**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ**

**Московский государственный технический университет**

**им. Н.Э. Баумана**

**(МГТУ им. Н.Э. Баумана)**

**Кафедра «Системы обработки информации и управления» (ИУ5)**

Отчёт по лабораторной работе № 2

По курсу: «Базовые компоненты интернет-технологий»

Выполнил: Строганов Георгий Константинович

студент группы ИУ5-31Б.

Проверил: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата: \_\_\_.\_\_\_.2022г.

Подпись: \_\_\_\_\_\_\_\_\_\_\_.

г. Москва 2022 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.
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». 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/
10. Название фигуры («Прямоугольник», «Квадрат», «Круг») должно задаваться в виде поля данных класса и возвращаться методом класса.
11. 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.

**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

import sys

def get\_side\_rad(prompt = "Введите значение стороны"):

print(prompt)

try:

# Пробуем прочитать значение из командной строки

side\_str = sys.argvp

except:

# Вводим с клавиатуры

side\_str = input()

# Обрабатываем неправильный ввод

while True:

try:

side = float(side\_str)

except:

print("Введены неправильные данные.")

side\_str = input()

else:

return side

def get\_color(termpt):

print("Введите цвет будущего", termpt)

return str(input())

def main():

print("Выберете тип ввода:\n1 - с клавиатуры в консоли\n2 - вариант 16")

while True:

choise = int(input())

if choise == 1 or choise == 2:

break

else:

print("Введено неверное значение. Введите 1 или 2")

if choise == 1:

r = Rectangle(get\_color("прямоугольника"), get\_side\_rad(), get\_side\_rad())

c = Circle(get\_color("круга"), get\_side\_rad("Введите радиус"))

s = Square(get\_color("квадрата"), get\_side\_rad())

print(r, c, s, sep = "\n")

else:

print(Rectangle("синего", 16, 16))

print(Circle("зеленого", 16))

print(Square("красного", 16))

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

main()

**circle.py**

from lab\_python\_oop.figure import Figure

from lab\_python\_oop.color import FigureColor

import math

class Circle(Figure):

"""

Класс «Круг» наследуется от класса «Геометрическая фигура».

"""

type = "Круг"

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.type, self.fc.colorproperty, self.r, 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

**figure.py**

from abc import ABC, abstractmethod

#Абстрактный класс «Геометрическая фигура»

class Figure(ABC):

@abstractmethod

def square(self):

pass

**rectangle.py**

from lab\_python\_oop.figure import Figure

from lab\_python\_oop.color import FigureColor

class Rectangle(Figure):

"""

Класс «Прямоугольник» наследуется от класса «Геометрическая фигура».

"""

type = "Прямоугольник"

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.type, self.fc.colorproperty, self.width, self.height, self.square())

**square.py**

from lab\_python\_oop.rectangle import Rectangle

class Square(Rectangle):

"""

Класс «Квадрат» наследуется от класса «Прямоугольник».

"""

type = "Квадрат"

def \_\_init\_\_(self, color\_param, side\_param):

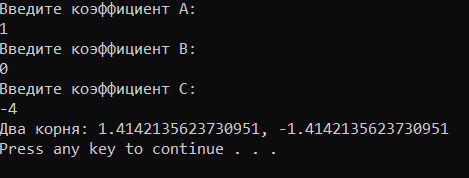
self.side = side\_param

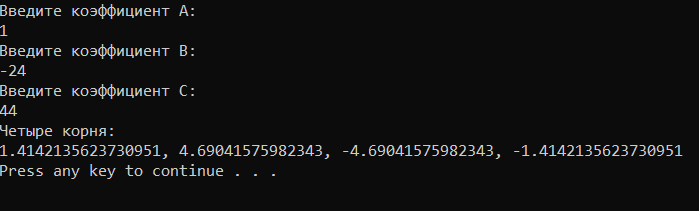
super().\_\_init\_\_(color\_param, self.side, self.side)

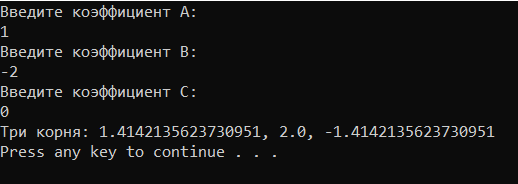
def \_\_repr\_\_(self):

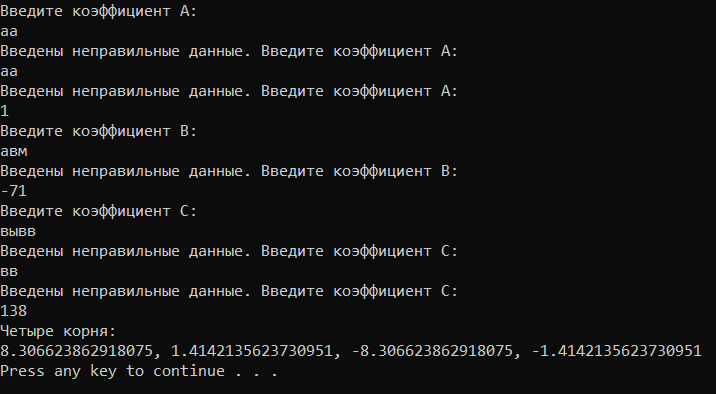
return '{} {} цвета со стороной {} площадью {}.'.format( Square.type, self.fc.colorproperty, self.side, self.square())

**Результаты тестирования:**









**Результат выполнения с виртуальным окружением:**

