**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

Лавшук А. Д.

Принял: преподаватель

Башаримов Ю. С.

Гомель 2024

Задания:

1. Описать блок-схему алгоритма согласно варианта (таблица 3).

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.

На рисунке 1 приведён фрагмент таблицы 3 с алгоритмом.



Рисунок 1 ­– Фрагмент таблицы 3

Ход работы(Вариант 12):

Алгоритм бинарного поиска представляет из себя следующую последовательность действий: разделение отсортированного массива на две половины, сравнение с искомым среднего элемента и в зависимости от этого выбор правой (верхней) половины или левой (нижней) для дальнейшего поиска. Таким же образом половины делятся на половины, пока искомый элемент не станет равен среднему.

Блок-схема алгоритма представлена на рисунке 2.

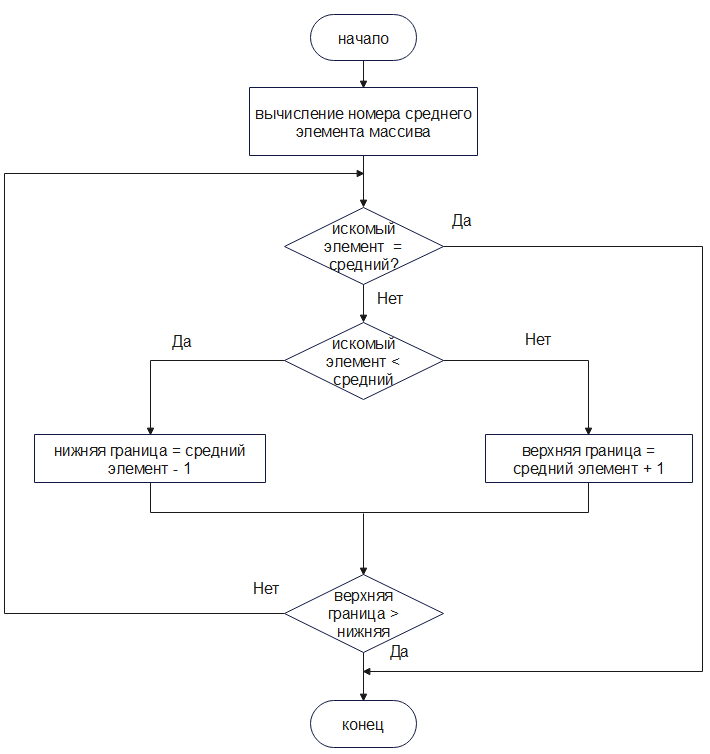


Рисунок 2 – Блок-схема алгоритма

Результат выполнения теста представлен на рисунке 3.

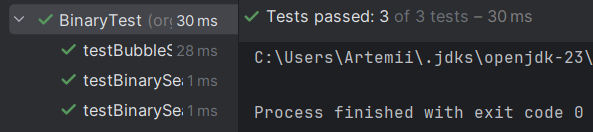


Рисунок 3 ­– Результат тестирования

Работа программы, демонстрирующей функционал класса представлена на рисунке 4.

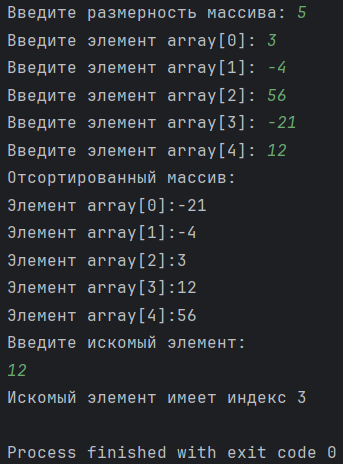


Рисунок 4 – Демонстрация работы программы

Листинг файла с описанием класса приведён в приложении А. Файл с тестом – в приложении Б. Листинг программы, демонстрирующей работу класса – в приложении В.

Вывод: В ходе лабораторной работы были дополнены знания синтаксиса языка Java, улучшены навыки работы с методами классов, а также было проведено ознакомление с библиотекой junit.

**Приложение А**

package org.example;

public class Binary {

public static int binarySearch(int[] sortedArray, int valueToFind, int low, int high) {

while (low <= high) {

int mid = low + (high - low) / 2;

if (sortedArray[mid] < valueToFind) {

low = mid + 1;

} else if (sortedArray[mid] > valueToFind) {

high = mid - 1;

} else {

return mid;

}

}

return -1;

}

}

Приложение Б

package org.example;

import org.junit.jupiter.api.Test;

import static org.junit.jupiter.api.Assertions.assertEquals;

class BinaryTest {

@Test

void testBinarySearchFound() {

int[] sortedArray = {1, 2, 5, 5, 6, 9};

int valueToFind = 5;

Binary binarySearchTest = new Binary();

int result = binarySearchTest.binarySearch(sortedArray, valueToFind, 0, sortedArray.length - 1); // Метод должен быть доступен

assertEquals(2, result);

}

@Test

void testBinarySearchNotFound() {

int[] sortedArray = {1, 2, 5, 5, 6, 9};

int valueToFind = 3;

Binary binarySearchTest = new Binary();

int result = binarySearchTest.binarySearch(sortedArray, valueToFind, 0, sortedArray.length - 1); // Метод должен быть доступен

assertEquals(-1, result);

}

}

Приложение В

import java.util.Scanner;

import static org.example.Binary.binarySearch;

import static org.example.Binary.bubbleSort;

public static void main(String[] args) {

Scanner in = new Scanner(System.in);

System.out.print("Введите размерность массива: ");

int size = in.nextInt();

int[] array = new int[size];

for (int i = 0; i < size; i++) {

System.out.print("Введите элемент array[" + i + "]: ");

array[i] = in.nextInt();

}

bubbleSort(array, size);

System.out.println("Отсортированный массив:");

for (int i = 0; i < size; i++) {

System.out.println("Элемент array[" + i + "]:" + array[i] );

}

System.out.println("Введите искомый элемент:");

int valueToFind = in.nextInt();

int resultIndex = binarySearch(array, valueToFind, 0, size - 1);

if (resultIndex == -1) {

System.out.println("Элемент не найден");

} else {

System.out.println("Искомый элемент имеет индекс " + resultIndex);

}

}