Министерство науки и высшего образования Российской Федерации

ФГБОУ ВО «Кубанский государственный технологический университет»

(ФГБОУ ВО «КубГТУ»)

Факультет информационных технологий и кибербезопасности

Кафедра информационных систем и программирования

**ОТЧЕТ**

по лабораторной работе №10.

Тема работы: «Упрощение создания тестов при помощи библиотеки Fluent Assertions».

Наименование дисциплины: «Тестирование и отладка программного обеспечения».

Подготовил:

Студент группы 23-КБ-ПР1

Юданов М.Д.

Краснодар

2025

**1 Цель работы**

Цель работы – изучить подход к созданию тестов с помощью библиотеки AssertJ.

**2 Краткое теоретическое описание**

Библиотека AsserJ предназначена для упрощения написания и понимания модульных тестов. Она предоставляет удобный «текучий» синтаксис и понятные сообщения об ошибках, которые помогают быстро определить причину сбоя теста. AsserJ не заменяет JUnit, а дополняет, улучшая читаемость и информативность проверок. Основой её работы является паттерн Fluent Interface и методы расширений, позволяющие создавать цепочки проверок в естественной форме.

**3 Задание**

1) На основе класса из предыдущей лабораторной работы (вариант задания оставить прежним) создать тестирующий проект, использующий библиотеку AssertJ.

2) Протестировать проект.

3) Оформить отчёт.

1. **Ход работы**

1-3) На основе класса из предыдущей лабораторной работы создал тестирующий класс, используя библиотеку AssertJ, протестировал проект и оформил отчёт.

import org.assertj.core.api.Assertions;

import org.junit.jupiter.api.BeforeAll;

import org.junit.jupiter.api.Test;

import java.io.IOException;

import java.io.StringReader;

import java.io.StringWriter;

import java.util.ArrayList;

import java.util.List;

public class AssertJGraphTest {

String htmlLineConnected =

"""

<html><head></head><body>\r

<h2>1 2 true</h2>\r

<h2>2 4 true</h2>\r

<h2>4 3 true</h2>\r

<h2>3 1 true</h2>\r

<body></html>\r

""";

String htmlLineIsNotConnected =

"""

<html><head></head><body>\r

<h2>1 2 true</h2>\r

<h2>2 4 true</h2>\r

<h2>4 3 true</h2>\r

<h2>1 3 true</h2>\r

<body></html>\r

""";

String txtLineMatrixConnected =

"""

-1 0 0 1\r

1 -1 0 0\r

0 0 1 -1\r

0 1 -1 0\r

""";

String htmlLineConnectedMatrixConnected =

"""

<html><head></head><body>\r

<h2>-1 0 0 1</h2>\r

<h2>1 -1 0 0</h2>\r

<h2>0 0 1 -1</h2>\r

<h2>0 1 -1 0</h2>\r

</body></html>\r

""";

String txtMatrixConnected =

"""

1 2 true

2 4 true

4 3 true

3 1 true

""";

int[][] arrayFromFileConnected = {

{-1,0,0,1},

{1,-1,0,0},

{0,0,1,-1},

{0,1,-1,0}

};

static List<Edge> listFromFileConnected = new ArrayList<>();

@BeforeAll

static void setListFromFileConnected() {

listFromFileConnected.add(new Edge(1, 2, true));

listFromFileConnected.add(new Edge(2, 4, true));

listFromFileConnected.add(new Edge(4, 3, true));

listFromFileConnected.add(new Edge(3, 1, true));

}

@Test

void readGraph() throws IOException {

try(StringReader reader = new StringReader(htmlLineConnected)) {

Graph graph = new Graph(new GraphReaderWriterHtml(), new GraphReaderWriterTxt());

graph.readFromFile(reader);

Assertions.assertThat(graph.incidenceMatrix)

.as("Считанная матрица равна null!")

.isNotNull()

.as("Считанная матрица пуста!")

.isNotEmpty()

.as("Считанная матрица не равна ожидаемому массиву!")

.isDeepEqualTo(arrayFromFileConnected);

}

}

@Test

void writeGraph() throws IOException {

Graph graph = new Graph(new GraphReaderWriterHtml(), new GraphReaderWriterTxt());

try(StringWriter writer = new StringWriter(); StringReader reader = new StringReader(htmlLineConnected)) {

graph.readFromFile(reader);

graph.writeToFile(writer);

Assertions.assertThat(writer.toString())

.as("Записанная строка пуста!")

.isNotEmpty()

.as("Записанная строка не равна ожидаемой матрице!")

.isEqualTo(txtLineMatrixConnected);

}

}

@Test

void graphReaderHtml() throws IOException {

try(StringReader reader = new StringReader(htmlLineConnected)) {

Assertions.assertThat(new GraphReaderWriterHtml().loadEdgeList(reader))

.as("Считанный список равен null!")

.isNotNull()

.as("Считанный список пуст!")

.isNotEmpty()

.as("Размер считанного списка не равен ожидаемому!")

.hasSize(4)

.as("Считанный список не равен ожидаемому списку!")

