** 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 «Информатика и системы управления» Кафедра «Системы обработки информации и управления»**

Лабораторная работа №2

**«Объектно-ориентированные возможности Python»**

по предмету

«Базовые компоненты интернет-технологий»

Выполнил:

студент группы № ИУ5-35Б

Обухов Антон

Проверил:

Преподаватель кафедры ИУ-5

Гапанюк Юрий

2022 г.

Постановка задачи:

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:

**Main.py**

from lab\_python\_oop.rectangle import Rectangle

from lab\_python\_oop.circle import Circle

from lab\_python\_oop.square import Square

import pandas as pd

def is\_float(a):

try:

float(a)

return True

except:

return False

def get\_coef(index, prompt):

try:

# Пробуем прочитать коэффициент из командной строки

coef\_str = sys.argv[index]

except:

print(prompt)

coef\_str = input()

# Переводим строку в действительное число

try:

coef = float(coef\_str)

except:

coef=None

return coef

def get\_str(index, prompt):

try:

# Пробуем прочитать коэффициент из командной строки

coef\_str = sys.argv[index]

except:

print(prompt)

coef\_str = input()

# Переводим строку в действительное число

try:

coef = str(coef\_str)

except:

coef=None

return coef

def main():

print('\n ---- LAB-01 ----\n')

a,b=None,None

i=0

while not is\_float(a):

i+=1

a = get\_coef(i, 'Введите сторону ширину прямоугольника:')

while not is\_float(b):

i+=1

b = get\_coef(i, 'Введите сторону высоту прямоугольника:')

i+=1

color=get\_str(i,'Введите какого цвета будет ваш прямоугольник:')

r = Rectangle(color, a, b)

a,b=None,None

while not is\_float(b):

i+=1

b = get\_coef(i, 'Введите радиус круга:')

i+=1

color=get\_str(i,'Введите какого цвета будет ваш круг:')

c = Circle(color, b)

while not is\_float(a):

i+=1

a = get\_coef(i, 'Введите сторону квадрата:')

i+=1

color=get\_str(i,'Введите какого цвета будет ваш квадрат:')

s = Square(color, a)

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

# пример выполнения импортированной библеотеки.

print('\n ---- Module Pandas ----\n')

table = pd.DataFrame({'Anton': ['Python', 2, 'BKIT'], 'Obukhov': ["Developer", 'informathion', 11]}, )

print(table)

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

main()

lab\_python\_oop/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».

"""

FIGURE\_TYPE = "Круг"

@classmethod

def get\_figure\_type(cls):

return cls.FIGURE\_TYPE

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

"""

Класс должен содержать конструктор по параметрам «радиус» и «цвет». В конструкторе создается объект класса «Цвет фигуры» для хранения цвета.

"""

self.r = r\_param

self.fc = FigureColor()

self.fc.colorproperty = color\_param

def square(self):

"""

Класс должен переопределять метод, вычисляющий площадь фигуры.

"""

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

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

return '{} {} радиусом {} площадью {}.'.format(

Circle.get\_figure\_type(),

self.fc.colorproperty,

self.r,

self.square()

)

lab\_python\_oop/color.py

class FigureColor:

"""

Класс «Цвет фигуры»

"""

def \_\_init\_\_(self):

self.\_color = None

@property

def colorproperty(self):

"""

Get-аксессор

"""

return self.\_color

@colorproperty.setter

def colorproperty(self, value):

"""

Set-аксессор

"""

self.\_color = value

lab\_python\_oop/figure.py

from abc import ABC, abstractmethod

class Figure(ABC):

"""

Абстрактный класс «Геометрическая фигура»

"""

@abstractmethod

def square(self):

"""

содержит виртуальный метод для вычисления площади фигуры.

"""

pass

lab\_python\_oop/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».

"""

FIGURE\_TYPE = "Прямоугольник"

@classmethod

def get\_figure\_type(cls):

return cls.FIGURE\_TYPE

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

"""

Класс должен содержать конструктор по параметрам «ширина», «высота» и «цвет». В конструкторе создается объект класса «Цвет фигуры» для хранения цвета.

"""

self.width = width\_param

self.height = height\_param

self.fc = FigureColor()

self.fc.colorproperty = color\_param

def square(self):

"""

Класс должен переопределять метод, вычисляющий площадь фигуры.

"""

return self.width \* self.height

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

return '{} {} шириной {} и высотой {} площадью {}.'.format(Rectangle.get\_figure\_type(),self.fc.colorproperty,self.width,self.height,self.square())

lab\_python\_oop/square.py

from lab\_python\_oop.rectangle import Rectangle

class Square(Rectangle):

"""

Класс «Квадрат» наследуется от класса «Прямоугольник».

"""

FIGURE\_TYPE = "Квадрат"

@classmethod

def get\_figure\_type(cls):

return cls.FIGURE\_TYPE

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

"""

Класс должен содержать конструктор по параметрам «сторона» и «цвет».

"""

self.side = side\_param

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

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

return '{} {} со стороной {} площадью {}.'.format(

Square.get\_figure\_type(),

self.fc.colorproperty,

self.side,

self.square()

)

Результаты выполнения программы:

