УДК 004.4
Шкода Д.С.[1], Куликовська Н.А.[2]
[1] студ. гр. КНТ-511сп НУ «Запорізька Політехніка»
[2] старш.викл. НУ «Запорізька Політехніка»

ПОРІВНЯННЯ ПОПУЛЯРНИХ ТЕСТОВИХ ФРЕЙМВОРКІВ JUNIT І TESTNG

Тестові фреймворки – це інструменти, що полегшують процес ство-рення, виконання та обробки результатів тестів для програмного забезпечен-ня.

JUnit і TestNG, безперечно, є двома найбільш популярними фрейм-ворками для модульного тестування (unit -тестування) в екосистемі Java. Хоча JUnit послужив натхненням для TestNG, другий має низку відмінностей і, на відміну від JUnit, працює для функціонального та вищих рівнів тестуван-ня.

TestNG реалізує core логіку JUnit, який старший за нього і послужив натхненням для першого. Але TestNG (NG означає Next Generation) спочатку створювався для функціонального і вищих рівнів тестування, позиціонувався як більш простий у використанні, ніж заслужив на визнання автоматизаторів.

1. Анотації життєвого циклу тестів

Junit5 має лише чотири анотації на позначення методів, що викону-ються перед чи після виконання тестів: @BeforeEach, @BeforeAll, @AfterEach та @AfterAll. Проте TestNG має можливість розширити ці методи шляхом наслідування від інтерфейсів BeforeAllCallback, BeforeEachCallback, BeforeTestExecutionCallback (для методів, які виконуватимуться прямо перед викликом тест методу, тобто після методів BeforeEach) та відповідних “after” інтерфейсів.

Таблиця 1 – Порівняльні характеристики

Опис	TestNG	JUnit 5
Виконується перед першим тестом класу	@BeforeClass	@BeforeAll
Виконується після останнього тесту класу	@AfterClass	@AfterAll
Виконується перед кожним тестом	@BeforeMethod	@BeforeEach
Виконується після кожного тесту	@AfterMethod	@AfterEach
Виконується перед набором тестів	@BeforeSuite	NA
Виконується після набору тестів	@AfterSuite	NA
Виконується перед тестами, що визначені всередині тегу test.	@BeforeTest	NA
Виконується після тестів, що визначені всередині тегу test.	@AfterTest	NA

Продовження таблиці 1

Опис	TestNG	JUnit 5
Виконується перед першим тестом групи	@BeforeGroups	NA
Виконується після останнього тесту групи	@AfterGroups	NA

2.Компонування анотацій.

Junit5 має можливість компонування анотацій за допомогою мета-анотацій – для цього власна анотація та анотації, які вона наслідує визначаються де інде. Зручність полягає у тому, що при потребі зміни компонованої анотації, це можна зробити один раз у місці її визначення, а не шукати кожне її входження у коді.

Таблиця 2 – Порівняльні характеристики анотацій

TestNG	JUnit 5

@Test(groups = {"smoke", "ui"})
public void test ()
{
}
	@Target({ ElementType.TYPE, ElementType.METHOD })
	@Retention(RetentionPolicy.RUNTIME)
	@Tag("smoke")
	@Tag("ui")
	public @interface SmokeUI {}
	@SmokeUI
	@Test public void test () {}

3.Підтримка м’яких асертів

Для виконання м'яких асертів у TestNG необхідно створювати об'єкт, викликати його методи усюди, де треба зробити перевірку та викликати асерт ол у кінці. Це створює нагромодження зайвого коду у тестах і робить їх менш читабельними. Junit5 виконує м’які асерти за допомогою лямбд, а також підтримує залежні асерти, що робить структуру тесту більш акуратною.

Таблиця 3 – Порівняльні характеристики асертів

TestNG	JUnit 5

@Test
Public void test () {
SoftAssert softassert = new SoftAssert();
softassert.assertEquals(result1,expected1);
softassert.assertEquals(result2,expected2);
softassert.assertAll();
}
	@Test
	public void test () {
	assertAll(
	() -> assertEquals(result1, expected1),
	() -> assertEquals(result2, expected2))}

4.Засоби розширення поведінки.

У TestNG засоби розширення представлені класами та інтерфесами, що наслідують інтерфейс-маркер Listener. Зручність лістенерів тестнг полягає у тому, що можна визначити безліч різноманітних лістенерів і підключати потрібні залежно від ситуації у файлі теснг хмл, не проходячись при цьому по всьому кодові і не змінюючи тестів вручну.

У Junit5 розширення реалізуються за допомогою Extension API.

Таким чином більшість дій, які можна виконати за допомогою JUnit5, мають засоби для аналогічних дій у TestNG.

5.Засоби параметризації.

Для можливості параметризувати JUnit5 тести до проекту необхідно підключати залежність junit-jupiter-params. Таке розділення на модулі може видаватися незручним порівняно з TestNG. Щоб створити параметризований тест замість звичайної анотації @Test потрібно вказати @ParameterizedTest та джерело даних.

6.Генерація звітів.

TestNG має вбудований репортер, який генерує результати у форматі HTML. TestNG також має засоби для створення користувацьких репортів шляхом імплементації інтерфейсу IReporter.

JUnit 5 не має вбудованих засобів для створення репортів. Таким чином генерація репортів відбувається за рахунок інструменту збірки. Одним з підходів до створення html-репортів JUnit 5 тестів є застосування плагіну Surefire Report. Плагін Surefire може використовуватися також і для виконання TestNG тестів.

В обох фреймворків при-близно однаковий функціонал. Є невеликі розбіжності у реалізації, і навіть, що менш істотно, у назві анотацій. Зараз вибір між JUnit або TestNG - це скоріше особисті переваги, або вже використовується на проекті стек техно-логій.

TestNG завжди важливий для більш потужних та привабливих проектів, особливо для великих проектів, але JUnit користується популярністю та добрим підходом для більш простих та швидких тестів.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Стась М. І., Порівняння тестових фреймворків JUnit та TestNG в екосистемі Java. / М.І.Стась, І.В.Холявко // Системи обробки інформації. 2021. – Вип. 1 (164). – С. 136-142. DOI: 10.30748/soi.2021.164.18.

2. Петренко С.А., Засоби розширення можливостей тестових фреймворків JUnit та TestNG. / С.А.Петренко, О.О.Бодров // Вісник Вінницького політехнічного інституту. 2020. – № 5. – С. 73-79. DOI: 10.31649/1997-9266-2020-152-5-73-79.

3. Кудрявцев Д. В., Порівняльний аналіз систем параметризації тестів у фреймворках JUnit та TestNG. / Д.В.Кудрявцев, А. В. Смирнов // Наукові вісті Далівського університету. 2019. № 16. С. 54-60.

4. Гороховський О. І., Генерація звітів з результатами тестування у фреймворках JUnit та TestNG. / О.І. Гороховський, В.І. Пожуєв // Проблеми інформатизації та управління. 2018. – Вип. 3 (55). – С. 27-32.