

## Практическое задание № 16

**Тема:** Составление программ для с ООП .

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

### Постановка задачи №1:

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.*

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

    def circle_square(self):
        return f'Площадь круга равна: {3.14 * self.radius ** 2}'

    def circle_length(self):
        return f'Длина круга равна: {3.14 * (2 * self.radius)}'

    def circle_diameter(self):
        return f'Диаметр круга равен: {2 * self.radius}'

circle = Circle(3)
print(circle.circle_square())
print(circle.circle_length())
print(circle.circle_diameter())
```

### Протокол работы программы:

Площадь круга равна: 28.26

Длина круга равна: 18.84

Диаметр круга равен: 6

Process finished with exit code 0

### Постановка задачи №2:

Создайте класс "Фигура", который содержит метод расчета площади фигуры.

Создайте классы "Квадрат" и "Прямоугольник", которые наследуются от класса "Фигура". Каждый класс должен иметь метод расчета площади собственной фигуры.

### Текст программы:

```
# Создайте класс "Фигура", который содержит метод расчета площади фигуры.
# Создайте классы "Квадрат" и "Прямоугольник", которые наследуются от класса
# "Фигура". Каждый класс должен иметь метод расчета площади собственной фигуры.
class Figure:
    def __init__(self, side, high):
        self.side = side
        self.high = high

    def figure_square(self):
        return f'{self.side * self.high}'

class Square(Figure):
    def square_square(self, side_sq):
        return f'{side_sq ** 2}'

class Rectangle(Figure):
    def rectangle_square(self, rec_length, rec_width):
        return f'{rec_length * rec_width}'

figure = Figure(3,6)
square = Square(3,4)

print(square.figure_square())
print(square.square_square(3))
print(issubclass(Square, Figure))
print(issubclass(Figure, Square))
print(square.__dict__)
```

### Протокол работы программы:

```
12
9
True
False
{'side': 3, 'high': 4}
```

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 режиме.

### Текст программы:

```

# 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 режиме.
import pickle

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

    def circle_square(self):
        return f'Площадь круга равна: {3.14 * self.radius ** 2}'

    def circle_length(self):
        return f'Длина круга равна: {3.14 * (2 * self.radius)}'

    def circle_diameter(self):
        return f'Диаметр круга равен: {2 * self.radius}'

def save_def():
    file = open('out.bin', 'wb')
    pickle.dump(circle1, file)
    pickle.dump(circle2, file)
    pickle.dump(circle3, file)
    file.close()

def load_def():
    file = open('out.bin', 'rb')
    first = pickle.load(file)
    second = pickle.load(file)
    third = pickle.load(file)
    print('\nИнформация, загруженная из бинарного файла', first, second, third, sep='\n')

circle1 = Circle(3)
circle2 = Circle(2)
circle3 = Circle(4)

print(circle1.circle_square())
print(circle2.circle_length())
print(circle3.circle_diameter())

save_def()
load_def()

```

### Протокол работы программы:

**Площадь круга равна: 28.26**

**Длина круга равна: 12.56**

**Диаметр круга равен: 8**

Информация, загруженная из бинарного файла

<\_\_main\_\_.Circle object at 0x0000020953A10760>

<\_\_main\_\_.Circle object at 0x0000020953A10520>

<\_\_main\_\_.Circle object at 0x0000020953A25B50>

Process finished with exit code 0

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

Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовый программный код выложен на GitHub.

