

The image shows the JUnit logo, which consists of the text "JUnit" in a light blue, sans-serif font. The text is positioned on the right side of the image. The background is a dark olive green with a subtle gradient and some faint, wavy lines at the top.

JUnit

Модульне тестування

Модульне тестування (юніт-тестування) - перевірка роботи окремих модулів.

Під модулем розуміється один клас або група пов'язаних класів. Такий модуль розглядається ізольовано.

Якщо він залежить від інших програм програм (наприклад, бази даних або мережевого з'єднання) його викокремлюють за допомогою спеціальних "заглушок". При цьому вважається, що оточення тестованого модуля працює коректно.

Цей вид тестування зазвичай виконується программістом-розробником класу. Зазвичай перевіряється функціональність коду, деколи продуктивність.

TEST-DRIVEN DEVELOPMENT

З тестуванням пов'язані деякі методики екстремального програмування, зокрема регресивне програмування та TDD.

Розробка через тестування (TDD - test-driven development) - одна з практичних програм екстремального програмування, яка передбачає розробку тестів до написання коду класів. Тобто спочатку розробляється набір тестів, який повинен пройти розроблений програмний модуль, а потім – код самої програми.

При зміні коду програмного модуля розроблені тести постійно контролюють працездатність модуля.

При додаванні нового функціоналу в програмний модуль, відповідно, спочатку додаються модульні тести.

ІНСТРУМЕНТИ МОДУЛЬНОГО ТЕСТУВАННЯ В JAVA

1. Фреймворки для написання та запуску тестів: *JUnit*, *TestNG*.
2. Бібліотеки перевірок: *FEST Assert*, *Hamcrest*, *XMLUnit*, *HTTPUnit*.
3. Бібліотеки для створення тестових дублерів: *Mockito*, *JMock*, *EasyMock*.

Бібліотеки перевірок, дозволяють розширити можливості перевірки результатів тестів, які мають є базовому фреймвору

Бібліотеки тестових дублерів дозволяють імітувати роботу з реальними модулями (наприклад, базою даних) через спеціальні об'єкти **Stub**-и та **Mock**-и.

JUNIT

JUnit - це один із найбільш популярних фрейм ворків для розробки юніт-тестів під Java.

Він був створений Кентом Беком і Еріком Гамою (перша версія фреймворку для мови Small talk з'явилася в 1999 р).

Аналогічні оболонки для модульного тестування існують і для інших мов програмування. Вони носять спільну назву - сімейство XUnit.

Найбільш популярним сьогодні є версії Junit 4.x, хоча на офіційному сайті junit.org в повному обсязі пропонується серія Junit 5.

Відмінності між поколіннями Junit (3, 4 чи 5) можуть виявитися суттєвими насамперед на рівні іменування методів та анотацій. Для зворотної сумісності створено проект Jupiter.

Тести JUnit

Тіло тесту повинно відповідати підходу AAA (Arrange, act, assert).

Це означає, що спочатку виконуються деякі підготовчі дії (arrange).

Наприклад, створюється об'єкт тестованого класу.

Потім запускається тестований метод (act).

І, нарешті, перевіряють результат його роботи (assert). Для такої перевірки якраз і використовуються методи класу Assert.

Анотації JUnit

@Test - вказує, що метод (**public void**) можна запустити, як тестовий випадок.

@BeforeEach методи з такою анотацією запускаються перед кожним тестом.

@AfterEach методи з такою анотацією запускаються після кожного тесту.

@BeforeAll метод виконується один раз перед запуском будь-якого тестового методу класу (метод повинен бути статичним).

@AfterAll метод виконується один раз після запуску будь-якого тестового методу класу (метод повинен бути статичним).

@Ignore тестові методи з такою анотацією не будуть виконуватися в поточному запуску.