

## Практическое занятие №16

**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.

### Постановка задачи:

#### Блок 1

Создайте класс “Круг”, который имеет атрибут радиуса и методы для вычисления площади, длины окружности и диаметра.

#### Блок 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.

#### Блок 3

Для задачи из блока 1 создать две функции, save\_def и load\_def, которые позволяют сохранять информацию из экземпляров класса (3 шт.) в файл и загружать ее обратно. Использовать модуль pickle для сериализации и десериализации объектов Python в бинарном формате.

### Текст программы:

#### Блок 1

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

    def calculate_area(self):
        return 3.14159 * self.radius ** 2

    def calculate_circumference(self):
        return 2 * 3.14159 * self.radius

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

# Пример использования класса Circle
radius = 5
circle = Circle(radius)

print("Площадь круга:", circle.calculate_area())
print("Длина окружности:", circle.calculate_circumference())
print("Диаметр круга:", circle.calculate_diameter())
```

#### Блок 2

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

class Car(Transport):
    def __init__(self, max_speed, wheels, brand, fuel_type):
        super().__init__(max_speed, wheels)
        self.brand = brand
        self.fuel_type = fuel_type

    def drive(self):
        return f"Автомобиль {self.brand} едет."
```

```

class Motorbike(Transport):
    def __init__(self, max_speed, wheels, brand, engine_type):
        super().__init__(max_speed, wheels)
        self.brand = brand
        self.engine_type = engine_type

    def start_engine(self):
        return f"Мотоцикл {self.brand} заводится."

# Пример использования классов
car1 = Car(200, 4, "Toyota", "Бензин")
print(car1.drive())

motorbike1 = Motorbike(180, 2, "Yamaha", "Электричество")
print(motorbike1.start_engine())

```

### Блок 3

```

import pickle

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

    def calculate_area(self):
        return 3.14159 * self.radius ** 2

    def calculate_circumference(self):
        return 2 * 3.14159 * self.radius

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

def save_def(obj, filename):
    with open(filename, 'wb') as file:
        pickle.dump(obj, file)

def load_def(filename):
    with open(filename, 'rb') as file:
        obj = pickle.load(file)
    return obj

# Пример использования класса Circle
radius = 5
circle = Circle(radius)

# Сохраняем данные в файл
save_def(circle, 'circle_data.pkl')

# Загружаем данные из файла
loaded_circle = load_def('circle_data.pkl')

print("Площадь круга:", loaded_circle.calculate_area())
print("Длина окружности:", loaded_circle.calculate_circumference())
print("Диаметр круга:", loaded_circle.calculate_diameter())

```

### Протокол работы программы 1:

Площадь круга: 78.53975  
 Длина окружности: 31.4159  
 Диаметр круга: 10

Process finished with exit code 0

**Протокол работы программы 2:**

Автомобиль Toyota едет.

Мотоцикл Yamaha заводится.

Process finished with exit code 0

**Протокол работы программы 3:**

Площадь круга: 78.53975

Длина окружности: 31.4159

Диаметр круга: 10

Process finished with exit code 0

**Вывод:** В процессе выполнения практического занятия выработала навыки составления программ с использованием ООП в IDE PyCharm Community. Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на GitHub.