

## Практическое занятие № 16

Тема: составление программ с использованием ООП в IDE PyCharm Community

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

# Задачи

## №1

Создать класс "круг", который имеет атрибут радиуса и методы для вычисления площади, длины окружности и диаметра

Текст программы:

```
# Создать класс "круг", который имеет атрибут радиуса и методы
# для вычисления площади, длины окружности и диаметра
class Cir:
    def __init__(self, r):
        self.r = r

    def dia(self):
        return self.r*2
    def len(self):
        return self.r*3.14*2
    def pl(self):
        return self.r**2*3.14
a = Cir(100)
print(a.dia())
```

Протокол работы:

200

## №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.

Код 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):
        pass
```

```

def pl(self):
    return 'площадь рассчитывается только для конкретных фигур'
class Square(Figure):
    def __init__(self, a):
        super().__init__()
        self.a = a
    def pl(self):
        return self.a ** 2

class Rectangle(Figure):
    def __init__(self, a, b):
        super().__init__()
        self.a = a
        self.b = b
    def pl(self):
        return self.a * self.b

# Пример использования:
sq = Square(5)
re = Rectangle(4, 6)

print(f"Площадь квадрата: {sq.pl()}")
print(f"Площадь прямоугольника: {re.pl()}")

```

Протокол работы:

**Площадь квадрата: 25**

**Площадь прямоугольника: 24**

### № 3

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

Код программы:

```

# для задачи из блока 1 создать две функции, save_def и load_def, которые
# позволяют
# сохранить информацию из экземпляров класса (3 шт.) в файл и загружать ее
# обратно.
# Использовать модуль pickle для сериализации и десериализации объектов
# Python в
# бинарном формате

import pickle
from PZ 16 1 import Cir

def save_def(cir, file):
    with open(f'{file}.bin', 'wb') as file:
        pickle.dump(cir, file)

```

```
def load_def(file):  
    with open(f'{file}.bin', 'rb') as file:  
        return pickle.load(file)  
circle0 = Cir(5)  
circle1 = Cir(10)  
circle2 = Cir(20)  
save_def(circle0, 'f0')
```

Протокол работы

**200**

В процессе выполнения задания выработал основные принципы составления программ, приобрел навыки составления программ с использованием ООП в IDE PyCharm Community

Готовые решения выложены на Github