1. Что такое процесс тестирования? Для чего?   
Тест Java — это простой класс java, методы которого помечены аннотацией @Test.

Задача такого тестирования — убедиться, что код работает как надо. И нам совершенно без разницы, как конкретно реализовать этот метод. Это и есть главный ответ на вопрос, заданный в начале урока. Тесты проверяют, что код (или приложение) работает корректно.

2. Что такое TestNG и Junit?

Фреймворки для модульного тестирования

3. Что такое юнит тестирование и какие его основные признаки?

**Модульное тестирование (Unit Testing)**– это тип тестирования программного обеспечения, при котором тестируются отдельные модули или компоненты программного обеспечения. Его цель заключается в том, чтобы проверить, что каждая единица программного кода работает должным образом.

4. Что такое тест-кейс?

Тест кейс — это проверка работоспособности программы или проекта.

5. Назначение @Test аннотации?

@Test: этот аннотированный метод помечает любой класс или метод как часть теста, который необходимо протестировать.

6. Что такое test suit? Как его создать?

**Тест Сьют** это набор [тест кейсов](http://software-testing.org/testing/chto-takoe-test-keys-test-case-iz-kakih-poley-sostoit-tipichnyy-test-keys.html), которые объединены тем что относятся к одному тестируемому модулю, функциональности, приоритету или одному типу тестирования. Каждый тест сьют состоит из более чем одного тест кейса и зачастую выполняется всей «пачкой» в процессе тестирования.

7. Что обеспечивает Assert class?

Assert — это специальная конструкция, позволяющая проверять предположения о значениях произвольных данных в произвольном месте программы.

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

1. @BeforeSuite – указывает, что данный метод будет запускаться перед любым методом тестового класса.

2. @AfterSuite – указывает, что данный метод, будет запускаться после всех методов тестового класса.

3. @BeforeTest – аннотированный метод будет запускаться до всех тестовых методов.

4. @AfterTest – аннотированный метод будет запущен после всех тестовых методов, принадлежащих классам внутри тега <test>.

5. @BeforeClass – указывает, что метод будет выполнен до всех тестовых методов тестового класса.

6. @AfterClass – аннотированный метод будет запущен после всех тестовых методов в текущем классе.

7. @BeforeMethod – аннотированный метод будет выполняться перед каждым тестовым методом.

8. @AfterMethod – аннотированный метод будет запускаться после каждого тестового метода.

9. Как и для чего используют параметризацию в тестировании?

Параметризованный тест должен выполнять один и тот же тест снова и снова, используя разные значения. Это помогает разработчику сэкономить время при выполнении одного и того же теста, который отличается только своими входами и ожидаемыми результатами.

10. Что такое TDD и DDT?

TDD, test-driven development или процесс разработки через тестирование — это методология разработки программного обеспечения, которая основывается на повторении коротких циклов разработки: изначально пишется тест, покрывающий желаемое изменение, затем пишется программный код, который реализует желаемое поведение системы и позволит пройти написанный тест, а затем проводится рефакторинг написанного кода с постоянной проверкой прохождения всех тестов.

Data Driven Testing (DDT) — подход к созданию/архитектуре автоматизированных тестов (юнит, интеграционных, чаще всего применимо к backend тестированию), при котором тест умеет принимать набор входных параметров, и эталонный результат или эталонное состояние, с которым он должен сравнить результат, полученный в ходе прогонки входных параметров.