

Языки прикладного программирования

Лекция 11

```
    for object to mirror
    mirror_mod.mirror_object = object

    if operation == "MIRROR_X":
        mirror_mod.use_x = True
        mirror_mod.use_y = False
        mirror_mod.use_z = False
    elif operation == "MIRROR_Y":
        mirror_mod.use_x = False
        mirror_mod.use_y = True
        mirror_mod.use_z = False
    elif operation == "MIRROR_Z":
        mirror_mod.use_x = False
        mirror_mod.use_y = False
        mirror_mod.use_z = True

    selection at the end - add
    mirror_ob.select= 1
    mirror_ob.select=1
    context.scene.objects.active = mirror_ob
    ("Selected" + str(modifier))
    mirror_ob.select = 0
    bpy.context.selected_objects = []
    data.objects[one.name].select = 1
    print("please select exactly one object")

- OPERATOR CLASSES -
types.Operator:
    X mirror to the selected object.mirror_mirror_x"
    "mirror X"
```

Тестирование ПО

Определение

- ❖ Тестирование – проверка соответствия между реальным и ожидаемым поведением программы, осуществляемая на конечном наборе тестов, выбранном определенным образом.

Для чего нужно тестирование?

Для чего нужно тестирование?

- ❖ **Чтобы определять некорректное поведение ПО и исправлять его**
- ❖ Помогает:
 - ❖ не сломать старую функциональность при добавлении новой
 - ❖ понять, как работает код
 - ❖ ускорить стабильную поставку нового функционала пользователям

Для чего нужно тестирование?

Зачем нужны тесты?

Любой код
меняется

Чтение / Создание =
 $[7..200+] : 1$

10:1



baythammer-klaus.medium.com
You spend much more
time reading code than
writing code.

I like code to be in order. I like code
to be easy to read and easy to
understand. And I'm willing to go the
extra mile to make my code...

Почему ПО все равно с багами?

Тут такое дело...

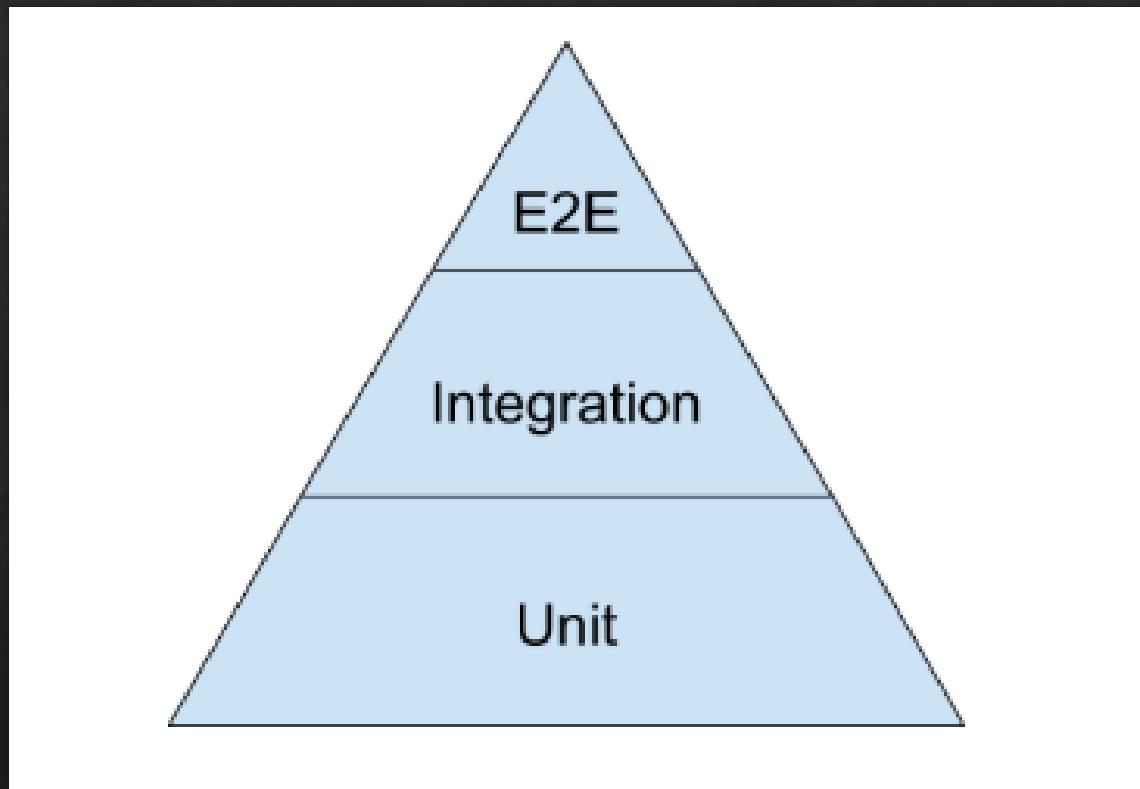
- ❖ Гарантировать что программа работает корректно практически невозможно.
- ❖ При помощи тестирования можно доказать, что программа работает неправильно, однако, доказать что она работает правильно - увы - нет

