

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

**Ордена Трудового Красного Знамени федеральное государственное бюджетное
образовательное учреждение высшего образования**

«Московский технический университет связи и информатики»

Кафедра “Математическая кибернетика и информационные технологии”

ОТЧЕТ
по лабораторной работе №6
по дисциплине «Введение в информационные технологии»

Тема: «Работа с классами ч.2»

Выполнил: студент группы БВТ2505
Кручко Александр Вадимович

Проверил: Павликov A. E.

Москва, 2025

Цель работы

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

Индивидуальное задание

Задание 1: Защита данных пользователя

1. Создайте класс UserAccount, который представляет аккаунт пользователя с атрибутами: имя пользователя (username), электронная почта (email) и приватный атрибут пароль (password).
2. Используйте конструктор `__init__` для инициализации этих атрибутов.
3. Реализуйте метод `set_password(new_password)`, который позволяет безопасно изменить пароль аккаунта.
4. Реализуйте метод `check_password(password)`, который проверяет, соответствует ли введённый пароль текущему паролю аккаунта и возвращает True или False.
5. Создайте объект класса UserAccount, попробуйте изменить пароль и проверить его с помощью методов `set_password` и `check_password`.

Задание 2: Полиморфизм и наследование

1. Определите базовый класс Vehicle с атрибутами: make (марка) и model (модель), а также методом `get_info()`, который возвращает информацию о транспортном средстве.
2. Создайте класс Car, наследующий от Vehicle, и добавьте в него атрибут fuel_type (тип топлива). Переопределите метод `get_info()` таким образом, чтобы он включал информацию о типе топлива.

Скриншоты выполнения

```
Пароль изменен
False
Daewoo, Matiz
Ford, Mustang, бензин
```

Исходный код программы

```
#Задание 1

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
```

```
        return f'Пароль изменен'

def chek_password(self, password):
    if self.__password == password:
        return True
    else:
        return False

account = UserAccount('sanchez', '1039sasha@gmail.com', '1234')
print(account.set_password('4321'))
print(account.chek_password('1244'))

#Задание 2

class Vehicle:
    def __init__(self, make, model):
        self.make = make
        self.model = model

    def get_info(self):
        return f'{self.make}, {self.model}'

class Car(Vehicle):
    def __init__(self, make, model, fuel_type):
        super().__init__(make, model)
        self.fuel_type = fuel_type

    def get_info(self):
        return f'{self.make}, {self.model}, {self.fuel_type}'

a = Vehicle('Daewoo', 'Matiz')
car1 = Car('Ford', 'Mustang', 'бензин')
print(a.get_info())
print(car1.get_info())
```

Заключение

В ходе выполнения лабораторной работы были успешно решены следующие задачи:

1. Получен практический опыт работы с ООП в Python.
2. Изучены такие принципы как инкапсуляция и наследование