Модульное тестирование на Java

План:

- Виды тестирования ПО. Модульное тестирование.
- Обзор JUnit и Mock тестирования.
- Применение JUnit и Mocks на примерах.
- Полезные ресурсы и рекомендации для дальнейшего изучения темы.

Тестирование ПО

- **Тестирование программного обеспечения (Software Testing)** - проверка соответствия между реальным и ожидаемым поведением программы, осуществляемая на конечном наборе тестов, выбранном определенным образом. *[IEEE Guide to Software Engineering Body of Knowledge, SWEBOK, 2004]*

Виды тестирования ПО

По целям тестирования:

- Функциональное
- Нефункциональное:

- пользовательского интерфейса
- удобства использования
- специальных возможностей
- безопасности
- инсталляционное
- конфигурационное
- совместимости
- отказоустойчивости
- локализации
- интернационализации
- производительности

Виды тестирования ПО

По степени автоматизации:

- Ручное
- Полуавтоматизированное
- Автоматизированное

Виды тестирования ПО

По позитивности сценария:

- Позитивное
- Негативное

Виды тестирования ПО

По знанию системы:

- Белого ящика
- Серого ящика
- Черного ящика

Виды тестирования ПО

По разработке тестовых сценариев:

- На основе требований
- По пользовательским сценариям
- На основе моделей

Виды тестирования ПО

По исполнителям тестирования:

- Альфа-тестирование
- Бета-тестирование

Виды тестирования ПО

По уровню тестирования:

- Модульное
- Интеграционное
- Системное

Виды тестирования ПО

По критерию запуска программы:

- Динамическое
- Статическое

Виды тестирования ПО

По уровню формальности:

- По тест-кейсам
- Исследовательское
- Свободное

Виды тестирования ПО

По временным критериям:

- Комплексное
- Тестирование сборки
- Входное (дымовое)
- Санитарное
- Повторное
- Регрессионное
- Приемочное

Модульное тестирование

Модульное тестирование - это проверка на корректность отдельных модулей исходного кода программы.

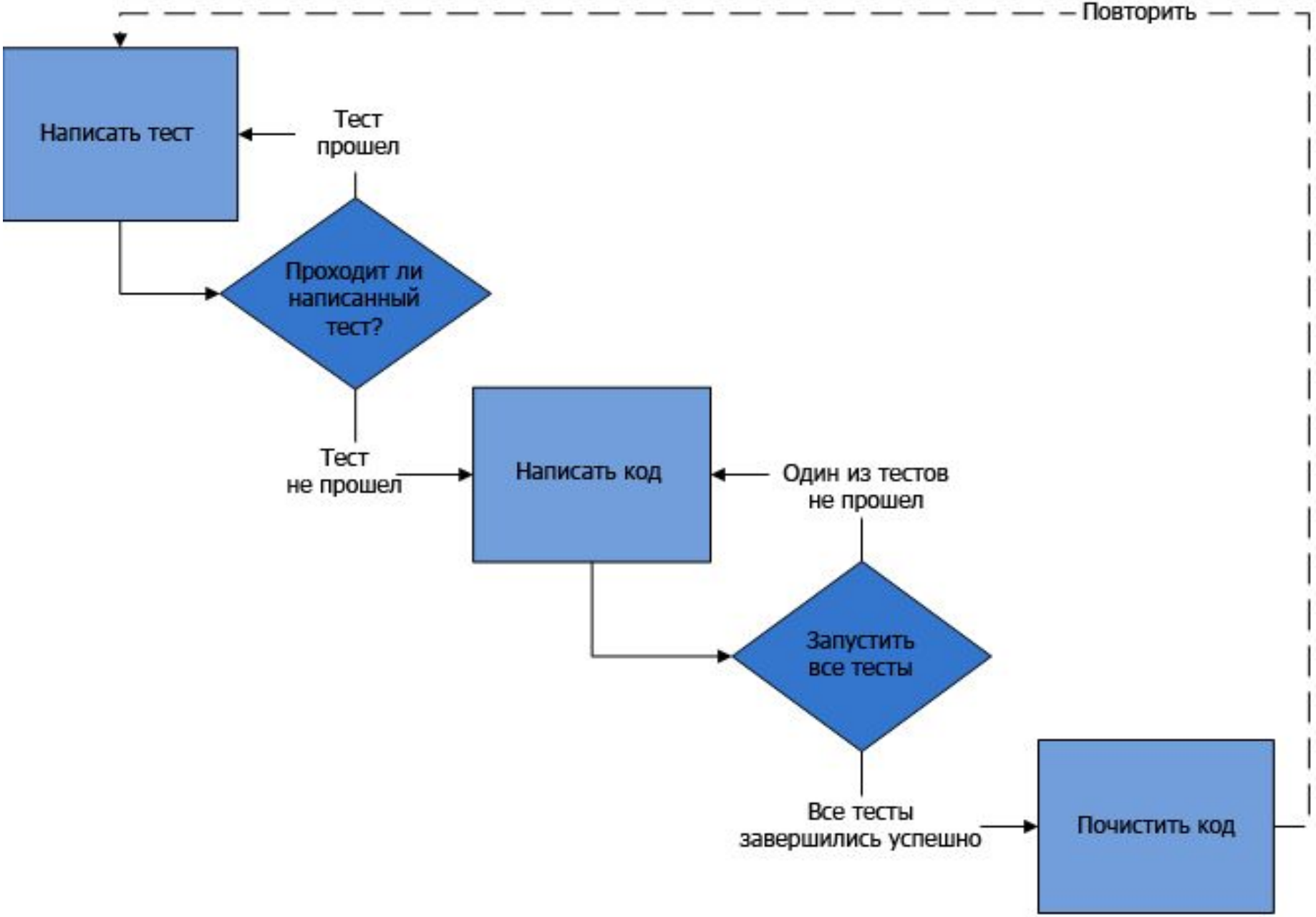
Это тестирование единицы системы (класса или модуля).

Инструменты и библиотеки модульного тестирования для Java

- JUnit
- TestNG
- JavaTESK
- Spock
- Java Mock Frameworks:

Mockito, EasyMock, Mockachino, PowerMock, JMock, JMockit, Unitils

TDD – разработка через тестирование



JUnit

JUnit – библиотека (фреймворк) для модульного и регрессионного тестирования на Java, который служит для написания и запуска тестов

JUnit. Основные аннотации

Аннотация	Описание
@Test public void method()	Аннотация **@Test** определяет что метод method() является тестовым.
@Before public void method()	Аннотация **@Before указывает** на то, что метод будет выполнятся перед каждым тестируемым методом @Test.
@After public void method()	Аннотация **@After указывает** на то что метод будет выполнятся после каждого тестируемого метода @Test
@BeforeClass public static void method()	Аннотация **@BeforeClass** указывает на то, что метод будет выполнятся в начале всех тестов, а точней в момент запуска тестов(перед всеми тестами @Test).
@AfterClass public static void method()	Аннотация **@AfterClass** указывает на то, что метод будет выполнятся после всех тестов.
@Ignore	Аннотация **@Ignore** говорит, что метод будет проигнорирован в момент проведения тестирования.
@Test (**expected** = Exception.class)	(**expected** = Exception.class) — указывает на то, что в данном тестовом методе вы преднамеренно ожидается Exception.
@Test (**timeout**=100)	(**timeout**=100) — указывает, что тестируемый метод не должен занимать больше чем 100 миллисекунд.

JUnit. Основные методы

Метод	Описание
fail(String)	Указывает на то что бы тестовый метод завалился при этом выводя текстовое сообщение.
assertTrue([message], boolean condition)	Проверяет, что логическое условие истинно.
assertsEquals([String message], expected, actual)	Проверяет, что два значения совпадают. *Примечание*: для массивов проверяются ссылки, а не содержание массивов.
assertNull([message], object)	Проверяет, что объект является пустым **null**.
assertNotNull([message], object)	Проверяет, что объект не является пустым **null**.
assertSame([String], expected, actual)	Проверяет, что обе переменные относятся к одному объекту.
assertNotSame([String], expected, actual)	Проверяет, что обе переменные относятся к разным объектам.

Demo

Исходный код:

https://github.com/ViktoriiaSilenko/jUnitExamples

Mockito

Mockito используется для создания заглушек интерфейсов, так что макет функционала может быть добавлен в мок интерфейса, который может использоваться в модульном тестировании.

Demo

Исходный код:

https://github.com/ViktoriiaSilenko/mockitoExamples

Полезные ресурсы

- http://www.protesting.ru/testing/
- http://qa-helper.com/testing-types/
- http://www.tutorialspoint.com/junit/
- http://www.tutorialspoint.com/mockito/
- https://habrahabr.ru/post/243155/
- https://habrahabr.ru/post/120101/
- http://devcolibri.com/864
- http://junit.org/junit4/
- http://mockito.org/
- http://easymock.org/
- http://code.google.com/p/powermock/
- http://www.jmock.org/

Примеры:

https://github.com/ViktoriiaSilenko/jUnitExamples

https://github.com/ViktoriiaSilenko/mockitoExamples