**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ**

**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

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

Кафедра «Математическая Кибернетика и Информационные технологии»

Лабораторная работа №6

Выполнил: Студент группы

БВТ2402

Мушта Александр

Москва

2024

# Лабораторная работа № 6 Работа с классами ч.2

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

## 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**.

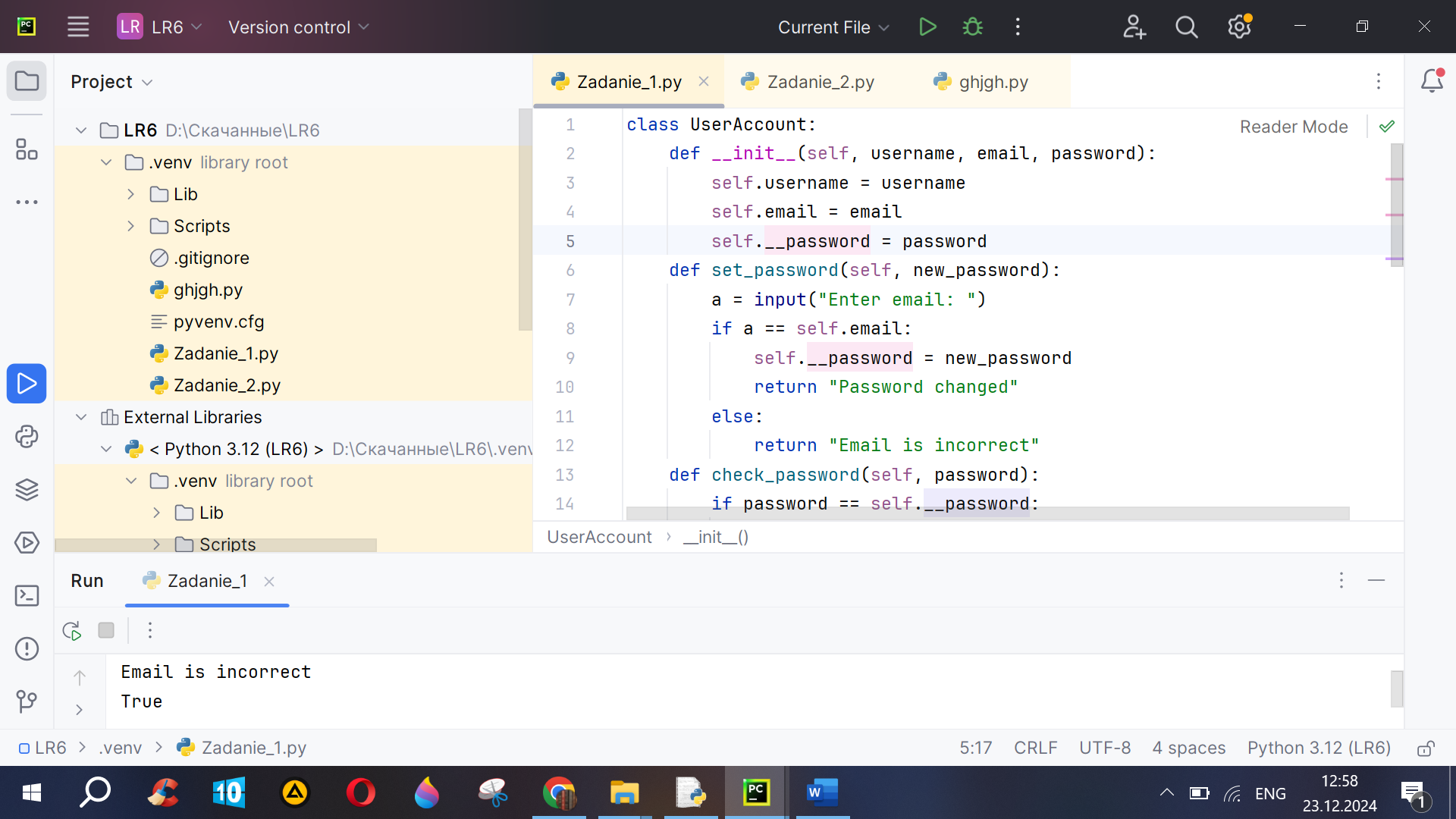
## Задание 2: Полиморфизм и наследование

