1. Что такое процесс тестирования? Для чего?

Процесс тестирования - это процесс проверки программного продукта на соответствие заданным требованиям, обнаружения ошибок и предотвращения возможных проблем при эксплуатации. Тестирование необходимо для повышения качества продукта и уменьшения риска возникновения ошибок в работе приложения.

1. Что такое TestNG и Junit?

TestNG и Junit - это фреймворки для тестирования Java-приложений. Они позволяют автоматизировать процесс тестирования, обнаруживать ошибки и создавать отчеты о прохождении тестов. TestNG предоставляет дополнительные возможности для тестирования в сравнении с Junit.

1. Что такео юнит тестирование и какие его основные признаки?

Юнит-тестирование - это метод тестирования, при котором тестируются отдельные модули программы: классы, методы, функции и т.д. Основные признаки юнит-тестирования: тестирование изолированных компонентов, автоматизированное тестирование, повторяемость тестов, тестирование с разными входными данными и проверка результата.

1. Что таое тест-кейс?

Тест-кейс - это инструкция для проведения тестирования определенной функциональности программного продукта. Он содержит описание того, что нужно сделать, чтобы протестировать определенный аспект приложения, какие данные использовать и что ожидается от результата.

1. Назначение @Test аннтоции?

Аннотация @Test в TestNG и Junit используется для обозначения методов, которые будут выполняться как тестовые. Она указывает, что данный метод должен быть запущен во время тестирования и что его результаты должны быть проверены на соответствие заданным требованиям.

1. Что такое test suit? Как его создать?

Test suit - это набор тестов, запускаемых одновременно для проверки большого числа функциональных возможностей приложения. TestNG позволяет создавать test suit, используя XML-файлы, в которых определяются наборы тестов для выполнения. (НА САМОМ ДЕЛЕ, ЭТО НЕВОЗМОЖНО СДЕЛАТЬ!!!!!!!!)

1. Что обеспечивает Assert class?

Assert class обеспечивает проверку результатов выполнения тестовых методов. Она сравнивает ожидаемые результаты с фактическими и генерирует исключение, если результат не соответствует ожиданиям.

1. Поясните назначение @BeforeSuite / @AfterSuite @BeforeTest / AfterTest @BeforeClass / @AfterClass @BeforeMethod / @AfterMethod?

Аннотации @BeforeSuite и @AfterSuite используются для выполнения методов до и после выполнения всех тестовых классов.

Аннотации @BeforeTest и @AfterTest запускают методы до и после выполнения набора тестов.

Аннотации @BeforeClass и @AfterClass используются для выполнения методов до и после классов тестов.

Аннотации @BeforeMethod и @AfterMethod запускают методы перед и после каждого тестового метода.

1. Как и для чего используют параметризацию в тестировании?

Параметризация тестов позволяет использовать разные входные данные для выполнения одного и того же теста. Это позволяет убедиться, что приложение работает корректно для разных условий и входных данных.

1. Что такео TDD и DDT?

TDD (Test-Driven Development) - это метод разработки программного обеспечения, основанный на написании тестов перед написанием самого кода. DDT (Data-Driven Testing) - это метод тестирования, при котором тестирование осуществляется с использованием множества разных входных данных, полученных из внешнего источника данных.