**Unittest модуль**

Модуль [Unittest](https://docs.python.org/2/library/unittest.html" \t "_self) поставляется со стандартной библиотекой Python. Она предоставляет собой класс под названием TestCase, из которого можно вызвать ваш класс. Затем можно переопределить метод setUp() чтобы подготовить среду до начала тестирования и/или метод класса classSetUp() чтобы подготовить среду для всех тестов (не очищающуюся между разными тестами). Существуют соответствующие методы tearDown() и classTearDown(), которые также можно переопределить.

Ниже приведены соответствующие разделы из нашего класса SelfDrivingCarTest. Я использую только метод setUp(). Я создаю новый экземпляр SelfDrivingCar и сохраняю его в self.car, поэтому он доступен для каждого теста.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | from unittest import TestCase        class SelfDrivingCarTest(TestCase):        def setUp(self):            self.car = SelfDrivingCar() |

Следующий шаг — написать специфические методы теста для тестирования кода внутри теста — в этом случае класс SelfDrivingCar — делает то, что он должен делать. Структура тестового метода довольно обычная:

* Подготовка среды (необязательно).
* Подготовьте ожидаемый результат.
* Вызовите код теста.
* Убедитесь, что фактический результат совпадает с ожидаемым результатом.

Обратите внимание, что результат не должен быть результатом метода. Он может быть изменением состояния класса, сторонним эффектом, например, как добавление новой строки в базе данных, записью файла или отправкой сообщения по электронной почте.

Например метод stop() класса SelfDrivingCar не возвращает ничего, но он меняет внутреннее состояние, устанавливая скорость на 0. Метод assertEqual(), предоставляемый базовым классом TestCaseиспользуется здесь для проверки, того что вызов stop() работает, как и требуется.

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11  12  13  14  15  16  17 | def test\_stop(self):            self.car.speed = 5            self.car.stop()            # Verify the speed is 0 after stopping            self.assertEqual(0, self.car.speed)                # Verify it is Ok to stop again if the car is already stopped            self.car.stop()            self.assertEqual(0, self.car.speed) |

Здесь на самом деле два теста. Первый тест, чтобы убедиться, что если скорость автомобиля равна 5 и stop() вызывается, то скорость становится равна 0. И еще один тест, чтобы убедиться, что ничего не случится, если вызвать stop() снова, когда автомобиль уже остановился.