|  |  |
| --- | --- |
| lu135925on3bu_tmp_3360867a00ce4d37 | **Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования** **«Московский государственный технический университет имени Н.Э. Баумана** **(национальный исследовательский университет)» (МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ Информатика и системы управления и искусственный интеллект

КАФЕДРА                  Системы обработки информации и управления

**Лабораторная работа №7**

### По курсу

### «Технологии разработки программного обеспечения»

**«Тестирование и оценка качества программного проекта»**

Подготовил:

Студент группы

**ИУ5-14Б Журавлев Н.В**

05.12.2023

Проверила:

**Виноградова М.В.**

*2023 г*.

**Цель работы:**

* Изучить методы подготовки и проведения тестирования.
* Получить навыки создания и выполнения тестов для приложений и их компонентов.

**Полученное задание:**

1. (базовое) Открыть исходный код тестируемого приложения (собственное или выданный преподавателем). Добавить Unit-тест для одной из функций. Запустить тест и просмотреть результаты. Создать несколько разных тестов для проверки значений и перехвата исключений.
2. (базовое) Установить параметры сбора статистики покрытия кода. Повторить модульные тесты и просмотреть данные о покрытии кода.
3. (расширенное) Создать тестовый проект по веб-тестам производительности (для своего сайта или любого стандартного). При этом записать сценарий работы с сайтом. Настроить параметры нагрузки (частота запросов и т.д.). Выполнить тест и просмотреть результаты.
4. (расширенное) Для тестируемого приложения (собственное или выданный преподавателем) провести профайлинг (оценку производительности). Выполнить тест и просмотреть результаты.

**Ход работы:**

1. **Unit-тестирование**

Для тестирования была выбрана библиотека inittest. В качестве тестируемого кода был выбран код, сделанные в ДЗ2.

Была создана папка tests, где хранились 5 файлов с тестами, представленные на рисунке 1.

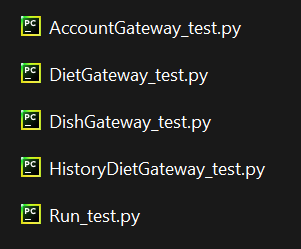


Рисунок . Файлы тестирования

Пример их содержания представлен на рисунке 2.

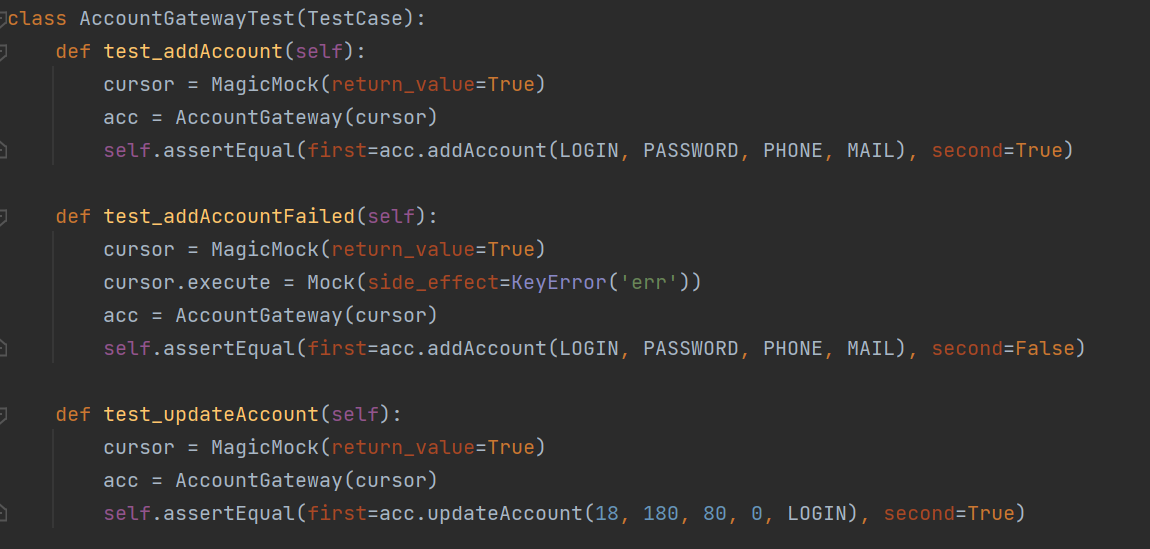


Рисунок . Пример файла тестирования

Запуск тестирования производится командой coverage run -m unittest main.py, tests\Run\_test.py,tests\AccountGateway\_test.py,tests\DietGateway\_test.py,tests\DishGateway\_test.py,tests\HistoryDietGateway\_test.py.

1. **Покрытие кода**

Для отображения процента покрытия кода используется команда python -m coverage report. Результат тестирования представлен на рисунке 3.

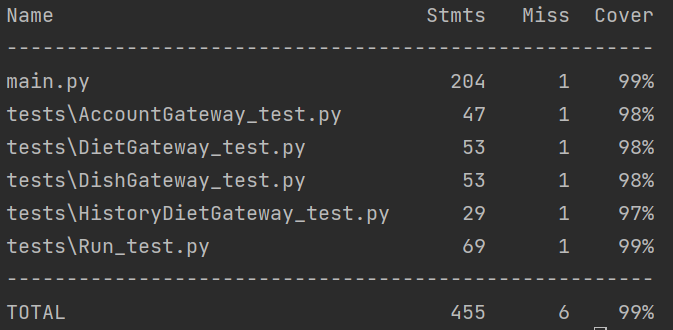


Рисунок . Статистика покрытия кода тестами

**Выводы:**

В ходе выполнения лабораторной работы были приобрести умения создания unit-тестирования и вывода статистики.

**Список источников**

1. Sparx Systems – Текст. Изображение.: электронные // Sparx Systems : [сайт]. – URL: https :// sparxsystems . com / (дата обращения 15.06.2022)
2. Виноградова, М. В. Унифицированный процесс разработки программного обеспечения: учебное пособие / М. В. Виноградова, В. И. Белоусова. — Москва: Издательство МГТУ им. Н.Э. Баумана, 2015. — 80, [2] с.: ил. ISBN 978-5-7038-4265-2
3. Якобсон А., Дуч Г., Рамбо Дж. Унифицированный процесс разработки программного обеспечения. / А. Якобсон, Г. Дуч, Дж. Рамбо. – Спб.: Питер. – 2002.
4. Арлоу Д., Нейштадт И. UML 2 и Унифицированный процесс. Практический объектно-ориентированный анализ и проектирование, 2-е издание. – Пер. с англ. – СПб: СимволПлюс, 2007. – 624 с., ил. ISBN13: 9785932860946 ISBN10: 5932860944
5. Руководство пользователя Enterprise Architect 15.1 – Текст. Изображение.:электронные //SparxSystems https :// sparxsystems . com / enterprise \_ architect \_ user \_ guide /15.1/ index / index . html обращения 15.06.2022)
6. Технологии разработки программного обеспечения: Учебник/ С. Орлов. — СПб.:Питер, 2002. — 464 с.: ил. ISBN 5-94723-145-Х