

Московский государственный технический
университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Базовые компоненты интернет-технологий»

Отчет по лабораторной работе №2
“Основные конструкции языка Python”

Выполнил:
студент группы ИУ5-34Б:
Малютин И.Д.
Подпись и дата:

Проверила:
преподаватель каф. ИУ5
Гапанюк Ю.Е.
Подпись и дата:

Москва, 2022

Описание задания:

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

12. 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`.

13. Дополнительное задание. Протестируйте корректность работы Вашей программы с помощью модульного теста.

Текст программы

Main.py:

```
from lab_python_oop.rectangle import Rectangle
from lab_python_oop.circle import Circle
from lab_python_oop.square import Square
from lab_python_oop.ellipse import Ellipse
import cowsay

def main():
    r = Rectangle("blue", 3, 2)
    c = Circle("green", 5)
    s = Square("red", 5)
    aboba = Ellipse("yellow", 5, 10)
    print(r)
    print(c)
    print(s)
    print(aboba)

    cowsay.milk('Hello world')

if __name__ == "__main__":
    main()
```

rectangle.py

```
from lab_python_oop.figure import Figure
from lab_python_oop.color import FigureColor

class Rectangle(Figure):
    FIGURE_TYPE = "Rectangle"

    @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 = FigureColor()
        self.fc.colorproperty = color_param

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

    def __repr__(self):
        return '{} {} {} {} {}.'.format(
            Rectangle.get_figure_type(),
            self.fc.colorproperty,
            self.width,
            self.height,
            self.square()
        )

```

figure.py

```

from abc import ABC, abstractmethod

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

```

ellipse.py

```

from lab_python_oop.figure import Figure
from lab_python_oop.color import FigureColor
import math

class Ellipse(Figure):
    FIGURE_TYPE = 'ellipse'

    def get_figure_type(cls):
        return cls.FIGURE_TYPE

    @classmethod
    def __init__(self, self_color, self_a, self_b):
        self.a = self_a
        self.b = self_b
        self.fc = FigureColor()
        self.fc.colorproperty = self_color

    def square(self):
        return self.a * self.b * math.pi

    def __repr__(self):
        return '{} {} , {} {}, {}'.format(
            Ellipse.get_figure_type(self),
            self.fc.colorproperty,
            self.a,
            self.b,
            self.square()
        )

```

color.py

```

class FigureColor:
    def __init__(self):
        self._color = None

    @property

```

```

def colorproperty(self):
    return self._color

@colorproperty.setter
def colorproperty(self, value):
    self._color = value

```

circle.py

```

from lab_python_oop.figure import Figure
from lab_python_oop.color import FigureColor
import math

class Circle(Figure):
    FIGURE_TYPE = "circle"

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

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

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

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

```

square.py

```

from lab_python_oop.figure import Figure
from lab_python_oop.color import FigureColor

class Square(Figure):
    FIGURE_TYPE = "square"

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

    def __init__(self, color_param, side_param):
        self.side = side_param
        self.fc = FigureColor()
        self.fc.colorproperty = color_param

    def __repr__(self):
        return '{} {} {} {}'.format(
            Square.get_figure_type(),
            self.fc.colorproperty,
            self.side,
            self.square()
        )

```

```
def square(self):
    return self.side * self.side
```

Выполнение программы:

```
Rectangle blue 3 2 6.  
circle green 5 78.53981633974483.  
square red 5 25.  
ellipse yellow , 5 10, 157.07963267948966.
```

```
-----
| Hello world |
=====
```

```
Process finished with exit code 0
```