Какие виды тестирования бывают?

Ручное vs Автоматическое

- ❖ Ручное тестирование - производится тестировщиком без использования программных средств, для проверки программы или сайта путем моделирования действий пользователя
- ❖ Автоматическое тестирование - производится с помощью программных средств автоматизации

Виды тестирования по уровню абстракции



Модульное

- ❖ Модульное (unit) тестирование – это проверка корректности работы отдельных модулей программы
- ❖ Тестируется корректность работы *всех** нетривиальных методов

Хранение
данных

Бизнес
логика

UI

Интеграционное

- ❖ Это тестирование корректности работы двух или более модулей в связке друг с другом – их интеграции

Хранение
данных

Бизнес
логика

Бизнес
логика

UI

Сервис
заказов

Сервис
отчетов

End to End

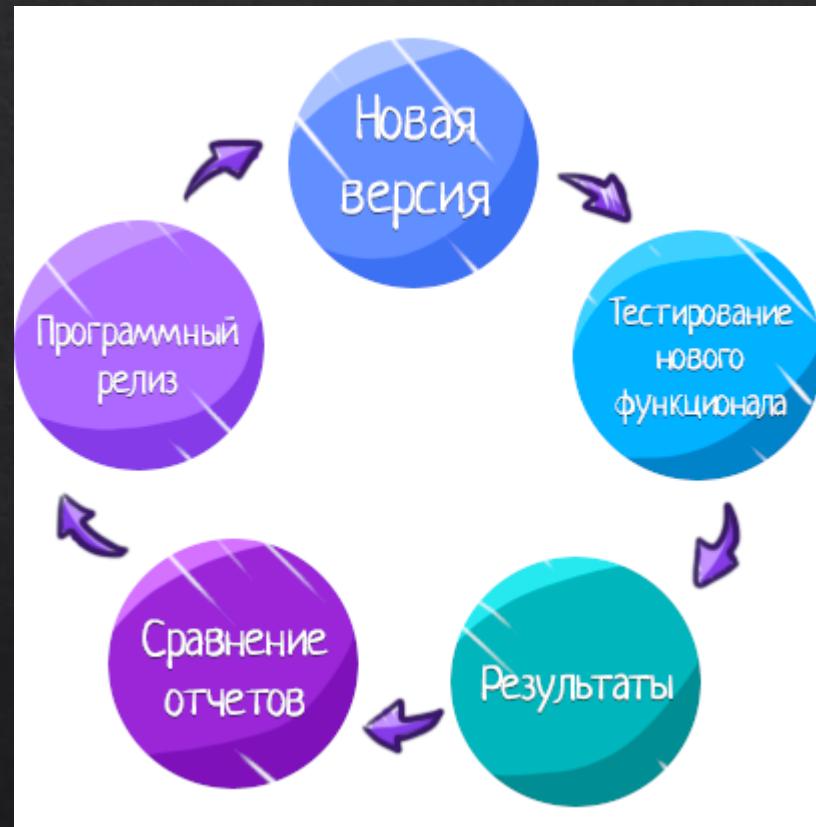
- ❖ Тестирование работы системы целиком в конкретном пользовательском сценарии – от начала и до конца.



