КУРСОВАЯ РАБОТА

по теме

Телеграм бот “Агрегатор цен на wildberries”

по дисциплине «Технологии разработки качественного программного обеспечения»

|  |  |
| --- | --- |
| Выполнили  студенты гр. 5130940/00105 | Квасков С.С.  Ганенко Л.Н.  Степанов Г.Ж. |
| Преподаватель | Маслаков А.П. |

**1. Описание выполненной работы, использованных инструментах,**

**применённых техниках тест-дизайна.**

Для проведения unit-тестирования была использована библиотека pytest в связке c библиотекой unittest языка программирования python, которые предоставляют такую функциональность как удобный рекурсивный запуск тестовых сценариев, создание и настройка моков, подсчёт покрытия unit-тестами проекта, и так далее.

Запуск тестов был налажен и автоматизирован при помощи библиотеки python doit, благодаря чему запускать unit-тесты на весь проект можно одной консольной командой

**Классы эквивалентности**

Смысл этого подхода заключается в выборе значений, представляющих различные классы тестовых данных, чтобы мы могли проверить требования к продукту

пример:

./src/provider/database/unit\_test.py/304 (тесты про list)

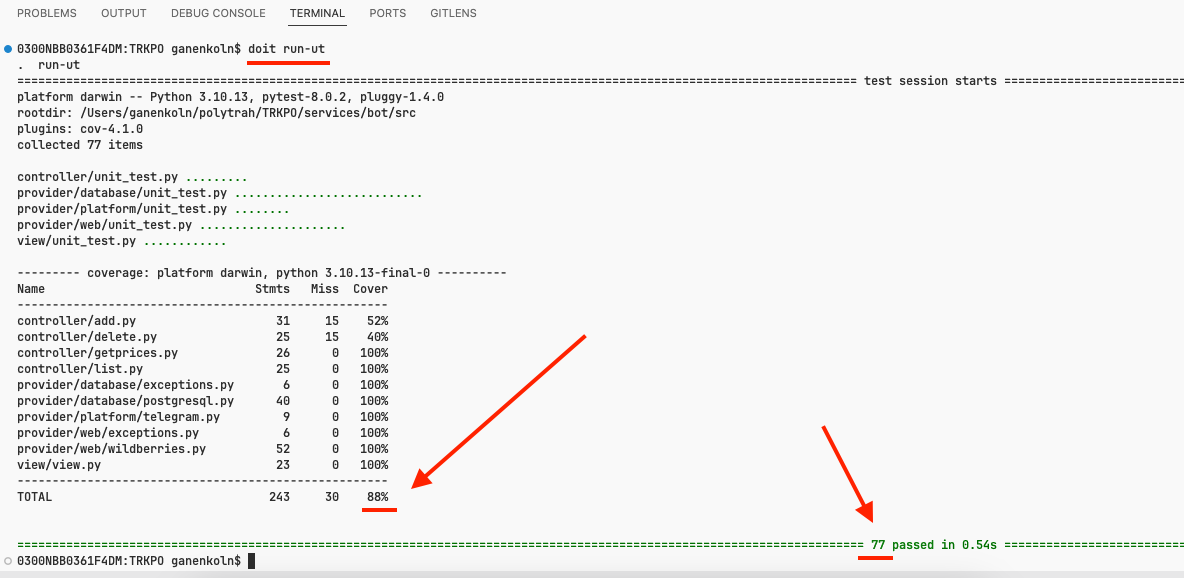
**Граничные условия**

Это техника проверки поведения продукта на крайних (граничных) значениях входных данных

пример:

./src/controller/unit\_test.py/227 (тесты про getprices)

**2. Отчёт о прохождении тестов с результатами и оценкой покрытия кода тестами.**



**3. Описание процедуры расширения тестового набора на примере**

**добавления нового блока кода, алгоритма, метода.**

Для добавления тестов на вновь написанный код необходимо: создать unit\_test.py файл в новой директории, в этом файле объявить стандартную структуру unit-тестов для pytest/unittest, написать в корне проекта команду `doit run-ut`, которая автоматически подхватит вновь созданный тестовый файл. В случае, если новый код добавляется в директорию, в которой уже присутствует unit\_test.py файл, то необходимо просто по примеру добавить новые тест-кейсы.