Министерство цифрового развития и массовых коммуникаций Российской Федерации

Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования «Московский технический университет связи и информатики» (МТУСИ)

Кафедра Информационных технологий

Отчет по лабораторной работе № 6

по дисциплине «Введение в информационные технологии» на тему: «Работа с классами»

Выполнил: студент группы БВТ2402 Левая Валерия Валерьевна **Цель работы**: Получить практический опыт работы с ООП в Python. использование инкапсуляции, наследования.

Ход работы:

Задание 1: Защита данных пользователя.

- 1. Создадим класс **UserAccount**, который представляет аккаунт пользователя с атрибутами: имя пользователя (**username**), электронная почта (**email**) и приватный атрибут пароль (**password**).
- 2. Используем конструктор <u>init</u> для инициализации этих атрибутов.
- 3. Peaлизуем метод **set_password**(**new_password**), который позволяет безопасно изменить пароль аккаунта.
- 4. Реализуем метод **check_password**(**password**), который проверяет, соответствует ли введённый пароль текущему паролю аккаунта и возвращает **True** или **False**.
- 5. Создадим объект класса **UserAccount**, изменим пароль и проверить его с помощью методов **set password** и **check password**.

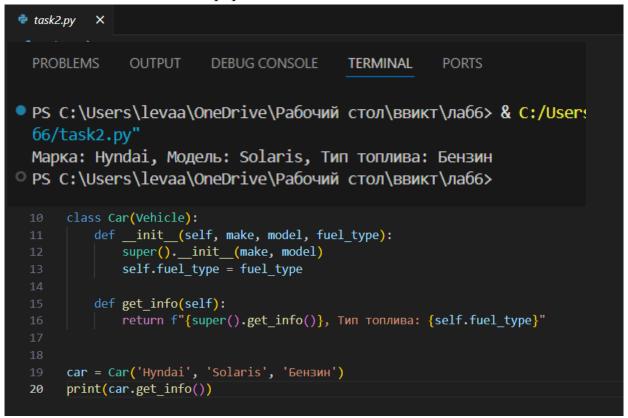
```
task1.py
task1.py > ધ UserAccount
  1 v class UserAccount:
          def init (self, username, email, password):
              self.username = username
              self.email = email
              self. password = password
          def set password(self, new password):
              self. password = new password
               print('Пароль изменен')
          def check password(self, password):
              return self.__password == password
      user = UserAccount('user123', 'user@ex.com', 'initialpass')
      user.set password('newpass')
      print(user.check password('newpass'))
      print(user.check password('wrongpass'))
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\levaa\OneDrive\Pa6oчий стол\ввикт\лаб6> & C:/Use 66/task1.py"
Пароль изменен
True
False
PS C:\Users\levaa\OneDrive\Pa6oчий стол\ввикт\лаб6>
```

Задание 2: Полиморфизм и наследование.

- 1. Определим базовый класс **Vehicle** с атрибутами: **make** (марка) и **model** (модель), а также методом **get_info**(), который возвращает информацию о транспортном средстве.
- 2. Создадим класс **Car**, наследующий от **Vehicle**, и добавим в него атрибут **fuel_type** (тип топлива). Переопределим метод **get_info**() таким образом, чтобы он включал информацию о типе топлива.



Результат:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\levaa\OneDrive\Paбoчий стол\ввикт\лаб6> & C:/Users 66/task2.py"

Марка: Hyndai, Модель: Solaris, Тип топлива: Бензин
PS C:\Users\levaa\OneDrive\Paбoчий стол\ввикт\лаб6>
```

Вывод: Я получила практический опыт работы с ООП в Python. Использовала инкапсуляции, наследования.