.isEqualTo(listFromFileConnected);

}

}

@Test

void writeGraphHtml() throws IOException {

try(StringWriter writer = new StringWriter()) {

new GraphReaderWriterHtml().saveIncidenceMatrix(writer, arrayFromFileConnected);

Assertions.assertThat(writer.toString())

.as("Записанная матрица пуста!")

.isNotEmpty()

.as("Размер записанной матрицы не совпадает с размером ожидаемой матрицы!")

.hasSize(htmlLineConnectedMatrixConnected.length())

.as("Записанная матрица не совпадает с ожидаемой!")

.isEqualTo(htmlLineConnectedMatrixConnected);

}

}

@Test

void graphReaderTxt() throws IOException {

try(StringReader reader = new StringReader(txtMatrixConnected)) {

Assertions.assertThat(new GraphReaderWriterTxt().loadEdgeList(reader))

.as("Считанный список равен null!")

.isNotNull()

.as("Считанный список пуст!")

.isNotEmpty()

.as("Размер считанного списка не равен ожидаемому!")

.hasSize(4)

.as("Считанный список не равен ожидаемому списку!")

.isEqualTo(listFromFileConnected);

}

}

@Test

void writeGraphTxt() throws IOException {

try(StringWriter writer = new StringWriter()) {

new GraphReaderWriterTxt().saveIncidenceMatrix(writer, arrayFromFileConnected);

Assertions.assertThat(writer.toString())

.as("Записанная матрица пуста!")

.isNotEmpty()

.as("Размер записанной матрицы не совпадает с размером ожидаемой матрицы!")

.hasSize(txtLineMatrixConnected.length())

.as("Записанная матрица не совпадает с ожидаемой!")

.isEqualTo(txtLineMatrixConnected);

}

}

@Test

void isConnected() throws IOException {

try(StringReader reader = new StringReader(htmlLineConnected)) {

Graph graph = new Graph(new GraphReaderWriterHtml(), new GraphReaderWriterTxt());

graph.readFromFile(reader);

Assertions.assertThat(graph.isConnected())

.as("Граф не связный, ожидалось обратное!")

.isTrue();

}

}

@Test

void isNotConnected() throws IOException {

try(StringReader reader = new StringReader(htmlLineIsNotConnected)) {

Graph graph = new Graph(new GraphReaderWriterHtml(), new GraphReaderWriterTxt());

graph.readFromFile(reader);

Assertions.assertThat(graph.isConnected())

.as("Граф связный, ожидалось обратное!")

.isFalse();

}

}

}

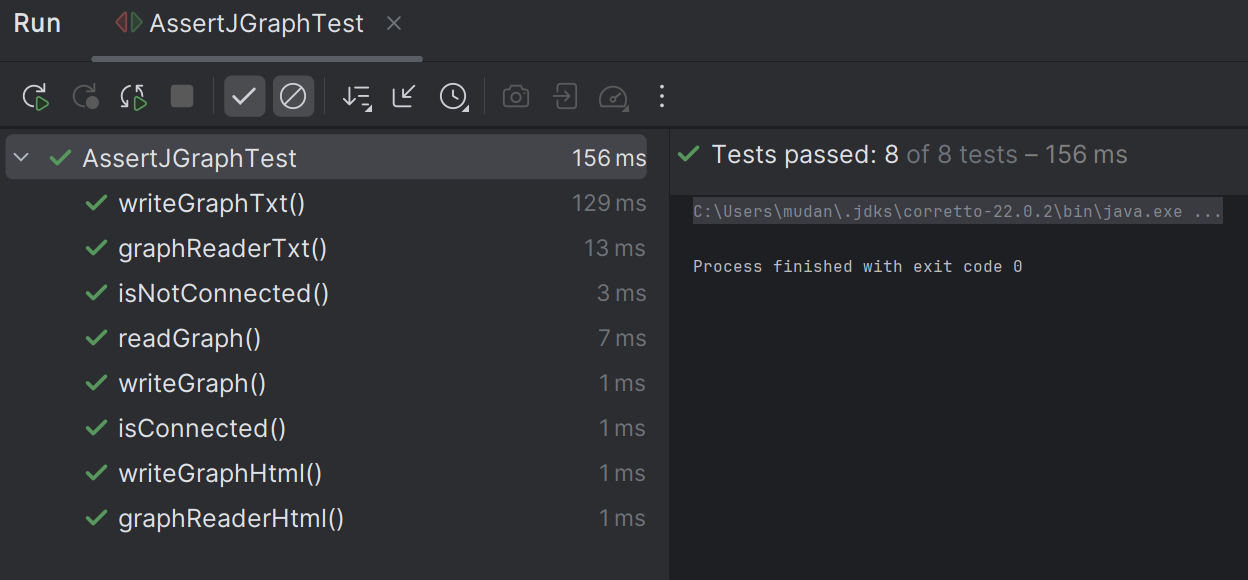


Рисунок 1 – Результат тестирования

**5 Вывод**

В ходе выполнения данной лабораторной работы удалось изучить подход к созданию тестов с помощью библиотеки AssertJ.