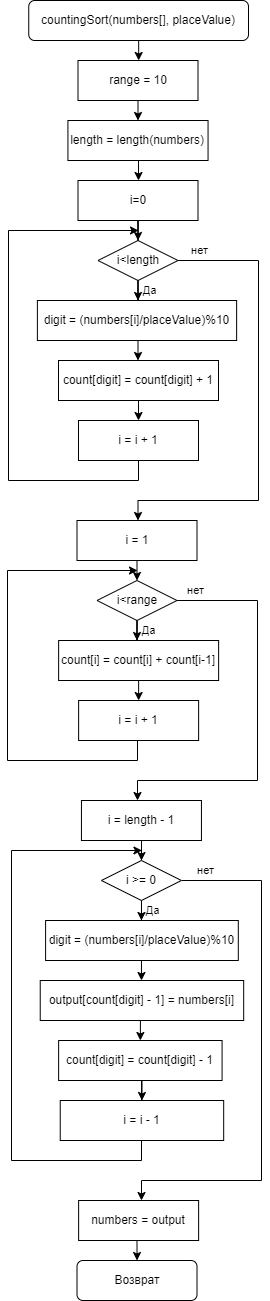
**

Рисунок 4 – Блок-схема алгоритма *countingSort*

Реализуем эти алгоритмы на языке *Java*. Представим их как класс *RadiusSort.* Далее реализуем тесты с помощью библиотеки *JUnit.* Результат выполнения тестов представлен на рисунке 5.

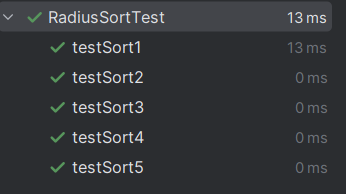


Рисунок 5 – Результаты выполнения тестов

Создадим консольное приложение и отсортируем свой массив с помощью поразрядной сортировки. Результат представлен на рисунке 6.



Рисунок 6 – Результат сортировки

**Выводы:** в результате выполнения данной лабораторной работы были изучены моделирование и алгоритмизация как средства проектирования программного обеспечения, а также библиотека *Junit* для создания модульных тестов.

**ПРИЛОЖЕНИЕ А**

(обязательное)

**RadiusSort.java\**

import java.util.\*;

import java.lang.\*;

public class RadiusSort {

public static void sort(int[] numbers) {

int maximumNumber = findMaximumNumberIn(numbers);

int numberOfDigits = calculateNumberOfDigitsIn(maximumNumber);

int placeValue = 1;

while (numberOfDigits-- > 0) {

applyCountingSortOn(numbers, placeValue);

placeValue \*= 10;

}

}

private static int findMaximumNumberIn(int[] numbers) {

int max = Integer.MIN\_VALUE;

for (int number : numbers) {

max = Math.max(max, number);

}

return max;

}

private static int calculateNumberOfDigitsIn(int number) {

return (int) Math.log10(number) + 1;

}

private static void applyCountingSortOn(int[] numbers, int placeValue) {

int range = 10;

int length = numbers.length;

int[] output = new int[length];

int[] count = new int[range];

for (int number : numbers) {

int digit = (number / placeValue) % range;

count[digit]++;

}

for (int i = 1; i < range; i++) {

count[i] += count[i - 1];

}

for (int i = length - 1; i >= 0; i--) {

int digit = (numbers[i] / placeValue) % range;

output[count[digit] - 1] = numbers[i];

count[digit]--;

}

System.arraycopy((Object)output, 0, (Object)numbers, 0, length);

}

}

**RadiusSortTest.java**

import org.junit.Test;

import static org.junit.Assert.assertArrayEquals;

public class RadiusSortTest {

@Test

public void testSort1() {

int[] actual = {68, 123, 37, 769, 134, 7, 221, 468, 387};

int[] expected = {7, 37, 68, 123, 134, 221, 387, 468, 769};

RadiusSort.sort(actual);

assertArrayEquals(expected, actual);

}

@Test

public void testSort2() {

int[] actual = {5, 8, 8, 1, 10, 4, 9, 7, 8, 3, 2, 6};

int[] expected = {1, 2, 3, 4, 5, 6, 7, 8, 8, 8, 9, 10};

RadiusSort.sort(actual);

assertArrayEquals(expected, actual);

}

@Test

public void testSort3() {

int[] actual = {0, 0, 0, 0, 0, 0};

int[] expected = {0, 0, 0, 0, 0, 0};

RadiusSort.sort(actual);

assertArrayEquals(expected, actual);

}

@Test

public void testSort4() {

int[] actual = {};

int[] expected = {};

RadiusSort.sort(actual);

assertArrayEquals(expected, actual);

}

@Test

public void testSort5() {

int[] actual = {5};

int[] expected = {5};

RadiusSort.sort(actual);

assertArrayEquals(expected, actual);

}

}

**Main.java**

import java.util.\*;

import java.lang.\*;

public class Main {

public static void main(String[] args) {

int[] array = {18, 17, 89, 5, 73,128, 28, 64, 11, 36, 50};

System.out.println(Arrays.toString(array));

RadiusSort.sort(array);

System.out.println(Arrays.toString(array));

}

}