

Парадигмы и конструкции языков программирования.

ОТЧЕТ
по лабораторной работе №2
«Объектно-ориентированные возможности языка Python».

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.
6. Класс «Цвет фигуры» содержит свойство для описания цвета геометрической фигуры.
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».

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.
- 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. 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.

Программа.

Ниже вы можете ознакомиться с классами файла lab_python_oop.

Класс Figure:

```
from abc import ABC, abstractmethod

class Figure (ABC):
    @abstractmethod
    def square (self):
        pass
```

Класс Color:

```
class Color:
    def __init__(self):
        self.color = None

    @property
    def colorproperty(self): # Get-аксесор
        return self._color

    @colorproperty.setter
    def colorproperty (self, value): #Set-аксесор
```

```
self._color = value
```

Класс Rectangle:

```
from lab_python_oop_figure import Figure
from lab_python_oop_color import Color

class Rectangle (Figure):
    figure_type = 'Прямоугольник'

    @classmethod
    def get_figure_type(cls):
        return cls.figure_type

    def __init__(self, color_param, width_param, height_param):
        self.width = width_param
        self.height = height_param
        self.fc = Color()
        self.fc.colorproperty = color_param

    def square(self):
        return (self.width * self.height)

    def __repr__(self):
        return '{}: цвет -- {}, ширина -- {}, высота -- {},
площадь -- {}'.format(
            Rectangle.get_figure_type(),
            self.fc.colorproperty,
            self.width,
            self.height,
            self.square ()
        )
```

Класс Square:

```
from lab_python_oop_rectangle import Rectangle

class Square(Rectangle):
    figure_type = 'Квадрат'

    @classmethod
    def get_figure_type(cls):
        return cls.figure_type

    def __init__(self, color_param, side_param):
        self.side = side_param
        super().__init__(color_param, self.side, self.side)

    def _repr__(self):
```

```

        return '{}: цвет -- {}, сторона -- {}, площадь --
{}'.format(
            Square.get_figure_type(),
            self.fc.colorproperty,
            self.side,
            self.square()
        )

```

Класс Circle:

```

from lab_python_oop_figure import Figure
from lab_python_oop_color import Color
import math

class Circle(Figure):
    figure_type = "Круг"

    @classmethod
    def get_figure_type(cls):
        return cls.figure_type

    def __init__(self, color_param, r_param):
        self.r = r_param
        self.fc = Color()
        self.fc.colorproperty = color_param

    def square(self):
        return math.pi*(self.r**2)

    def __repr__(self):
        return '{}: цвет -- {}, радиус -- {}, площадь --
{}'.format(
            Circle.get_figure_type(),
            self.fc.colorproperty,
            self.r,
            self.square()
        )

```

main.py:

```

from lab_python_oop_rectangle import Rectangle
from lab_python_oop_circle import Circle
from lab_python_oop_square import Square

def main():
    N = 12
    rect = Rectangle ('синий', N, N)
    circ = Circle ('зеленый', N)
    sq = Square ('красный', N)
    print (rect)

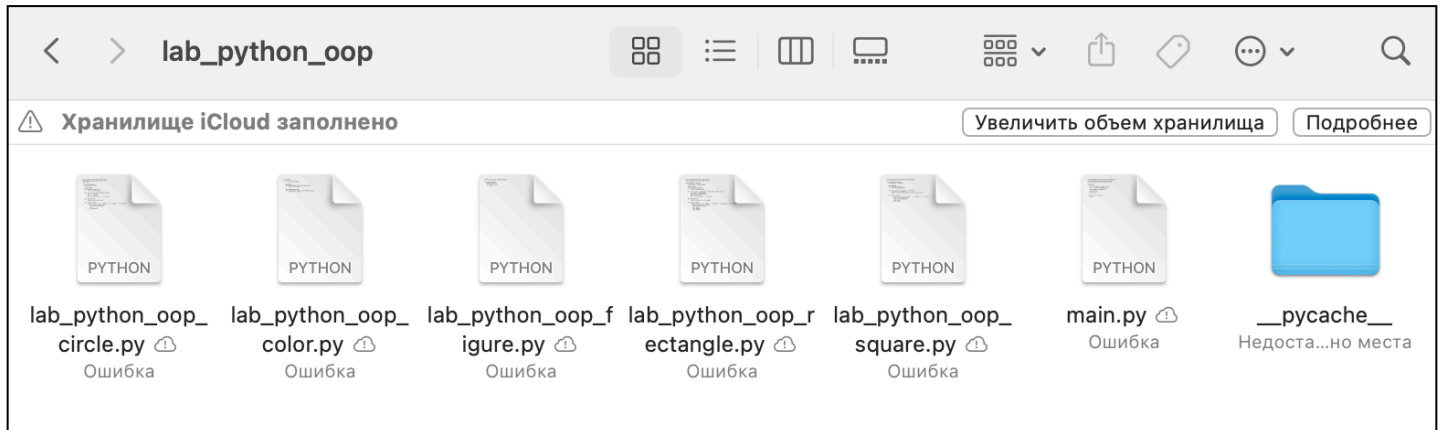
```

```

print (circ)
print (sq)

if __name__ == "__main__":
    main()

```



```

= RESTART: /Users/veraleonteva/Desktop/Парадигмы и конструкции программирования/
ЛР2/lab_python_oop/main.py
Прямоугольник: цвет -- синий, ширина -- 12, высота -- 12, площадь -- 144.
Круг: цвет -- зеленый, радиус -- 12, площадь -- 452.3893421169302.
Прямоугольник: цвет -- красный, ширина -- 12, высота -- 12, площадь -- 144.
>>> |

```