

# Реализация ядра системы. Тесты.

**№ урока:** 3 **Курс:** Python Practice

**Средства обучения:** Интерпретатор Python, virtualenv, текстовый редактор

## Обзор, цель и назначение урока

Основная цель урока: начать реализацию проекта, написать первый модуль, который будет сохранять курсы в БД. Создать базу данных для проекта. Знакомство с библиотекой реееее, создание модуля models.py. Ознакомить слушателя с особенностями тестирования в интерпретируемых языках. Познакомить студентов с библиотекой unittest, написать и запустить первые тесты.

## Изучив материал данного занятия, учащийся сможет:

- Создавать Sqlite базу данных
- Описывать модели для работы с библиотекой реееее, создавать таблицы
- С помощью библиотеки реееее извлекать и добавлять данные в БД
- Составлять, запускать unit-тесты для написанных модулей

## Содержание урока

1. База данных Sqlite
2. Библиотека реееее для доступа к базе
3. Класс SqliteDatabase, создание объекта и файла базы данных
4. Класс Model и создание собственных моделей.
5. Обзор основных реееее классов для описания полей таблиц: IntegerField, BooleanField, DoubleField, CharField, DateTimeField.
6. Создание модели XRate в модуле и таблицы xrates в базе данных
7. Создание новой записи в таблице xrates с помощью реееее
8. Получение данных из таблицы xrates с помощью реееее.
9. Создание тестового модуля для управления курсами
10. Подключение библиотеки unittest. Класс TestCase, методы setUp, tearDown(), правила именования методов. Основные методы проверки assert
11. Создание первого unit-теста. Запуск его
12. Особенности тестирования в python

## Резюме

1. Sqlite база данных - это простая база данных, написанная на C. Она не требует сервера БД, данные могут храниться в файле на диске или в памяти системы. Работа с ней доступна с версии 2.5 - не требует никакой дополнительной установки для начала работы. Встроенный модуль sqlite3 позволяет быстро начать работу с базой.

2. Peewee - простая, но мощная ORM библиотека, поддерживает работу с основными СУБД - Sqlite, MySQL, Postgres.
3. Для создание объекта базы данных Sqlite с использованием библиотеки peewee, необходимо создать экземпляр класса SqliteDatabase, указав в качестве параметра путь к файлу на диске или ключевое слово: **memory:**. Если файл на диске не существует, он будет создан, как только будут произведены первые действия с python объектом базы данных - например, создана таблица.
4. Для создания моделей необходимо создать новый класс, унаследованный от класса Model, указать название таблицы, а также объект базы данных, к которой описываемая таблица относится. Также необходимо указать поля, которые соответствуют столбцам таблицы, в зависимости от типа.
5. В описании модели также могут быть указаны индексы, значения полей по умолчанию и тп.
6. Полученный класс XRate для описания таблицы курсов.
7. Для создания записи в таблице с помощью peewee необходимо создать объект класса модели и вызвать у него метод save(). Или воспользоваться методом класса create, который доступен у каждого наследника класса Model. Проверим это в python консоли.
8. Получение данных из базы с помощью метода select(). Добавление условий where(). Работа в консоли.
9. Создание первого модуля для проекта, test\_api.py. Этот модуль пока что не будет осуществлять получение информации о курсах по api, но будет просто получать данные о курсах из базы и пере сохранять их с обновленным updated и значением курса, увеличенным на 0.01. Это поможет нам проверить, что работа с БД налажена корректно. Итак, напишем этот модуль, в нем опишем функцию update\_xrates(from, to). Параметры from, to - это валюты курса из таблицы xrates.
10. Библиотека unittest - фреймворк для unit тестирования. Unit тестирование - это метод тестирования, который позволяет в автоматическом режиме протестировать поведение модулей системы в определенных условиях. Каждый юнит тест проверяет определенное ожидаемое поведение системы в тех или иных условиях, логику работы отдельных функций, классов, модулей.
11. Для создания тестов, необходимо создать класс, унаследованный от класса TestCase. А у этого класса определить методы, название которых начинается с **test\_**. Эти методы и будут являться тест-кейсами.
12. Также у класса могут быть описаны методы setUp, tearDown(), которые будут выполняться перед каждым тестом и после каждого теста, соответственно. Иногда удобно производить инициализацию данных до запуска или удаление после.
13. Создание теста для проверки работы модуля test\_api.py. Методы assert у класса TestCase: assertEquals, assertNotEqual, assertIs, assertIsNot, assertIsNone, assertIsNotNone, assertIn, assertNotIn, assertRaises. Запуск теста в консоли.
14. Особенности тестирования в python. Так как python - интерпретируемый язык, то в отличие от компилируемых языков, в которых исходный код (каждая строка!) анализируется компилятором, а затем компилируется в байт-код, так вот в python, некоторые строки кода могут быть и не запущены никогда (например, редкое условие if или функция, которая описана, но не вызывается) или запущены в очень редких случаях. А в них может скрываться ошибка. В python тесты используются еще и для того, чтоб покрыть исходный код, выполнить все if и все функции и методы. Например, Exception() вместо raise Exception() - это бывает чаще, чем вы думаете - и в результате ошибка появляется совсем не там и не так, как ты ожидаешь.

## Закрепление материала

- Особенности БД Sqlite, работа с ней в python

- Как создать файл базы данных Sqlite на диске? (с помощью python, peewee)
- Как создать таблицу в базе данных (с помощью python, peewee)
- Как создать запись в таблице - укажите два способа. Когда удобнее применять каждый из них?
- Для чего нужно юнит тестирование? Каковы особенности тестирования в python?
- Какие методы проверки существуют в классе TestCase (назовите минимум 4)?

### Дополнительное задание

#### Задание 1

Добавить данных в таблицу xrates, попробовать добавить данные по курсу (from, to), который уже есть в базе. Проверить, что произойдет. Какие еще исключения peewee сможем сгенерировать.

#### Задание 2

Дополнить тест другими проверками assert

#### Задание 3

Напишите с помощью peewee запрос для получения курсов по валюте конвертации(from) доллар (840), и время обновления которых больше сегодня 12 дня.

### Самостоятельная деятельность учащегося

#### Задание 1

Создать базу данных в памяти. Определить, в какой момент происходит очистка БД

#### Задание 2

Описать модель для хранения кодов валют. Изменить описание модели XRate с добавлением поля ForeignKeyField

#### Задание 3

Написать тест, который ломает выполнение модуля test\_api. Подправить модуль, добавить проверку и явное исключение, добавить тест с assertRaises

### Рекомендуемые ресурсы

<https://python-scripts.com/sqlite>

<https://docs.python.org/2/library/sqlite3.html>

<http://docs.peewee-orm.com/en/latest/peewee/database.html>

<http://docs.peewee-orm.com/en/latest/peewee/querying.html>

<http://python-lab.ru/documentation/27/stdlib/unittest.html#module-unittest>