

## 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 №1.

Создайте класс "Машина" с атрибутами "марка", "модель" и "год выпуска".  
Напишите метод, который выводит информацию о машине в формате "Марка: марка, Модель: модель, Год выпуска: год".

**Текст программы:**

```
# Вариант 14 class Машина:
    def __init__(self, марка, модель, год_выпуска):
        self.марка = марка
        self.модель = модель
        self.год_выпуска = год_выпуска

    def __str__(self):
        return f'Марка: {self.марка}, Модель: {self.модель}, Год выпуска: {self.год_выпуска}'

машина = Машина('BMW', 'M5', 2021)
print(машина)
```

**Протокол работы программы:**

Марка: BMW, Модель: M5, Год выпуска: 2021  
Process finished with exit code 0

### Постановка задачи №2.

Создайте базовый класс "Форма" со свойствами "цвет" и "тип". От этого класса унаследуйте класс "Круг" и добавьте в него свойство "радиус". Определите методы вычисления площади и периметра.

**Текст программы:**

```
# Вариант 14 import math
class Форма:
    def __init__(self, цвет, тип):
        self.цвет = цвет
        self.тип = тип

class Круг(Форма):
    def __init__(self, цвет, тип, радиус):
        super().__init__(цвет, тип)
        self.радиус = радиус

    def площадь(self):
        return math.pi * self.радиус ** 2

    def периметр(self):
        return 2 * math.pi * self.радиус

круг = Круг('красный', 'круг', 5)
print(f'Площадь круга: {круг.площадь()}')
print(f'Периметр круга: {круг.периметр()}')
```

### Протокол работы программы:

Площадь круга: 78.53981633974483

Периметр круга: 31.41592653589793

Process finished with exit code 0

### Постановка задачи №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:

```
# Вариант 14 import pickle

class Машина:
    def __init__(self, марка, модель, год_выпуска):
        self.марка = марка
        self.модель = модель
        self.год_выпуска = год_выпуска

    def __str__(self):
        return f'Марка: {self.марка}, Модель: {self.модель}, Год выпуска: {self.год_выпуска}'

def save_def(objects, filename):
    with open(filename, 'wb') as f:
        pickle.dump(objects, f)

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

машины = [
    Машина('BMW', 'M5', 2021),
    Машина('Mercedes-Benz', 'S-Class', 2022),
    Машина('Audi', 'A8', 2023),
]

save_def(машины, 'машины.dat')

загруженные_машины = load_def('машины.dat')

for машина in загруженные_машины:
    print(машина)
```

### Протокол работы программы:

Марка: BMW, Модель: M5, Год выпуска: 2021

Марка: Mercedes-Benz, Модель: S-Class, Год выпуска: 2022

Марка: Audi, Модель: A8, Год выпуска: 2023

Process finished with exit code 0

**Вывод:** в процессе выполнения практического занятия закрепил усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрел навыки составления программ с использованием ООП в IDE PyCharm Community.