Еще несколько видов
тестирования...

Регрессионное

- ❖ Тесты, проверяющие, что после внесения изменений старый функционал продолжает работать как ожидается
- ❖ Модульные, функциональные и интеграционные тесты могут выступать в роли регрессионного, если запущены после



Нагрузочное

- ❖ Применяется в основном для тестирования высоконагруженного ПО
- ❖ Тестируется работа ПО под большой нагрузкой (например – количество HTTP запросов к серверу)

Классы эквивалентности

- ❖ Класс эквивалентности — подмножество из множества допустимых значений, для которых поведение системы считается одинаковым
- ❖ *Сколько тестов нужно написать для каждого класса?*

Классы эквивалентности

- ❖ Для каждого класса эквивалентности достаточно написать 1 тест, также необходимы тесты на границах классов эквивалентности
- ❖ Пример: Хотим выдавать скидку 10% на товары, если их 5 и более



Какие классы эквивалентности будут для
кода решающего квадратное уравнение?

Классы эквивалентности

$$ax^2 + bx + c = 0$$

$$x = \frac{-b \pm \sqrt{D}}{2a}, \text{ где}$$
$$D = b^2 - 4ac$$

Пример 1

$D > 0$ – два корня

$$x^2 - 5x + 6 = 0$$

$$x = \frac{5 \pm \sqrt{5^2 - 4 \cdot 1 \cdot 6}}{2 \cdot 1}$$

$$x = 3 \text{ или } x = 2$$

Пример 2

$D < 0$ – корней нет

$$x^2 - 5x + 7 = 0$$

$$D = 5^2 - 4 \cdot 7 < 0$$

корней нет

Пример 3

$D = 0$ – один корень

$$x^2 - 10x + 25 = 0$$

$$x = \frac{10 \pm \sqrt{100 - 4 \cdot 25}}{2}$$

$$x = 5$$

Решение квадратного уравнения $Ax^2 + Bx + C = 0$

Подробнее о Unit тестировании

Какие задачи решает?

- ❖ Определять некорректное поведение модуля и исправлять его
- ❖ Помогает не сломать старую функциональность при добавлении новой
- ❖ Помогает понять, как работает код (документирование)
- ❖ Помогает ускорить стабильную поставку нового функционала пользователям

Какие задачи решает?

ms

Быстро получать обратную связь о том, что код соответствует ожиданиям

разработчика

пользователя

Mock, STUB, FAKE



ARRANGE - ПОДГОТОВКА
ДАННЫХ



ACT - ДЕЙСТВИЕ (КОТОРОЕ
ТЕСТИРУЕМ)



