**Московский государственный технический** **университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Отчет по лабораторной работе №2

«Объектно-ориентированные возможности языка Python»

Выполнила: Проверил:

Студентка группы ИУ5-33Б преподаватель каф. ИУ5

Коренева София Гапанюк Ю. Е.

Москва, 2023 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.](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»: o Определите метод "repr", который возвращает в виде строки основные параметры фигуры, ее цвет и площадь. Используйте метод format.

o Название фигуры («Прямоугольник», «Квадрат», «Круг») должно задаваться в виде поля данных класса и возвращаться методом класса.

1. 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. o 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 numpy as np   def main():  r = Rectangle(15, 15, "синего")  print(r.\_\_repr\_\_())  c = Circle(15, "зеленого")  print(c.\_\_repr\_\_())  s = Square(15, "красного")  print(s.\_\_repr\_\_())  a = np.array([[1, 2, 3], [4, 5, 6]])  print(a)   if \_\_name\_\_ == "\_\_main\_\_":  main() |

lab\_python\_oop/figure

from abc import ABC, abstractmethod  
  
  
class Figure(ABC):  
  
 @abstractmethod  
 def square(self):  
 pass

lab\_python\_oop/color

|  |
| --- |
| class Color:   def \_\_init\_\_(self):  self.\_color = None   @property  def color\_property(self):  return self.\_color   @color\_property.setter  def color\_property(self, color):  self.\_color = color |

lab\_python\_oop/rectangle

from lab\_python\_oop.figure import Figure  
from lab\_python\_oop.color import Color  
  
  
class Rectangle(Figure):  
  
 FIGURE\_TYPE = "Прямоугольник"  
  
 @classmethod  
 def get\_figure\_type(cls):  
 return cls.FIGURE\_TYPE  
  
 def \_\_init\_\_(self, width, height, color):  
 self.width = width  
 self.height = height  
 self.color = Color()  
 self.color.color\_property = color  
  
 def square(self):  
 return self.width \* self.height  
  
 def \_\_repr\_\_(self):  
 return '{} {} цвета шириной {} и высотой {} площадью {}.'.format(  
 Rectangle.get\_figure\_type(), self.color.color\_property,  
 self.width, self.height, self.square())

lab\_python\_oop/circle

|  |
| --- |
| from lab\_python\_oop.figure import Figure from lab\_python\_oop.color import Color import math   class Circle(Figure):   FIGURE\_TYPE = "Круг"   @classmethod  def get\_figure\_type(cls):  return cls.FIGURE\_TYPE   def \_\_init\_\_(self, radius, color):  self.radius = radius  self.color = Color()  self.color.color\_property = color   def square(self):  return math.pi \* (self.radius \*\* 2)   def \_\_repr\_\_(self):  return '{} {} цвета радиусом {} площадью {}.'.format(  Circle.get\_figure\_type(), self.color.color\_property, self.radius, self.square()) |

lab\_python\_oop/square

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, side, color):  
 self.side = side  
 super().\_\_init\_\_(self.side, self.side, color)  
  
 def \_\_repr\_\_(self):  
 return '{} {} цвета со стороной {} площадью {}.'.format(  
 Square.get\_figure\_type(), self.color.color\_property, self.side, self.square())

Работа программы

