**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Ордена Трудов ого Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский технический университет связи и информатики»**

Кафедра Программная инженерия

ОТЧЕТ

по лабораторной работе №6

по дисциплине «Введение в информационные технологии»

Тема: «Работа с классами ч.2»

Выполнил: студент группы БВТ2501

Лебедев Даниил Владимирович

Москва, 2025

**Цель работы**

Получить практический опыт работы с ООП в Python. использование инкапсуляции, наследования.

**Индивидуальное задание**

1. Защита данных пользователя

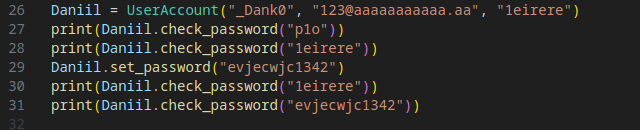
1 Создать класс UserAccount, который представляет аккаунт пользователя с атрибутами: имя пользователя (username), электронная почта (email) и приватный атрибут пароль (password).  
2. Использовать конструктор \_\_init\_\_ для инициализации этих атрибутов.  
3. Реализовать метод set\_password(new\_password), который позволяет безопасно изменить пароль аккаунта.  
4. Реализовать метод check\_password(password), который проверяет, соответствует ли введённый пароль текущему паролю аккаунта и возвращает True или False.  
5. Создать объект класса UserAccount, попробовать изменить пароль и проверить его с помощью методов set\_password и check\_password.

1. Полиморфизм и наследование

1. Определить базовый класс Vehicle с атрибутами: make (марка) и model (модель), а также методом get\_info(), который возвращает информацию о транспортном средстве.  
2. Создать класс Car, наследующий от Vehicle, и добавить в него атрибут fuel\_type (тип топлива). Переопределить метод get\_info() таким образом, чтобы он включал информацию о типе топлива.

Ход работы:

1. Напишем класс UserAccount (см рис2), потом создадим объект с этим классом (рис 1) и проверим работу класса, и посмотрим на вывод (рис 3), увидим, что всё работает отлично.

Рис 1

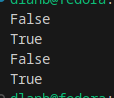
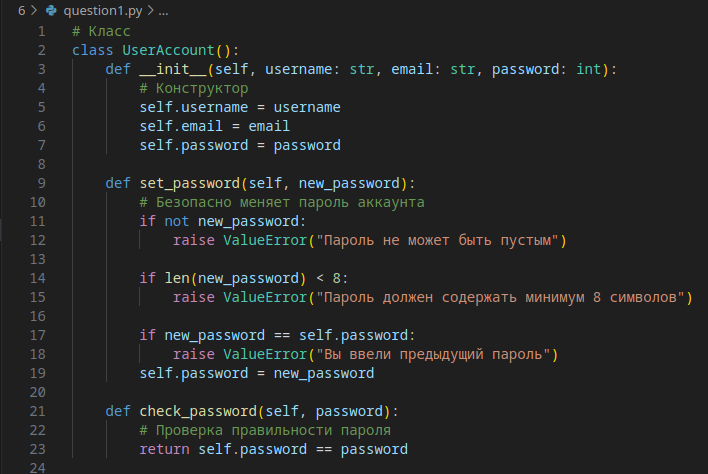
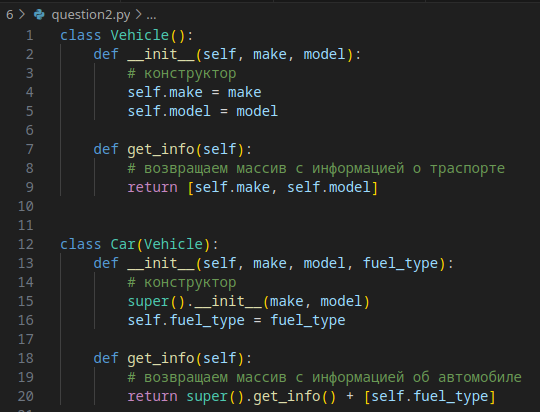


Рис 2

Рис 3

1. Напишем класс Vehicle и наследный от него Car (см рис 4), для соединения классов, используем super, которая обеспечивает доступ к оригинальным методам. Позже создадим объекты с этими классами (рис 5)

Рис 4

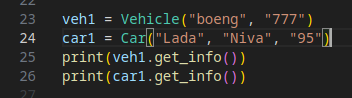


Рис 5

И проверим выводы (рис 6). Убедимся, что всё правильно вывело.

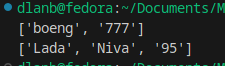


Рис 6

Заключениe:

Мы получили практический опыт работы с ООП в Python. использование инкапсуляции, наследования.

Приложение:

**github с лабами -** [**https://github.com/Bulbanator123/labsmtucivvit**](https://github.com/Bulbanator123/labsmtucivvit)