

**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

Вакалюк А.А.

Проверил: Павликов. А.Е.

Москва, 2025

Цель работы

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

Скриншоты выполнения:

```
email
Введите новый пароль: 1488
Пароль успешно изменен
Введите пароль для проверки: 1488
Пароль верный
Username: mail, Email: email Password: 1488
```

```
Базовый класс Vehicle:
Марка: Porsche 911, Модель: 911 Turbo серии 997
Марка: Toyota, Модель: Camry, Тип топлива: бензин
```

Исходный код программы:

```
1  class UserAccount:
2      def __init__(self, username, email, password):
3          self.username = username
4          self.email = email
5          self.__password = password
6
7      def set_password(self, new_password):
8          self.__password = new_password
9          return f"Пароль успешно изменен"
10
11     def check_password(self, password):
12         return password == self.__password
13
14     def get_account_info(self):
15         return f"Username: {self.username}, Email: {self.email} Password: {self.__password}"
16
17 user1 = UserAccount("mail", "email", "password")
18
19 #pw = input("Введите новый пароль: ")
20 #print(user1.set_password(pw))
21
22 user_input = input("Введите пароль для проверки: ")
23 if user1.check_password(user_input):
24     print("Пароль верный")
25 else:
26     print("Неверный пароль")
27 print(user1.get_account_info())
28
1  class Vehicle:
2      def __init__(self, make, model):
3          self.make = make
4          self.model = model
5
6      def get_info(self):
7          return f"Марка: {self.make}, Модель: {self.model}"
8
9  class Car(Vehicle):
10     def __init__(self, make, model, fuel_type):
11         super().__init__(make, model)
12         self.fuel_type = fuel_type
13
14     def get_info(self):
15         base_info = super().get_info()
16         return f"{base_info}, Тип топлива: {self.fuel_type}"
17
18
19 vehicle = Vehicle("Porsche 911", "911 Turbo серии 997")
20
21 car1 = Car("Toyota", "Camry", "бензин")
22 car2 = Car("Tesla", "Model S", "электричество")
23 car3 = Car("Volkswagen", "Golf", "дизель")
24
25 print("Базовый класс Vehicle:")
26 print(vehicle.get_info())
27 print(car1.get_info())
```

Заключение

В ходе лабораторной работы были изучены ключевые принципы объектно-ориентированного программирования в Python — инкапсуляция, наследование и полиморфизм. На примере класса `UserAccount` продемонстрировано создание приватных атрибутов и реализация методов, обеспечивающих безопасную работу с конфиденциальной информацией пользователя, включая проверку и изменение пароля.

Также рассмотрено наследование на основе базового класса `Vehicle` и его наследника `Car`. Был переопределён метод `get_info()`, что наглядно проиллюстрировало применение полиморфизма.