LUXOFT TRAINING

Разработка через тестирование XUnit

Test Driven Development

Ivan Dyachenko <IDyachenko@luxoft.com>

Module 2

Для кого этот тренинг?

LUXOFT Training Center

1. **Beginner**
   Хорошая точка входа
2. **Intermediate**
   Поможет лучше всё структурировать в голове и объяснять коллегам
3. **Advanced**
   Можно использовать для обучения и проверки других

Содержание

LUXOFT Training Center

1 xUnit

2

3

4

5

6

7

xUnit

LUXOFT Training Center

- Как и в любом другом деле, в модульном тестировании не обойтись без подходящих инструментов – нет смысла «забивать гвозди микроскопом»
- Для этого существуют xUnit и Mock-фреймворки, применяемые для state-based и interaction тестирования соответственно

Семейство xUnit

LUXOFT Training Center

- Самыми яркими представителями семейства xUnit являются фреймворки JUnit (для Java) и его портированная под .NET версия – Nunit
- Синтаксис обоих фреймворков практически идентичен, поэтому рассмотрим аннотации и методы JUnit

Аннотации JUnit

LUXOFT Training Center

- В ходе написания модульных тестов у нас появляются как сами тестовые классы и методы, так и вспомогательные
- Для их разделения в среде JUnit, начиная с 4-й версии, используются аннотации
- Аннотация – ключевое слово, начинающееся с символа “@”, и помещаемое перед объявлением класса и/или метода

Фикстура

Фикстура – разделяемые между тестами данные и бизнес-логика

Аннотации JUnit

LUXOFT Training Center

@Before
@After
@BeforeClass
@AfterClass

- Объявляет метод фикстурой
- Данный метод будет вызван единожды, перед началом (после выполнения) тестового набора
- Используется для инициализации (очистки) тестовых данных и объектов

