

## Практическая работа №17

**Тема:** составление программ с использованием ООП.

**Цель:** закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ с ООП в IDE PyCharm Community.

### Задача 1

**Постановка задачи:** создайте класс «Круг», который имеет атрибут радиуса и методы для вычисления площади, длины окружности и диаметра.

**Текст программы:**

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

import math

class circle():
    def __init__(self, radius):
        self.radius = radius

    def area(self):
        return math.pi * (self.radius ** 2)

    def perimeter(self):
        return 2 * math.pi * self.radius

    def diametr(self):
        return 2 * self.radius

r = int(input("Введите радиус круга: "))
obj = circle(r)
print("Площадь круга:", round(obj.area(), 2))
```

```
print("Длина окружности:", round(obj.perimeter(), 2))
print("Диаметр: ", round(obj.diametr(), 2))
```

### Протокол работы программы:

```
Введите радиус круга: 7
Площадь круга: 153.94
Длина окружности: 43.98
Диаметр: 14
```

## Задача 2

**Постановка задачи:** 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.

### Текст программы:

```
# 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 Transport:
    def __init__(self, max_rate):
        self.max_rate = max_rate

    def drive(self):
        print("Едем")

class Avto(Transport):
    def __init__(self, max_rate, wheels=4):
        super().__init__(max_rate)
        self.__wheels=wheels

    def open_doors(self):
        print("Двери открылись")

    def close_doors(self):
```

```
print("Двери закрылись")

class Motorcycle(Transport):
    def __init__(self, max_rate, wheels=2, helmet=1):
        super().__init__(max_rate)
        self.__wheels = wheels
        self.__helmet = helmet
        self.__with_carriage = False

    def add_carriage(self):
        self.__with_carriage = True
        print("Добавлена коляска")

    def remove_carriage(self):
        self.__with_carriage = False
        print("Убрана коляска")

car = Avto(180)
car.open_doors()
car.close_doors()
car.drive()

moto = Motorcycle(120, helmet=2)
moto.add_carriage()
```

### Протокол работы программы:

Двери открылись

Двери закрылись

Едем

Добавлена коляска

**Вывод:** в процессе выполнения практического задания я закрепила усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрел навыки составления программ с ООП в IDE PyCharm Community. Выполнены разработка кода, отладка, тестирование, оптимизация программного кода.