



ЛАБОРАТОРНАЯ РАБОТА

№ 2

Объектно-ориентированные возможности языка Python.

Группа ИУ5-35Б

Студент _____ / _____ / _____ Александров А.В _____ /

(Подпись, дата)

(И.О.Фамилия)

Преподаватель _____ / _____ / _____

(Подпись, дата)

(И.О.Фамилия)

Задание:

1. Необходимо создать виртуальное окружение и установить в него хотя бы один внешний пакет с использованием `pip`.
2. Необходимо разработать программу, реализующую работу с классами. Программа должна быть разработана в виде консольного приложения на языке Python 3.
3. Все файлы проекта (кроме основного файла `main.py`) должны располагаться в пакете `lab_python_oop`.
4. Каждый из нижеперечисленных классов должен располагаться в отдельном файле пакета `lab_python_oop`.
5. Абстрактный класс «Геометрическая фигура» содержит абстрактный метод для вычисления площади фигуры. Подробнее про абстрактные классы и методы Вы можете прочитать [здесь](#).
6. Класс «Цвет фигуры» содержит свойство для описания цвета геометрической фигуры. Подробнее про описание свойств Вы можете прочитать [здесь](#).
7. Класс «Прямоугольник» наследуется от класса «Геометрическая фигура». Класс должен содержать конструктор по параметрам «ширина», «высота» и «цвет». В конструкторе создается объект класса «Цвет фигуры» для хранения цвета. Класс должен переопределять метод, вычисляющий площадь фигуры.
8. Класс «Круг» создается аналогично классу «Прямоугольник», задается параметр «радиус». Для вычисления площади используется константа `math.pi` из модуля [math](#).
9. Класс «Квадрат» наследуется от класса «Прямоугольник». Класс должен содержать конструктор по длине стороны. Для классов «Прямоугольник», «Квадрат», «Круг»: Определите метод `"repr"`, который возвращает в виде строки основные параметры фигуры, ее цвет и площадь. Используйте метод `format` - <https://pyformat.info/>
Название фигуры («Прямоугольник», «Квадрат», «Круг») должно задаваться в виде поля данных класса и возвращаться методом класса.
10. В корневом каталоге проекта создайте файл `main.py` для тестирования Ваших классов (используйте следующую конструкцию - https://docs.python.org/3/library/_main_.html). Создайте следующие объекты и выведите о них информацию в консоль (N - номер Вашего варианта по списку группы):
Прямоугольник синего цвета шириной N и высотой N.
Круг зеленого цвета радиусом N.
Квадрат красного цвета со стороной N.
Также вызовите один из методов внешнего пакета, установленного с использованием `pip`.

Текст программы

```
from abc import ABC, abstractmethod
```

```
!python -m venv myenv
```

```
!myenv\Scripts\activate
```

```
import numpy as np
```

```
!pip install numpy
```

Создание одномерного массива

```
one_array = np.array([1, 2, 3, 4, 5])
```

```
print("Одномерный массив:")
```

```
print(one_array)
```

Одномерный массив:

[1 2 3 4 5]

```
class GeometricFigure(ABC):
```

```
def square(self):
```

pass

```
class FigureColor:
```

```
def __init__(self, color): #конструктор
```

```
self._color = color
```

@property

```
def color_property(self):
```

```
return self._color #защищенный атрибут
```

@color_property.setter

```
def color_property(self, value):
```

```
self._color = value
```

```
def __repr__(self):
```

```

return self._color

```

```
class Rectangle(GeometricFigure):
```

```
fR = "Прямоугольник" #наша фигура
```

```
@classmethod
```

```
def get fR(cls): #метод класса, обращаемся не к экземпляру
```

```

return cls.fR

```

```
def __init__(self, width, height, col):
```

```
self.width = width
```

```
self.height = height
```

```
self.col = FigureColor(col)
```

```
def square(self):
```

```
return self.width*self.height
```

```
def __repr__(self):
```

```

return "{} {} цвета, шириной {}, высотой {} , площадью {}".format(Rectangle.get_fR(),
self.col,
self.width,
self.height,

```

self.square())

```
class Circle(GeometricFigure):
    fC = "Круг" #наша фигура
    @classmethod
    def get_fC(cls): #метод класса, обращаемся не к экземпляру
        return cls.fC
    def __init__(self, radius, col):
        self.radius = radius
        self.col = FigureColor(col)
    def square(self):
        from math import pi
        return pi*(self.radius**2)
    def __repr__(self):
        return "{} {} цвета, радиус равен {}, а площадь равна {}".format(Circle.get_fC(),
                                                                            self.col,
                                                                            self.radius,
                                                                            self.square())

class Square(Rectangle):
    fS = "Квадрат" #наша фигура
    @classmethod
    def get_fS(cls): #метод класса, обращаемся не к экземпляру
        return cls.fS
    def __init__(self, side, col):
        self.side = side
        super().__init__(side, side, col)
    def __repr__(self):
        return "{} {} цвета, сторона равна {}, а площадь также {}".format(Square.get_fS(),
                                                                            self.col,
                                                                            self.side,
                                                                            self.square())

def main():
    r = Rectangle(1, 1, "синего")
    c = Circle(1, "зеленого")
    s = Square(1, "красного")
    print(r)
    print(c)
    print(s)
main()
```

Прямоугольник синего цвета, шириной 1, высотой 1 , площадью 1.

Круг зеленого цвета, радиус равен 1, а площадь равна 3.141592653589793.

Квадрат красного цвета, сторона равна 1, а площадь также 1.