**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

# Работа с классами ч.2

Выполнил: Студент группы

БВТ2402

Бадма-Халгаев Артен

Москва

2024

**Цель работы:**

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

**Задание:**

1. Программа «Защита данных пользователя» - программа, содержащая класс, который представляет аккаунт пользователя с атрибутами: имя пользователя, электронная почта и приватный атрибут пароль, а также методы в этом классе, позволяющие проверить и изменить пароль.

2. Программа «Полиморфизм и наследование» - программа, содержащая два класса: второй наследуется от первого и содержит один дополнительный атрибут.

**Ход работы:**

Код программы «Защита данных пользователя»:

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  
  
 def check\_password(self, password):  
 return self.password == password  
  
  
user = UserAccount("user1", "user1@gmail.com", "qwerty123")  
user.set\_password("password")  
print(user.check\_password("qwerty123"))  
print(user.check\_password("password"))

Код программы «Полиморфизм и наследование»:

class Vehicle:  
 def \_\_init\_\_(self, make, model):  
 self.make = make  
 self.model = model  
  
 def get\_info(self):  
 return "Марка: {}, модель: {}".format(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 "Марка: {}, модель: {}, тип топлива: {}".format(self.make, self.model, self.fuel\_type)  
  
  
car1 = Vehicle("audi", "a5")  
car2 = Car("bmw", "m5", "gas")  
print(car1.get\_info())  
print(car2.get\_info())

**Вывод**: Мы получили практический опыт работы с ООП в Python. использовали инкапсуляции, наследования.