



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ

Факультет компьютерных наук
Департамент программной инженерии

ОБЕСПЕЧЕНИЕ КАЧЕСТВА И ТЕСТИРОВАНИЕ

Семинар 4: Модульное тестирование (xUnit)
практика

Москва, 2020



UNITTEST НАЧАЛО РАБОТЫ

1. Подключить библиотеку unittest(import unittest)
2. Тесты должны наследоваться от unittest.TestCase



ТЕСТОВЫЕ МЕТОДЫ (ОБОЗНАЧЕНЫ ПРЕФИКСОМ `_TEST`)

- Аннотации

`@unittest.skip("demonstrating skipping")` — пропустить тест

`@unittest.skipIf(mylib.__version__ < (1, 3),`

`"not supported in this library version")` — пропустить тест если выполнено условие

`@unittest.skipUnless(sys.platform.startswith("win"), "requires Windows")`

`@unittest.expectedFailure` — обозначает что при выполнении теста ожидается исключение

- Также можно пропускать весь класс с тестами

`@unittest.skip("showing class skipping")`



БАЗОВЫЕ МЕТОДЫ В SELF

- `self.fail(String)` - Указывает на то, что бы тестовый метод завалился при этом выводя текстовое сообщение.
- `self.assertEquals` - проверяет, что два значения совпадают
- `self.assertArrayEquals` - сравнения содержимого массивов
- `self.assertTrue` - проверяет, что логическое условие истинно.
- `self.assertIsNone` - проверяет, что объект является none
- `self.assertIs` - проверяет, что обе переменные относятся к одному объекту



SETUP \ TEARDOWN

- **setUp()** - вызывается перед запуском каждого метода, здесь производится инициализация ресурсов требуемых для выполнения теста
- **tearDown()** - вызывается по завершению теста, в том числе после возникновения ошибки. В этом методе производится освобождение ресурсов



SETUPCLASS \ TEARDOWNCLASS

- **setUpClass()** - вызывается перед запуском всего набора тестов в классе, может использоваться для инициализации ресурсов требуемых для всех тестов, например, для запуска БД или http сервера
- **tearDownClass()** - вызывается по завершению всех тестов
- Оба метода должны сопровождаться аннотацией `@classmethod`



ПОЛЕЗНЫЕ МЕТОДЫ КЛАССА

- `.main()` - вызывает все тесты и выдает результат в консоль



ПОКРЫТИЕ КОДА

- **Покрытие кода** — мера, используемая при тестировании программного обеспечения. Она показывает процент исходного кода программы, который был выполнен в процессе тестирования.



ПОКРЫТИЕ КОДА

- На основе структурных элементов тестируемой системы, которые выполняются или задействуются в ходе тестирования.
- На основе структуры входных данных, используемых во время тестирования.
- На основе элементов требований, проверяемых при выполнении тестов.
- На основе явно сформулированных предположений об ошибках, выявление которых должны обеспечить тесты.
- На основе произвольных моделей устройства или функционирования тестируемой системы



ИНСТРУМЕНТЫ

- Coverage
- PyCoCo
- Pytest-cov



ЛИТЕРАТУРА

1. Мануал по тестам на питоне - <https://docs.python.org/3/library/unittest.html>
2. Про питон и тесты на хабре - <https://habr.com/ru/post/121162/>
3. Кент Бек – «Экстремальное программирование. Разработка через тестирование»
4. Gerard Meszaros – «xUnit Test Patterns»
5. JaCoCo – <https://pypi.org/project/pycoco/>
6. Цикломатическая сложность – https://ru.wikipedia.org/wiki/Цикломатическая_сложность

СПАСИБО! ВОПРОСЫ?



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