**Урок No15. ООП**

1.

# Задание No1

# Есть родительский класс:

# class Transport:

# def \_\_init\_\_(self, name, max\_speed, mileage):

# self.name = name

# self.max\_speed = max\_speed

# self.mileage = mileage

# Создайте объект Autobus, который унаследует все переменные и методы

# родительского класса Transport и выведете его.

# Ожидаемый результат вывода:

# Название автомобиля: Renault Logan Скорость: 180 Пробег: 12

#

# https://github.com/A-l-E-v/PySynergy/blob/main/U-15/bus\_name.py

#

class Transport:

# конструктор

def \_\_init\_\_(self, name, max\_speed, mileage):

self.name = name

self.max\_speed = max\_speed

self.mileage = mileage

# метод печати объекта

def print\_me(self):

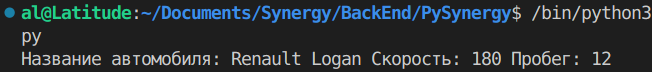
return print (f'Название автомобиля: {self.name} Скорость: {self.max\_speed} Пробег: {self.mileage}' )

# создаём объект

Autobus = Transport ('Renault Logan', 180, 12)

# вызываем метод печати

Autobus.print\_me()

****

2.

# Задание No2

# Создайте класс Autobus, который наследуется от класса Transport.

# Дайте аргументу Autobus.seating\_capacity() значение по умолчанию 50.

# Используйте переопределение метода.

# Используйте следующий код для родительского класса транспортного

# средства:

# class Transport:

# def \_\_init\_\_(self, name, max\_speed, mileage):

# self.name = name

# self.max\_speed = max\_speed

# self.mileage = mileage

# def seating\_capacity(self, capacity):

# return f"Вместимость одного автобуса {self.name} {capacity} пассажиров"

# Ожидаемый результат вывода:

# Вместимость одного автобуса Renault Logan: 50 пассажиров

#

# https://github.com/A-l-E-v/PySynergy/blob/main/U-15/bus\_capacity.py

#

# Базовый класс

class Transport:

def \_\_init\_\_(self, name, max\_speed, mileage):

self.name = name

self.max\_speed = max\_speed

self.mileage = mileage

# метод базового класса

def seating\_capacity(self, capacity):

return f"Вместимость одного автобуса {self.name} {capacity} пассажиров"

class Autobus(Transport):

# если вместимость не передана, то подставляем 50

def seating\_capacity(self, capacity = 50):

return super ().seating\_capacity (capacity)

# испытаем метод родительского класса

tr\_test = Transport('Transport test',120,23243)

print (tr\_test.seating\_capacity(59))

# используя метод дочернего класса, передадим вместимость 76

the\_bus = Autobus('The\_bus',110,34432)

print(the\_bus.seating\_capacity(76))

print()

# выполним задание - параметр вместимости не передан, значит будет 50

a\_bus = Autobus('Renault Logan',130,54837)

print(a\_bus.seating\_capacity())

