[Задание 1. Ответьте письменно на вопросы:](https://github.com/San4ousoff/UnitTests/blob/master/UnitTests/src/test/java/Seminar4/HW/textHW.md#%D0%B7%D0%B0%D0%B4%D0%B0%D0%BD%D0%B8%D0%B5-1-%D0%BE%D1%82%D0%B2%D0%B5%D1%82%D1%8C%D1%82%D0%B5-%D0%BF%D0%B8%D1%81%D1%8C%D0%BC%D0%B5%D0%BD%D0%BD%D0%BE-%D0%BD%D0%B0-%D0%B2%D0%BE%D0%BF%D1%80%D0%BE%D1%81%D1%8B)

**1. Почему использование тестовых заглушек может быть полезным при написании модульных тестов?**

* В ходе тестирования отдельных изолированных модулей тестовые заглушки обеспечивают максимальную изоляцию модуля от зависимостей, таких как внешние ресурсы или другие модули, что позволяет избежать непредсказуемого внешнего воздействия на результат.
* Тестовые заглушки позволяют проще воспроизвести сложные сценарии.
* Тестовые заглушки позволяют увеличить скорость тестирования

**2. Какой тип тестовой заглушки следует использовать, если вам нужно проверить, что метод был вызван с определенными аргументами?**

Для проверки, что метод был вызван с определенными аргументами, следует использовать Mock. Mock предназначен для проверки конкретных взаимодействий с зависимостями.

**3. Какой тип тестовой заглушки следует использовать, если вам просто нужно вернуть определенное значение или исключение в ответ на вызов метода?**

Если нужно просто вернуть определенное значение или исключение в ответ на вызов метода, вы можете использовать Stub. Stub позволяет подставить фиксированные или предварительно настроенные ответы

**4. Какой тип тестовой заглушки вы бы использовали для имитации взаимодействия с внешним API или базой данных?**

Для имитации взаимодействия с внешним API или базой данных можно использовать Fake. Fake используется для замены внешнего тяжеловесного объекта его лёгкой эмуляцией.