**Московский государственный технический**

**университет им. Н.Э. Баумана**

**Факультет ИУ**

**Кафедра ИУ5**

**Курс «Парадигмы и конструкции языков программирования»**

**Отчет лабораторной работе №2**

Выполнил студент группы ИУ5-33Б:

Костяев В.Д.

Подпись и дата:

Проверил преподаватель каф.:

Гапанюк Ю. Е.

Подпись и дата:

Москва, 2024 г

**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.**](https://docs.python.org/3/library/abc.html)
6. **Класс «Цвет фигуры» содержит свойство для описания цвета геометрической фигуры. Подробнее про описание свойств Вы можете прочитать**[**здесь.**](https://docs.python.org/3/library/functions.html#property)
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.**](https://docs.python.org/3/library/math.html)
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/**](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**](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.**
11. **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 colorama import Fore, Style

def main():

N = 5 # Замените на номер вашего варианта

# Создание объектов

rectangle = Rectangle(width=N, height=N, color\_name="Синий")

circle = Circle(radius=N, color\_name="Зелёный")

square = Square(side\_length=N, color\_name="Красный")

# Вывод информации о фигурах

print(Fore.BLUE + str(rectangle) + Style.RESET\_ALL)

print(Fore.GREEN + str(circle) + Style.RESET\_ALL)

print(Fore.RED + str(square) + Style.RESET\_ALL)

# Вызов метода внешнего пакета colorama

print(Fore.YELLOW + "Это сообщение выделено с помощью внешнего пакета colorama." + Style.RESET\_ALL)

if \_\_name\_\_ == "\_\_main\_\_":

main()

**square.py**

from lab\_python\_oop.rectangle import Rectangle

class Square(Rectangle):

figure\_name = "Квадрат"

def \_\_init\_\_(self, side\_length, color\_name):

super().\_\_init\_\_(width=side\_length, height=side\_length, color\_name=color\_name)

self.side\_length = side\_length # Дополнительный атрибут для удобства

def \_\_repr\_\_(self):

return "{name}: Сторона={side}, Цвет={color}, Площадь={area}".format(

name=self.figure\_name,

side=self.side\_length,

color=self.color.color\_name,

area=self.calculate\_area()

)

**color.py**

class Color:

def \_\_init\_\_(self, color\_name):

self.\_color\_name = color\_name

@property

def color\_name(self):

return self.\_color\_name

@color\_name.setter

def color\_name(self, value):

self.\_color\_name = value

**geometric\_figure.py**

from abc import ABC, abstractmethod

class GeometricFigure(ABC):

@abstractmethod

def calculate\_area(self):

pass

**rectangle.py**

from lab\_python\_oop.geometric\_figure import GeometricFigure

from lab\_python\_oop.color import Color

class Rectangle(GeometricFigure):

figure\_name = "Прямоугольник"

def \_\_init\_\_(self, width, height, color\_name):

self.width = width

self.height = height

self.color = Color(color\_name)

def calculate\_area(self):

return self.width \* self.height

def \_\_repr\_\_(self):

return "{name}: Ширина={width}, Высота={height}, Цвет={color}, Площадь={area}".format(

name=self.figure\_name,

width=self.width,

height=self.height,

color=self.color.color\_name,

area=self.calculate\_area()

)

**init.py**

#python -m venv venv

#venv\Scripts\activate

#pip install colorama

#pip freeze > requirements.txt

**circle.py**

import math

from lab\_python\_oop.geometric\_figure import GeometricFigure

from lab\_python\_oop.color import Color

class Circle(GeometricFigure):

figure\_name = "Круг"

def \_\_init\_\_(self, radius, color\_name):

self.radius = radius

self.color = Color(color\_name)

def calculate\_area(self):

return math.pi \* self.radius \*\* 2

def \_\_repr\_\_(self):

return "{name}: Радиус={radius}, Цвет={color}, Площадь={area:.2f}".format(

name=self.figure\_name,

radius=self.radius,

color=self.color.color\_name,

area=self.calculate\_area()

)

**test\_figures.py**

import unittest

from lab\_python\_oop.rectangle import Rectangle

from lab\_python\_oop.circle import Circle

from lab\_python\_oop.square import Square

class TestGeometricFigures(unittest.TestCase):

def test\_rectangle\_area(self):

rect = Rectangle(width=4, height=5, color\_name="Красный")

self.assertEqual(rect.calculate\_area(), 20)

def test\_circle\_area(self):

circle = Circle(radius=3, color\_name="Зелёный")

self.assertAlmostEqual(circle.calculate\_area(), 28.274333882308138)

def test\_square\_area(self):

square = Square(side\_length=4, color\_name="Синий")

self.assertEqual(square.calculate\_area(), 16)

def test\_square\_inheritance(self):

square = Square(side\_length=4, color\_name="Синий")

self.assertIsInstance(square, Rectangle)

self.assertIsInstance(square, Square)

def test\_color\_property(self):

rect = Rectangle(width=2, height=3, color\_name="Жёлтый")

self.assertEqual(rect.color.color\_name, "Жёлтый")

rect.color.color\_name = "Фиолетовый"

self.assertEqual(rect.color.color\_name, "Фиолетовый")

if \_\_name\_\_ == '\_\_main\_\_':

unittest.main()

**Экранные формы с примерами выполнения программы:**