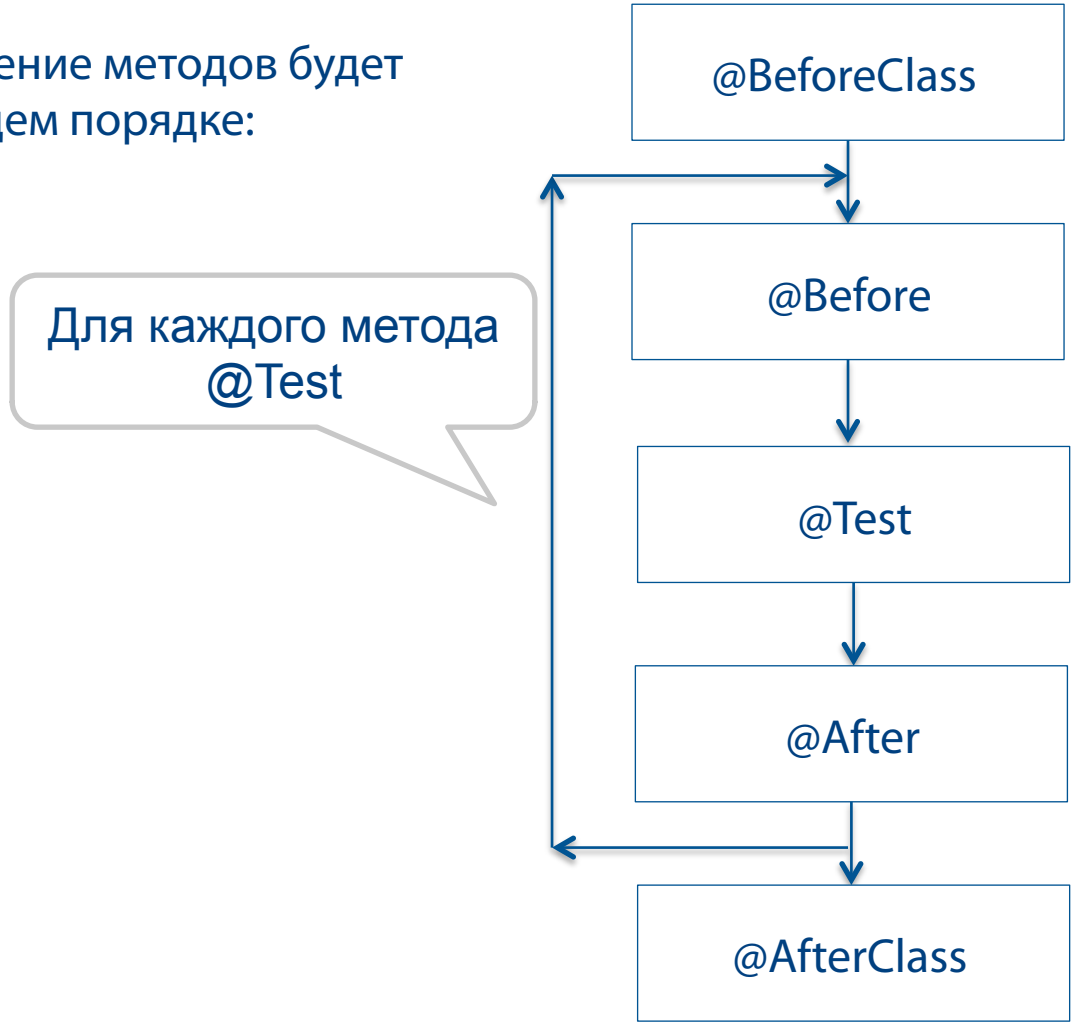
Аннотации JUnit

LUXOFT Training Center

@Before	▪ Объявляет метод фикстурой ▪ Данный метод будет вызываться перед началом (после завершения) КАЖДОГО тестового метода
@Test	▪ Объявляет метод тестом

Аннотации JUnit

Таким образом, выполнение методов будет выполняться в следующем порядке:



Пример

- TBD

Методы assert*

assertTrue(False)([message], condition)

Для проверки значений, возвращаемых тестируемыми методами, используются следующие методы JUnit:

- Проверка истинности выражения
- Опционально комментируется сообщением message

Методы assert*

assertEquals([message], obj1, obj2)

- проверка эквивалентности объектов
- перегружена для базовых классов
- опционально комментируется сообщением message

assert(Not)Null([message], obj)

- проверка на null
- опционально комментируется сообщением message

Использование JUnit

LUXOFT Training Center

- Итак, для написания модульного теста в среде JUnit нам необходимо:
  - создать класс
  - каждый тестовый случай описать в отдельном методе (с аннотацией @Test)
  - при необходимости написать методы инициализации / очистки (с аннотациями @Before/After[Class])
- Рассмотрим небольшой пример тестирования математических операций

Пример использования JUnit

LUXOFT Training Center

- TBD

Использование JUnit

LUXOFT Training Center

- Как мы видим, в последнем тестовом методе используется метод fail([message]);
- Этот метод используется для прямого сообщения фреймворку об ошибке (в данном случае – если не было вызвано ожидаемое исключение, или оно не того типа)

Запуск тестов из консоли

LUXOFT Training Center

```
java -cp \
C:\testing\lib\junit-4.5.jar; \
C:\testing\bin org.junit.runner.JUnitCore \
MathTestClass
```

Запуск тестов в IDE

В среде Eclipse\IntelliJ IDEA достаточно кликнуть правой кнопкой на тестовом классе и выбрать Run As -> JUnit Test

Запуск тестов в IDE

Результат:

Использование JUnit

LUXOFT Training Center

- Как мы видим, в последнем тестовом методе используется метод fail([message]);
- Этот метод используется для прямого сообщения фреймворку об ошибке (в данном случае – если не было вызвано ожидаемое исключение, или оно не того типа)

Вопросы ?

Разработка через тестирование

IDyachenko@luxoft.com

git clone git://github.com/ivan-dyachenko/Trainings.git

https://github.com/ivan-dyachenko/Trainings