*Федеральное государственное бюджетное образовательное учреждение*

*высшего профессионального образования*

****

***«Московский государственный технический университет   
имени Н.Э. Баумана»***

***(МГТУ им. Н.Э. Баумана)***

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Программное обеспечение ЭВМ и информационные технологии

Отчет по лабораторной работе №2

по курсу «Тестирование и отладка ПО»

Тема: Модульное, интеграционное и системное тестирование клиент-серверного веб-приложения, реализующего алгоритм А\*

|  |  |
| --- | --- |
| Студенты: | Сальников Сергей |
| Ионов Алексей |
| Митягин Никита |
| Группа: | ИУ7-72 |
| Преподаватель: | Рогозин Олег Викторович |

Москва, 2015

# Цель работы

Реализовать клиент-серверное веб-приложение с использованием паттерна проектирования MVC, обеспечивающее хранение графа и нахождение пути в графе алгоритмом A\*. Написать модульные, системные и интеграционные тесты для приложения.

# Теоретическая часть

Серег, вставь что-нибудь про А\*

# Используемые технологии

Серверная часть приложения написана на python 3.5 с использованием фреймворка Django. Данный выбор был сделан в целях изучения языка python и фреймворка Django.

Клиентская часть реализована с использованием HTML (Django Templates), CSS, Javascript.

Для модульных и интеграционных тестов используется модуль unittest языка python.

Для системных тестов используется фреймворк Selenium и веб-драйвер браузера Chrome.

# Модульные тесты

Каждый класс программы проектировался на основе заранее написанных тестовых методов. В процессе написания кода классов тесты корректировались.

|  |  |
| --- | --- |
| Класс эквивалентности | Тестовый метод |
| class Graph | |
| Метод \_\_init\_\_(self, nodes=None, arcs=None) | |
| nodes=None, arcs=None | testCreationWithoutNodesAndArcs |
| nodes!=None, arcs=None | testCreationWithNodesAndWithoutArcs |
| nodes=None, arcs!=None | testCreationWithoutNodesAndWithArcs |
| nodes!=None и nodes не является типом set | testCreationWithWrongTypeOfNodesParameter |
| arcs!=None и arcs не является типом set | testCreationWithWrongTypeOfArcsParameter |
| nodes и arcs – пустые set’ы | testCreationWithEmptyNodesAndEmptyArcsParameter |
| nodes не пустое set и arcs пустое set | testCreationWithNotEmptyNodesAndEmptyArcsParameter |
| nodes пустое set и arcs не пустое set | testCreationWithEmptyNodesAndNotEmptyArcsParameter |
| nodes и arcs - не пустые set’ы | testCreationWithNotEmptyNodesAndNotEmptyArcsParameter |
| nodes и arcs не консистентные set’ы | testCreationWithNotConsistentNodesAndArcsSets |
| Метод addnode(self, node) | |
| node = None | testAddNodeWithNodeEqualsNone |
| node != None и node не является узлом | testAddNodeWithWrongTypeOfNodeParameter |
| node уже есть в графе | testAdditionOfExistingNode |
| node отсутствует в графе | testAdditionOfNewNode |
| Метод addarc(self, arc) | |
| arc = None | testAddArcWithArcEqualsNone |
| arc != None и arc не является объектом Arc | testAddArcWithWrongTypeOfArcParameter |
| arc уже есть в графе | testAdditionOfExistingArc |
| arc ссылается на несуществующую(е) вершину(ы) | testAdditionOfArcWithNotExistingNodes |
| arc ссылается на существующие вершины | testAdditionOfArcWithExistingNodes |
| Метод delarc(self, arc) | |
| arc = None | testDeleteArcWithArcEqualsNone |
| arc != None и arc не является экземп. кл. Arc | testDeleteArcWithWrongTypeOfArcParameter |
| arc отсутствует в графе | testDeletionOfNotExistingArc |
| arc присутствует в графе | testDeletionOfExistingArc |
| Метод delnode(self, arc) | |
| node = None | testDeleteNodeWithNodeEqualsNone |
| node != None и arc не явл. экземп. кл. Node | testDeleteNodeWithWrongTypeOfNodeParameter |
| node отсутствует в графе | testDeletionOfNotExistingNode |
| node присутствует в графе и не имеет дуги | testDeletionOfExistingNodeWithoutArcs |
| node присутствует в графе и имеет дуги | testDeletionOfExistingNodeWithArcs |
| Метод getarcs(self, node) | |
| node = None | testGetArcsWithNodeEqualsNone |
| node != None и node не является узлом | testGetArcsWithWrongTypeOfNodeParameter |
| node отсутствует в графе | testGetArcsByNotExistingNode |
| node присутствует в графе | testGetArcsOfExistingNode |

# Интеграционные тесты

Серег, вставь что-нибудь тоже ☺

# Системные тесты

Для системного тестирования используется фреймворк Selenium (реализация для python) и веб-драйвер браузера Chrome.

Тестируется сценарий создания графа и сохранения его на сервер. Порядок действий в тесте следующий:

* Перейти на страницу <http://127.0.0.1/>
* Проверить заголовок страницы на равенство "Graph utils"
* Выбрать элемент DOM с атрибутом id=”inputGraphArea” – поле ввода нового графа
* Ввести в это поле текст “ nodes:1,2;arcs:1->2[9]”
* Выбрать элемент DOM с атрибутом id=”saveGraphButton” – кнопка сохранения введенного графа
* Послать нажатие кнопки Enter выбранному элементу кнопки

Ниже приведен код теста:

|  |
| --- |
| class GraphClientTests(unittest.TestCase):  def setUp(self):  self.driver = webdriver.Chrome()  def test\_save\_graph\_on\_server(self):  driver = self.driver  driver.get("http://127.0.0.1:8000/")  self.assertIn("Graph utils", driver.title)  graph\_input\_area = driver.find\_element\_by\_id("graphInputArea")  graph\_input\_area.send\_keys("nodes:1,2;arcs:1->2[9]")  save\_graph\_button = driver.find\_element\_by\_id("saveGraphButton")  save\_graph\_button.send\_keys(Keys.RETURN)  def tearDown(self):  self.driver.close()  if \_\_name\_\_ == "\_\_main\_\_":  unittest.main() |

# Результаты тестов

Все тесты завершаются успешно перед запуском приложения (и (или) при его работе).

# Выводы

В ходе данной работы нами был получен опыт комплексного тестирования клиент-серверного веб приложения, получены навыки работы с языком программирования python 3.5 и фреймворком Django.

Был реализован и успешно протестирован алгоритм А\*.