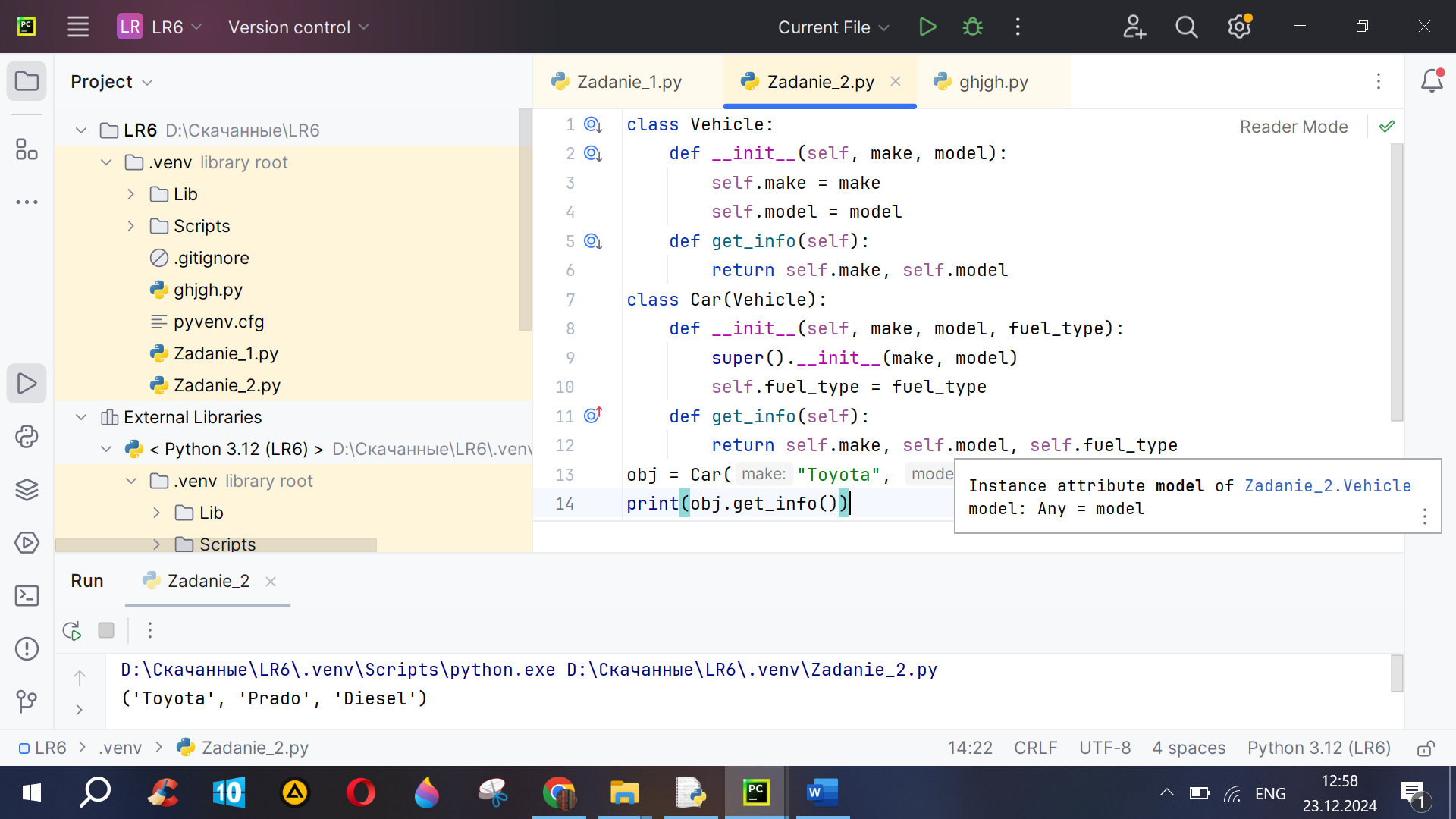
## 

1. Определите базовый класс **Vehicle** с атрибутами: **make** (марка) и **model** (модель), а также методом **get\_info**(), который возвращает информацию о транспортном средстве.

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.

|  |
| --- |
| # 1  **class** **UserAccount:**  **def** \_\_init\_\_**(**self**,** username**,** email**,** password**):**  self**.***username* **=** username  self**.***email* **=** email  self**.***\_\_password* **=** password  **def** set\_password**(**self**,** new\_password**):**  a **=** **input(**"Enter email: "**)**  **if** a **==** self**.***email***:**  self**.***\_\_password* **=** new\_password  **return** "Password changed"  **else:**  **return** "Email is incorrect"  **def** check\_password**(**self**,** password**):**  **if** password **==** self**.***\_\_password***:**  **return** **True**  **else:**  **return** **False**  obj **=** UserAccount**(**"Alex"**,** "123@ht.com"**,** "0000"**)**  **print(**obj**.***set\_password***(**"1111"**))**  **print(**obj**.***check\_password***(**"0000"**))**  # 2  **class** **Vehicle:**  **def** \_\_init\_\_**(**self**,** make**,** model**):**  self**.***make* **=** make  self**.***model* **=** model  **def** get\_info**(**self**):**  **return** 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 self.make, self.model, self.fuel\_type  obj = Car("Toyota", "Prado", "Diesel")  print(obj.get\_info()) |





Вывод: получил практический опыт работы с ООП в Python. использование инкапсуляции, наследования.