

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ  
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»  
(БГТУ им. В. Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и  
автоматизированных систем

### **Лабораторная работа № 14**

по дисциплине: Объектно-ориентированное программирование

тема: «Тестирование. Знакомство с TDD. Тесты как способ формирования  
архитектуры»

Выполнил: ст. группы ПВ-223

Игнатъев Артур Олегович

Проверил:

асс. Черников Сергей Викторович

Белгород 2024г.

## Лабораторная работа №14

### «Знакомство с библиотеками Python. PyQT»

**Цель работы:** знакомство с понятием тестирования. Получение практических навыков для написания модельных тестов.

#### Вариант 3

3.	Разработать подсистему эмулирования процесса открытия двери. Дверь можно открыть, закрыть. Дверь соединяет два помещения, соответственно, когда она закрыта попасть в другое помещение нельзя.	Теперь появляется возможность устанавливать разные замки на двери. Механический и цифровой. При добавлении цифрового замка должна быть специальная функция для сброса пароля.
----	--	---

Тесты для начального условия:

```
import pytest
from LOOP_14_1 import Door, Room, main
import sys

def test_initial_door_state():
    door = Door()
    assert not door.is_open

def test_open_door():
    door = Door()
    door.open_door()
    assert door.is_open

def test_close_door():
    door = Door(is_open=True)
    door.close_door()
    assert not door.is_open
```

Код для начальных условий задач:

```
class Door:
    def __init__(self, is_open=False):
        self.is_open = is_open

    def open_door(self):
        self.is_open = True
        print("Дверь открыта")

    def close_door(self):
        self.is_open = False
        print("Дверь закрыта")

class Room:
    def __init__(self, number):
        self.number = number

def main():
    room1 = Room(1)
    room2 = Room(2)
    door = Door()

    choice = input("Выберите действие: 1 - войти в помещение 2, 2 - остаться в этом помещении: ")

    if choice == "1":
        if not door.is_open:
            door.open_door()
            print("Пользователь вошел в помещение 2")
        else:
            print("Дверь уже открыта")
    elif choice == "2":
        print("Пользователь остается в этом помещении. Дверь закрыта")

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

## Тесты для финальных условий:

```
import pytest
from LOOP_14_2 import Door, LockType

def test_door_init_defaults():
    door = Door()
    assert not door.is_open
    assert door.lock_type == LockType.MECHANICAL
    assert door.password == "123"

def test_open_door():
    door = Door()
    door.open_door()
    assert door.is_open

def test_close_door():
    door = Door(is_open=True)
    door.close_door()
    assert not door.is_open

def test_set_lock_type_digital():
    door = Door()
    door.set_lock_type(LockType.DIGITAL)
    assert door.lock_type == LockType.DIGITAL

def test_set_lock_type_mechanical():
    door = Door(lock_type=LockType.DIGITAL)
    door.set_lock_type(LockType.MECHANICAL)
    assert door.lock_type == LockType.MECHANICAL

def test_reset_password_digital():
    door = Door(lock_type=LockType.DIGITAL)
    door.reset_password('new_pass')
    assert door.password == 'new_pass'

def test_reset_password_mechanical():
    door = Door(lock_type=LockType.MECHANICAL)
    door.reset_password('new_pass')
    assert door.password == '123'
```

Код для финальных условий:

```
from enum import Enum

class LockType(Enum):
    DIGITAL = 'digital'
    MECHANICAL = 'mechanical'

class Door:
    def __init__(self, is_open=False, lock_type=LockType.MECHANICAL, password=None):
        self.is_open = is_open
        self.lock_type = lock_type
        self.password = password if password else "123"

    def open_door(self):
        self.is_open = True
        print("Дверь открыта")

    def close_door(self):
        self.is_open = False
        print("Дверь закрыта")

    def set_lock_type(self, lock_type):
        if lock_type == LockType.DIGITAL:
            print("Цифровой замок успешно установлен")
        elif lock_type == LockType.MECHANICAL:
            print("Механический замок успешно установлен")
        self.lock_type = lock_type

    def check_password(self, entered_pass):
        if self.lock_type == LockType.DIGITAL and self.password == entered_pass:
            print("Пароль принят, проходите")
        else:
            print("Пароль не верный, дверь остается закрытой")

    def reset_password(self, new_password):
        if self.lock_type == LockType.DIGITAL:
            self.password = new_password
            print("Пароль успешно сброшен")

class Room:
    def __init__(self, number):
        self.number = number

def main():
    room1 = Room(1)
    room2 = Room(2)
    choice = input("Выберите действие: 1 - войти в соседнюю комнату, 2 - остаться в текущей комнате: ")
    door = None
```

```

    if choice == "1":
        door = Door()
        lock_type_input = input("Выберите тип замка на двери (digital/mechanical): ")
        lock_type = LockType.DIGITAL if lock_type_input == 'digital' else LockType.MECHANICAL
        door.set_lock_type(lock_type)

        if lock_type == LockType.DIGITAL:
            password = input("Введите пароль от замка: ")
            door.check_password(password)
            reset_password = input("Желаете изменить пароль от цифрового замка? y/n:")
            if reset_password.lower() == "y":
                new_pass = input("Введите новый пароль: ")
                door.reset_password(new_pass)

        if not door.is_open:
            door.open_door()
            print("Вы вошли в соседнюю комнату")

    elif choice == "2":
        door = Door(is_open=False) # Дверь закрыта, если пользователь остается в текущей комнате
        print("Дверь закрыта, вы остаетесь в текущей комнате")

if __name__ == "__main__":
    main()

```

Результат работы программы:

```

Выберите действие: 1 - войти в соседнюю комнату, 2 - остаться в текущей комнате: 2
Дверь закрыта, вы остаетесь в текущей комнате

```

```

Выберите действие: 1 - войти в соседнюю комнату, 2 - остаться в текущей комнате: 1
Выберите тип замка на двери (digital/mechanical): mechanical
Механический замок успешно установлен
Дверь открыта
Вы вошли в соседнюю комнату

```

```
Выберите действие: 1 - войти в соседнюю комнату, 2 - остаться в текущей комнате: 1
Выберите тип замка на двери (digital/mechanical): digital
Цифровой замок успешно установлен
Введите пароль от замка: 715
Пароль не верный, дверь остается закрытой
Вы хотите изменить пароль от цифрового замка? y/n:y
Введите новый пароль: 564
Пароль успешно сброшен
Дверь открыта
Вы вошли в соседнюю комнату
```

**Вывод:** на этой лабораторной работе ознакомился с понятием тестирования. Получил практические навыки для написания модельных тестов