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

(МГТУ им. Н. Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления » (ИУ)

КАФЕДРА «Системы обработки информации и управления » (ИУ5)

Отчёт по лабораторной работе №2

По курсу «Базовые компоненты интернет-технологий»

Студент:
Козлов Егор Васильевич, группа ИУ5-32Б (2 курс)

_____
(подпись, дата)

Проверил:
Гапанюк Юрий Евгеньевич

_____
(подпись, дата)

Москва, 2021

Задание:

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:

Файл main.py:

```
from lab_python_oop.rectangle import Rectangle
from lab_python_oop.circle import Circle
from lab_python_oop.square import Square
from rich import print

def main():
    r = Rectangle("синего", 20, 20)
    c = Circle("зеленого", 20)
    s = Square("красного", 20)
    print("[italic blue]Этот", r)
    print("[italic green]Этот",c)
    print("[italic red]Этот",s)

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

Файл circle.py:

```
from lab_python_oop.figure import Figure
from lab_python_oop.color import FigureColor
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 = FigureColor()
        self.fc.colorproperty = color_param

    def square(self):

        return math.pi*(self.r**2)

    def __repr__(self):
        return '{} {} цвета радиусом {} площадью {}.'.format(
            Circle.get_figure_type(),
```

```
            self.fc.colorproperty,
            self.r,
            self.square()
        )
```

Файл 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
```

Файл figure.py:

```
from abc import ABC, abstractmethod


class Figure(ABC):

    @abstractmethod
    def square(self):

        pass
```

Файл rectangle.py:

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


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 = FigureColor()
        self.fc.colorproperty = color_param

    def square(self):

        return self.width*self.height
```

```
    def __repr__(self):
        return '{} {} цвета шириной {} и высотой {} площадью {}.'.format(
            Rectangle.get_figure_type(),
            self.fc.colorproperty,
            self.width,
            self.height,
            self.square()
        )
```

Файл square.py:

```
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 '{} {} цвета со стороной {} площадью {}.'.format(
            Square.get_figure_type(),
            self.fc.colorproperty,
            self.side,
            self.square()
        )
```

Примеры выполнения: