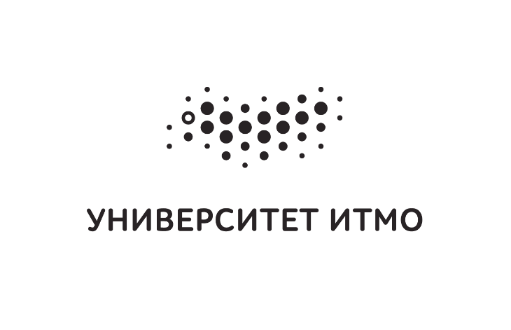
Санкт-Петербургский Национальный Исследовательский Университет Информационных Технологий, Механики и Оптики

Оптики, кафедра Информатики и прикладной математики



Лабораторная работа №1

по дисциплине

Тестирование программного обеспечения

Вариант 2

Выполнила: Жетесова Дана Айбековна

Принимающий: Исаев Илья Владимирович

Санкт-Петербург

2020

Оглавление

[Цель 2](#_Toc35431199)

[Описание JUnit 5 2](#_Toc35431200)

[Требования 3](#_Toc35431201)

[Код с комментариями 4](#_Toc35431202)

[Найденные баги 4](#_Toc35431203)

[Выводы и заключение 4](#_Toc35431204)

# Цель

Изучить JUnit 5, использовать полученные знания на практике, а именно: реализовать консольное приложение по варианту по сформулированным мной требованиям и прокрыть его модульными тестами на 70%. Посмотреть, какие баги будут найдены и определить для себя важность модульного тестирования и какие проблемы оно решает или помогает избегать, насколько оно полезно и как влияет на мою производительность.

# Описание JUnit 5

JUnit 5 состоит из нескольких разных модулей из трех разных подпроектов: П**латформа JUnit**, **Юнит Юпитер** и **Юнит Винтаж**.

Платформа отвечает за запуск сред тестирования на JVM. Он определяет стабильный и мощный интерфейс между JUnit и его клиентом, таким как инструменты сборки. Конечная цель - как легко интегрировать своих клиентов с JUnit в обнаружение и выполнение тестов. Он также определяет [TestEngine](http://junit.org/junit5/docs/current/api/org/junit/platform/engine/TestEngine.html)API для разработки инфраструктуры тестирования, работающей на платформе JUnit (IntelliJ IDEA, Eclipse, NetBeans, and Visual Studio Code). Таким образом, вы можете подключить сторонние библиотеки тестирования непосредственно в JUnit, реализовав собственный TestEngine.

JUnit Jupiter - расширение для написания тестов и расширений в JUnit 5.

JUnit Vintage предоставляет платформу для запуска тестов на основе JUnit 3 и JUnit 4.

Основные аннотации

* @BeforeAll **и**@BeforeEach (перед всеми сразу или перед каждым по отдельности тестами в классе; важно отметить, что метод с аннотацией @BeforeAll должен быть статическим, иначе код не будет компилироваться)
* @DisplayName **и**@Disabled (изменение отображаемого имени и отключение с комментарием)
* @AfterEach **и**@AfterAll (выполнение после каждого теста в классе или после всех)

В JUnit 5 есть два способа тестирования исключений. Оба они могут быть реализованы с помощью метода assertThrows().

@Test

void shouldThrowException() {

Throwable exception = assertThrows(UnsupportedOperationException.class, () -> {

throw new UnsupportedOperationException("Not supported");

});

assertEquals(exception.getMessage(), "Not supported");

}

@Test

void assertThrowsException() {

String str = null;

assertThrows(IllegalArgumentException.class, () -> {

Integer.valueOf(str);

});

}

Первый пример используется для проверки более подробной информации о сгенерированном исключении, а второй только проверяет тип исключения.

@SelectPackages и @SelectClasses – аннотации для создания тестовых наборов (список пакетов или список классов).

@TestFactory – для создания динамических тестов.

# Требования

Веб-скрапер. Создайте инструмент, принимающий URL в качестве input и возвращающий контент в виде HTML или XML.

Функциональные требования:

1. Система должна представлять собой консольное приложение и запускаться из консоли.
2. Системе должны передаваться входные параметры, определенные в таблице 1.  
     
   Таблица 1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Параметр 1 | Параметр 2 (опционально) | Параметр 3 (опционально) | Параметр 4 (опционально) | Параметр 5 (опционально) |
| -h | Ничего | Ничего | Ничего | Ничего |
| Строка URL скачиваемого ресурса | Ничего / -text / -body / -head / -all | Ничего | Ничего | Ничего |
| -с | Ничего / -text / -body / -head / -all | Ничего | Ничего |
| -f | Ничего / -text / -body / -head / -all | Ничего / -txt / -html / -xml | Ничего / Название выходного файла |

1. Система должна проверять корректность входных данных.
2. Система должна проверять входные данные на правильность их порядка.
3. В случае неправильного ввода входных параметров система должна вывести подробное описание ошибки в консоль согласно таблице 2.  
     
   Таблица 2.

|  |  |
| --- | --- |
| Случай с ошибкой | Текст исключения |
| Введено 0 параметров | Please provide URL of downloading resource. For help use command -h. |
| Введено более 5ти параметров | You can’t input more than 5 parameters! For help use command -h. |
| После команды -h введены ещё какие-то параметры | -h command has no arguments! For help use command -h. |
| URL не существует | This URL address doesn’t exist! |
| URL введён правильно, далее следует команда -f и неправильное название выходного файла | Invalid filename! |
| Первый аргумент не URL и не -h или URL введён неверно | Invalid URL! Please, write URL correctly. For help use command -h. |
| Первый аргумент URL, второй -c, третий – не аргумент скачиваемой части контента по URL | After -c output type you can write only scrapping document part argument! (-text / -body / -head / -all) For help use command -h. |
| Первый аргумент URL, второй не тип вывода данных и не аргумент скачиваемой части контента по URL | You must write output type command or part of scrapping page after URL! For help use command -h. |
| Первый аргумент URL, второй не тип вывода данных и дальше ещё идут аргумент(ы) | You can write other args only after type command (-c or -f)! For help use command -h. |
| Первый аргумент URL, второй -f, третий не аргумент скачиваемой части контента по URL и не формат сохраняемого файла | After -f must be part of scrapping page or file type argument before filename! For help use command -h. |
| Нарушен порядок в цепочке из пяти входных параметров (первый аргумент URL, второй -f, дальше – путаница) | You must write arguments in order: url, -f, part of scrapping page argument, file type argument, filename! For help use command -h. |

1. В случае возникновения непредвиденных исключений при получении данных по URL, система должна вывести стектрейс.
2. В случае возникновения непредвиденных исключений при сохранении данных в файл, система должна вывести стектрейс.
3. Система должна выводить справку по команде -h.
4. Справка должна выглядеть так: «».
5. Система должна скачивать данные по переданному ей URL.
6. Система должна скачивать определённые данные со страницы в соответствии с переданным аргументом типа данных со страницы.
7. В случае, если системе такой аргумент не передан, система должна скачивать весь html контент по URL.
8. Если системе был передан аргумент -all, она должна скачать весь html контент страницы.
9. Если системе был передан аргумент -head, она должна скачать head html контента страницы.
10. Если системе был передан аргумент -body, она должна скачать весь body html контента страницы.
11. Если системе был передан аргумент -text, она должна скачать весь текст страницы без html тегов, преобразуя страницу из формата html в текстовый.
12. Система должна выводить данные в консоль по команде -с.
13. Система должна сохранять скаченные данные в файл по команде -f.
14. В случае если не использованы никакие вспомогательные команды, система должна выводить данные в консоль.
15. В случае, если не приведено конкретное название файла, система должна сохранить скаченные данные в файл, название которого будет равно URL.
16. В случае, если системе дано конкретное название файла, система должна сохранить скаченные данные в файл с таким названием.
17. Если файла с конкретным переданным системе названием не существует, система должна создать такой файл и сохранить в него скаченные данные.
18. В случае, если файл с введённым пользователем названием уже существует и заполнен, система должна записать скаченные данные в него поверх ранее существовавших данных.
19. В случае, если файл с названием, соответствующим URL, уже существует и заполнен, система должна записать скаченные данные в него поверх ранее существовавших данных.
20. Система должна проверять валидность URL.
21. Система должна сохранять данные в формате в зависимости от переданного аргумента типа сохраняемого файла.
22. Система должна дополнить название файла или URL до указания формата файла в соответствии с типом сохраняемого файла.
23. В случае, если системе был передан аргумент -txt, система должна сохранить файл в формате TXT.
24. В случае, если системе был передан аргумент -xml, система должна сохранить файл в формате XML.
25. В случае, если системе был передан аргумент -html, система должна сохранить файл в формате HTML.
26. В случае, если аргумента нет, система должна сохранить данные в файл формата TXT.

# Код с комментариями

# Найденные баги

# Выводы и заключение