

Отчёт

Практическое занятие №17

Тема: составление программ с использованием ООП.

Цель: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ с использованием ООП.

Постановка задач:

- 1)Создайте класс «Круг», который имеет атрибут радиуса и методы для вычисления площади, длины окружности и диаметра.
- 2)Создание базового класса "Транспортное средство" и его наследование для создания классов "Автомобиль" и "Мотоцикл". В классе "Транспортное средство" будут общие свойства, такие как максимальная скорость и количество колес, а классы наследники будут иметь свои уникальные свойства и методы.

Тип алгоритма: линейный.

Текст программы:

1)# Создайте класс «Круг», который имеет атрибут радиуса и методы для вычисления
площади, длины окружности и диаметра.

```
class Krug:
    def __init__(self,radius):
        self.radius = radius

    def get_r(self):
        return self.radius

    def pl(self):
        return 3.14 * (self.radius ** 2)

    def dlina(self):
        return 2 * 3.14 * self.radius

    def diameter(self):
```

```
return 2 * self.radius
```

```
k = Krug(5)
print("Радиус:", k.get_r())
print("Площадь:", k.pl())
print("Длина:", k.dlina())
print("Диаметр:", k.diameter())
```

2) # Создание базового класса "Транспортное средство" и его наследование для создания классов "Автомобиль" и "Мотоцикл". В классе "Транспортное средство" будут общие свойства, такие как максимальная скорость и количество колес, а классы наследники будут иметь свои уникальные свойства и методы.

```
class Transp:
    def __init__(self, max_speed, wheels):
        self.max_speed = max_speed
        self.wheels = wheels

    def get_ms(self):
        return self.max_speed

    def set_ms(self, new_ms):
        self.max_speed = new_ms

    def get_wh(self):
        return self.wheels

    def set_wh(self, new_wh):
        self.wheels = new_wh

class Avto(Transp):
    def __init__(self, max_speed, wheels, brand):
        Transp.__init__(self, max_speed, wheels)
        self.brand = brand

    def get_brand(self):
        return self.brand

    def set_brand(self, new_b):
        self.brand = new_b

    def honk(self):
        print('бип бип')

class Moto(Transp):
    def __init__(self, max_speed, wheels, color):
        Transp.__init__(self, max_speed, wheels)
        self.color = color

    def get_color(self):
        return self.color

    def set_color(self, new_c):
        self.color = new_c

    def round(self):
```

```
print('влево вправо')

t = Transp(60, 4)
a = Avto(120, 4, 'мерседес')
m = Moto(50, 2, 'blue')
m.set_ms(23)
print(t.get_ms(), t.get_wh())
print(a.get_brand(), a.get_ms(), a.get_wh())
print(m.get_color(), m.get_ms(), m.get_wh())
a.honk()
m.round()
```

Протокол работы программы:

1)Радиус: 5

Площадь: 78.5

Длина: 31.400000000000002

Диаметр: 10

2)60 4

мерседес 120 4

blue 23 2

бип бип

влево вправо

Вывод: В процессе выполнения практического занятия я закрепила усвоенные знания, понятия, алгоритмы, основные принципы составления программ и приобрела навыки составления программ с использованием ООП в IDE PyCharm Community.