

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

Факультет «Информатика и управление»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Базовые компоненты интернет-технологий»

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

Выполнил:

студент группы ИУ5-31Б
Комиссаров Михаил Вячеславович

Проверил:

преподаватель каф. ИУ5
Гапанюк Юрий Евгеньевич

Подпись и дата:

Подпись и дата:

Москва, 2021 г.

Задание:

1. Необходимо для произвольной предметной области реализовать от одного до трех шаблонов проектирования: один порождающий, один структурный и один поведенческий. В качестве справочника шаблонов можно использовать [следующий каталог](#). Для сдачи лабораторной работы в минимальном варианте достаточно реализовать один паттерн.
2. Вместо реализации паттерна Вы можете написать тесты для своей программы решения биквадратного уравнения. В этом случае, возможно, Вам потребуется доработать программу решения биквадратного уравнения, чтобы она была пригодна для модульного тестирования.
3. В модульных тестах необходимо применить следующие технологии:
 - TDD - фреймворк.
 - BDD - фреймворк.
 - Создание Mock-объектов.

```
from abc import ABC, abstractmethod

class Image(ABC):
    @abstractmethod
    def download(self) -> None:
        pass

class RealImage(Image):
    def download(self) -> None:
        print("Загрузка картинки")

class ProxyImage(Image):
    def __init__(self, real_image: RealImage) -> None:
        self._real_image = real_image

    def download(self) -> None:
        if self.check_access():
            self._real_image.download()

    def check_access(self) -> bool:
        print("Проверка запроса на загрузку перед реальной загрузкой")
        return True

def download_request(image: Image) -> None:
    image.download()

def main():
    print("Выполняем загрузку картинки сразу:")
    real_image = RealImage()
    download_request(real_image)

    print("-----")

    print("Выполняем загрузку через прокси:")
    proxy_image = ProxyImage(real_image)
```

```
download_request(proxy_image)

if __name__ == "__main__":
    main()
```

Выполняем загрузку картинки сразу:

Загрузка картинки

-----|

Выполняем загрузку через прокси:

Проверка запроса на загрузку перед реальной загрузкой

Загрузка картинки

Process finished with exit code 0