**Задание 1. Ответьте письменно на вопросы:**  
  
1)  Почему использование тестовых заглушек может быть полезным при написании модульных тестов?  
  
2) Какой тип тестовой заглушки следует использовать, если вам нужно проверить, что метод был вызван с определенными аргументами?  
  
3) Какой тип тестовой заглушки следует использовать, если вам просто нужно вернуть определенное значение или исключение в ответ на вызов метода?  
  
4) Какой тип тестовой заглушки вы бы использовали для имитации  взаимодействия с внешним API или базой данных?

**Ответьты:**

1) Концепция модульных тестов заключается в легкости и понятности кода теста, а также в скорости тестирования. Не всегда объекты, которые нужны для проверки методов класса представлены простыми значениями, чаще всего это объекты со сложной структурой. Но на момент тестирования нам не важно как создается такой объект и насколько корректно. Нам нужно реализовать зависимость нашего тестируемого класса от такого сложного объекта. Тут нам на помощь и приходят заглушки. Засчет подмены сложного объекта более легковесным тесты проходят быстрее и легче. Бывают ситуации необходимой проверки работы с базой данных, запросов на сервера, что значительно замедлит работу теста, поэтому и тут необходимы заглушки. Они позволят проверить корректность методов взаимодействия тестируемого класса с БД или сервером путем иммитации такого взаимоотношения не совершая реальных подключений, а совершив моделироание такого подключения.

2) Тестовая заглушка mock позволяет проследить вызывался ли метод и с какими параметрами он вызывался. Благодаря библиотеке Mockito мы получаем доступ к методам, которые достаточно понятно и лаконично организовывают такую работу. (when…thenReturn, verify)

3) Если нужно вернуть определенное значения, то используется stub. Настроить проброс исключений можно и с использованием фреймворка Mockito в кострукции when(…).thenThrow(new …Exception). В лекционном материале также рассматривали использование фэйк-заглушки как вариант предоставить простые и быстрые значения.

4) На семинаре мы рассматривали итерфейс подключения к базе данных, на основе которого делали stub заглушку. Иммитацию БД в стаб заглушке представляли в виде поля типа Map. Это позволило протестировать работу методов, связанных с взаимодействием с БД без физического подключения. На лекции предлагалось заменить реальный объект на эмулятор для конкретного приложения БД в памяти (fake database) или фальшивый веб-сервис с помощью простой и легковесной организации работы внутри фейк-заглушки. Собственно и с помощью тех же mock-ов можно протестировать работу с БД, создав интерфей на основе реальлной БД и передать его в качестве объекта-основы в конструктор создания mock-а