

Московский государственный технический
университет им. Н.Э. Баумана.

Факультет «Системы обработки информации и управления»

Кафедра ИУ5. Курс «РИП»

Отчет по лабораторной работе №4
«Шаблоны проектирования и модульное тестирование в Python»

Выполнил:

студент группы ИУ5-53
Шамаилов М. Т.

Подпись и дата: 28.12.2020

Проверил:

преподаватель каф. ИУ5
Гапанюк Ю. Е.

Подпись и дата:

Москва, 2020 г.

Оглавление

1. Цель лабораторной работы	3
2. Задание.....	3
3. Исходный код программы	4
3.1. Файл <i>adapter.py</i>	4
3.2. Файл <i>observer.py</i>	5
3.3. Файл <i>test_observers.py</i>	7
4. Результаты работы программы	8
4.1. Файл <i>adapter.py</i>	8
4.2. Файл <i>observer.py</i>	8
4.3. Файл <i>test-observers.py</i>	8
5. Github — репозиторий	8

1. Цель лабораторной работы

Изучение реализации шаблонов проектирования и возможностей модульного тестирования в языке Python.

2. Задание

1. Необходимо для произвольной предметной области реализовать три шаблона проектирования: один порождающий, один структурный и один поведенческий. В качестве справочника шаблонов можно использовать [следующий каталог](#).
2. Для каждой реализации шаблона необходимо написать модульный тест. В модульных тестах необходимо применить следующие технологии:
 - TDD - фреймворк.
 - BDD - фреймворк.
 - Создание Mock-объектов.

3. Исходный код программы

3.1. Файл *adapter.py*

Структурный паттерн Adapter

```
class Old:
    def get(self):
        return "1234"

class New:
    def get_integer_type(self):
        return 456

class Adapter(New):
    """
    Создал сам паттерн
    (в качестве первого взял простейший адаптер)
    """

    def get(self):
        return str(self.get_integer_type())

def main(obj):
    print("Результат: " + obj.get())

if __name__ == '__main__':
    obj = Adapter()
    main(obj)
```

3.2. Файл *observer.py*

```
# Поведенческий паттерн Observer
# Реализация на примере паблика в социальной сети

from abc import ABC, abstractmethod

class Observer(ABC):
    # *** интерфейс подписчиков

    @abstractmethod
    def update(self, message: str) -> None:

        pass

class Memodel(ABC):

    """
    Тут будут авторы публикаций в паблике
    """

    def __init__(self) -> None:
        self.observers = []

    def subscribe_new_follower(self, observer: Observer) -> None:

        print (observer.name + " присоединился к нашей команде!")
        self.observers.append(observer)

    def unsubscribe(self, observer):
        print (observer.name + " покинул нас... =(")
        self.observers.remove(observer)

    def talk_and_send_message_to_other_followers(self, message: str) -> None:

        for observer in self.observers:
            print(observer.update(message))

class Publication(Memodel):

    """
    Тут будет какое-то конкретное обсуждение публикации/записи
    """

    def add_news(self, posts: str) -> None:

        self.talk_and_send_message_to_other_followers(posts)
```

```

class Follower(Observer):

    """
    А тут сидит подписчик, следящий за обновлением группы
    """

    def __init__(self, name: str) -> None:
        self.name = name

    def update(self, message: str):

        return f'{self.name} получил уведомление: {message}'

# главная функция в лабе
def main():
    post = Publication()

    is_follower_1=Follower('sky_active')
    is_follower_2=Follower('markant')

    post.subscribe_new_follower(is_follower_1)
    post.subscribe_new_follower(is_follower_2)

    post.add_news('Наблюдатель - поведенческий шаблон проектирования')
    post.unsubscribe(is_follower_1)

if __name__ == '__main__':
    main()

```

3.3. Файл test_observers.py

```
# реализуем технологию тестирования tdd на примере паттерна Observer

# ! сначала импортнём все, что нужно
import unittest
from observer import Follower
from observer import Memodel

class ObserverTestCase(unittest.TestCase):

    """
    Главный тестовый класс
    """

    # этот небольшой минитест должен проверять добавился ли новый подписчик к нам
    # в паблик
    def subscribe_new_follower_minitest(self):
        follower_subscribed_nickname = Follower("Петя")
        post = Memodel()

        post.subscribe_new_follower(follower_subscribed_nickname)

        self.assertEqual(type(follower_subscribed_nickname), type(post.observers[0]))

    # а этот небольшой минитест должен проверять отписался ли подписчик щт пабл
    # ика
    def test_unsubscribe(self):
        foll1 = Follower("Петя")
        post = Memodel()
        post.subscribe_new_follower(foll1)

        post.unsubscribe(foll1)

        self.assertEqual(0, len(post.observers))

    def test_for_users_and_followers(self):
        post = Memodel()
        foll1 = Follower("Петя")
        message = 'Наблюдатель - поведенческий шаблон проектирования'

        self.assertEqual(f'Петя получил уведомление: {message}',
                          foll1.update(message))

if __name__ == '__main__':
    unittest.main()
```

4. Результаты работы программы

4.1. Файл *adapter.py*

```
ПРОБЛЕМЫ 1 ВЫХОДНЫЕ ДАННЫЕ ТЕРМИНАЛ КОНСОЛЬ ОТЛАДКИ
PS C:\Users\Максим\Desktop\Бауманка\Бауманка, 3 курс\5 семестр\RIP\lab4> cd src
PS C:\Users\Максим\Desktop\Бауманка\Бауманка, 3 курс\5 семестр\RIP\lab4\src> python adapter.py
Результат: 456
PS C:\Users\Максим\Desktop\Бауманка\Бауманка, 3 курс\5 семестр\RIP\lab4\src> █
```

4.2. Файл *observer.py*

```
ПРОБЛЕМЫ 1 ВЫХОДНЫЕ ДАННЫЕ ТЕРМИНАЛ КОНСОЛЬ ОТЛАДКИ
PS C:\Users\Максим\Desktop\Бауманка\Бауманка, 3 курс\5 семестр\RIP\lab4> cd tests
PS C:\Users\Максим\Desktop\Бауманка\Бауманка, 3 курс\5 семестр\RIP\lab4\tests> python observer.py
sky_active присоединился к нашей команде!
markant присоединился к нашей команде!
sky_active получил уведомление: Наблюдатель - поведенческий шаблон проектирования
markant получил уведомление: Наблюдатель - поведенческий шаблон проектирования
sky_active покинул нас... =(
PS C:\Users\Максим\Desktop\Бауманка\Бауманка, 3 курс\5 семестр\RIP\lab4\tests> █
```

4.3. Файл *test-observers.py*

```
ПРОБЛЕМЫ 1 ВЫХОДНЫЕ ДАННЫЕ ТЕРМИНАЛ КОНСОЛЬ ОТЛАДКИ
PS C:\Users\Максим\Desktop\Бауманка\Бауманка, 3 курс\5 семестр\RIP\lab4> cd tests
PS C:\Users\Максим\Desktop\Бауманка\Бауманка, 3 курс\5 семестр\RIP\lab4\tests> python -m unittest test_observers
.Петя присоединился к нашей команде!
Петя покинул нас... =(
.
-----
Ran 2 tests in 0.002s

OK
PS C:\Users\Максим\Desktop\Бауманка\Бауманка, 3 курс\5 семестр\RIP\lab4\tests> █
```

5. Github — репозиторий

github.com/ShamailovMax/RIP/tree/master/lab4