ASSERT –
КОРРЕКТНОСТЬ ПРОВЕРКИ
РАБОТЫ

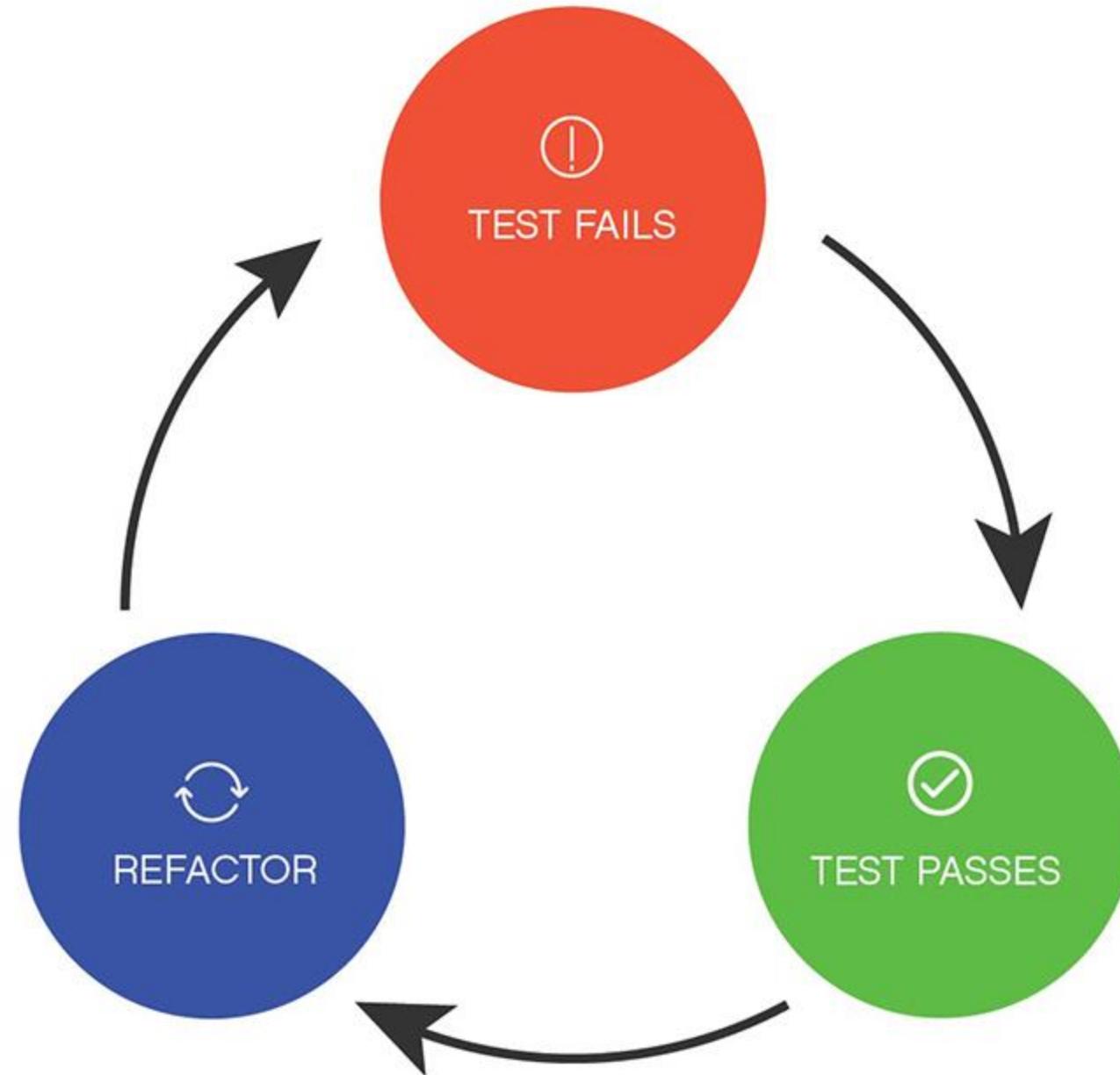
Mock, STUB, FAKE

- ❖ Используются, чтобы подменить зависимости при тестировании
- ❖ Stub (заглушка) - реализация объекта для теста – возвращающая фиксированные, известные в teste результаты
- ❖ Mock – настраиваемая реализация интерфейса (настраиваются входные, выходные параметры, возможно настроить проверки – вызывался ли метод и т.п.)
- ❖ Fake – «легкая»/простая реализация аналогичная используемой в реальной программе

Что такое хороший UNIT тест?

- ❖ Имеет понятное название
- ❖ Простой
- ❖ Быстро работает
- ❖ Проверяет один класс эквивалентности/граничное значение
- ❖ Если падает – по нему очевидно где проблема
- ❖ Изолирован т.е. не зависит от окружения и других тестов, не влияет на другие тесты
- ❖ Тестирует намерения а не реализацию

TDD Cycle



Тесты на C#

Тестовые фреймворки

- ❖ nUnit, xUnit, msTest
- ❖ Каждый тест запускается отдельно
- ❖ Для создания моков, зачастую, используется библиотека moq

Пример

Пойдем смотреть в коде