



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ

РАДИОТЕХНИЧЕСКИЙ

КАФЕДРА

СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2

Студент Столярова Ольга Денисовна
фамилия, имя, отчество

Группа РТ5-51Б

Название предприятия МГТУ им. Н. Э. Баумана

Студент

Столярова О.Д.

Преподаватель

Гапанюк Ю.Е.

2021 г.

Цель работы

Изучение объектно-ориентированных возможностей 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. 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»:

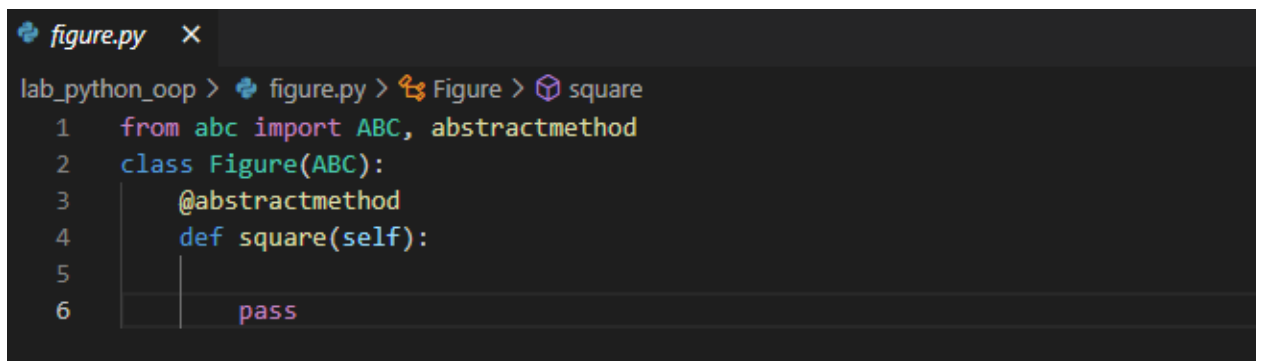
- Определите метод "repr", который возвращает в виде строки основные параметры фигуры, ее цвет и площадь. Используйте метод format - <https://pyformat.info/>
- Название фигуры («Прямоугольник», «Квадрат», «Круг») должно задаваться в виде поля данных класса и возвращаться методом класса.

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

figure.py



```

figure.py x
lab_python_oop > figure.py > Figure > square
1  from abc import ABC, abstractmethod
2  class Figure(ABC):
3      @abstractmethod
4      def square(self):
5
6      pass

```

circle.py

circle.py X

lab_python_oop > circle.py > Circle > square

```
1  from lab_python_oop.figure import Figure
2  from lab_python_oop.color import FigureColor
3  import math
4
5
6  class Circle(Figure):
7
8      FIGURE_TYPE = "Круг"
9
10     @classmethod
11     def get_figure_type(cls):
12         return cls.FIGURE_TYPE
13
14     def __init__(self, color_param, r_param):
15
16         self.r = r_param
17         self.fc = FigureColor()
18         self.fc.colorproperty = color_param
19
20     def square(self):
21
22         return math.pi*(self.r**2)
23
24     def __repr__(self):
25         return '{} {} цвета радиусом {} площадью {}'.format(
26             Circle.get_figure_type(),
27             self.fc.colorproperty,
28             self.r,
29             self.square()
30         )
```

rectangle.py

```
rectangle.py X
lab_python_oop > rectangle.py > Rectangle > square
1  from lab_python_oop.figure import Figure
2  from lab_python_oop.color import FigureColor
3
4
5  class Rectangle(Figure):
6
7      FIGURE_TYPE = "Прямоугольник"
8
9      @classmethod
10     def get_figure_type(cls):
11         return cls.FIGURE_TYPE
12
13     def __init__(self, color_param, width_param, height_param):
14
15         self.width = width_param
16         self.height = height_param
17         self.fc = FigureColor()
18         self.fc.colorproperty = color_param
19
20     def square(self):
21
22         return self.width*self.height
23
24     def __repr__(self):
25         return '{} {} цвета шириной {} и высотой {} площадью {}.'.format(
26             Rectangle.get_figure_type(),
27             self.fc.colorproperty,
28             self.width,
29             self.height,
30             self.square()
31         )
```

square.py

```
square.py X
lab_python_oop > square.py > Square > _repr_
1  from lab_python_oop.rectangle import Rectangle
2
3
4  class Square(Rectangle):
5
6      FIGURE_TYPE = "Квадрат"
7
8      @classmethod
9      def get_figure_type(cls):
10         return cls.FIGURE_TYPE
11
12     def __init__(self, color_param, side_param):
13         self.side = side_param
14         super().__init__(color_param, self.side, self.side)
15
16     def __repr__(self):
17         return '{} {} цвета со стороной {} площадью {}'.format(
18             Square.get_figure_type(),
19             self.fc.colorproperty,
20             self.side,
21             self.square()
22         )
```

color.py

```
color.py X
lab_python_oop > color.py > FigureColor > colorproperty
1  class FigureColor:
2
3      def __init__(self):
4          self._color = None
5
6      @property
7      def colorproperty(self):
8          """
9          Get-аксесор
10         """
11         return self._color
12
13     @colorproperty.setter
14     def colorproperty(self, value):
15         """
16         Set-аксесор
17         """
18         self._color = value
```

lab2.py

```
lab2.py X
lab2.py > ...
1 from lab_python_oop.rectangle import Rectangle
2 from lab_python_oop.circle import Circle
3 from lab_python_oop.square import Square
4 from colorama import init
5 init()
6 from colorama import Fore, Back, Style
7
8 def main():
9     r = Rectangle("синего", 18, 18)
10    c = Circle("зеленого", 18)
11    s = Square("красного", 18)
12    print(r)
13    print(c)
14    print(s)
15
16
17 print(Fore.GREEN + 'зеленый текст')
18 print(Back.YELLOW + 'на желтом фоне' + Style.RESET_ALL)
19
20 if __name__ == "__main__":
21     main()
```

Пример выполнения программы

```
PS C:\Users\olyas\python\lр2> & 'C:\Program Files\WindowsApps\PythonSoftwareFoundation.Python.3.9_3.9.2032.0_x64__qbz5n2kfra8p0\python3.9.exe
ions\ms-python.python-2021.10.1365161279\pythonFiles\lib\python\debugpy\launcher' '51453' '--' 'c:\Users\olyas\python\lр2\lab2.py'
I am init
зеленый текст
на желтом фоне
Прямоугольник синего цвета шириной 18 и высотой 18 площадью 324.
Круг зеленого цвета радиусом 18 площадью 1017.8760197630929.
Квадрат красного цвета со стороной 18 площадью 324.
PS C:\Users\olyas\python\lр2>
```

Ссылка на GitHub

<https://github.com/OlyaSto/Olyabmstu>