|  |  |
| --- | --- |
| Изображение выглядит как текст, керамические изделия, фарфор  Автоматически созданное описание | **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 «ГУИМЦ»**

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

Дисциплина «Базовые компоненты ИТ»

ОТЧЁТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ №2

«Объектно-ориентированные возможности языка Python»

Студент: Аброчнов Е.С., группа ИУ5Ц-53Б

Преподаватель: Гапанюк Ю.Е.

2021 г.

**1. Описание задания**

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.

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/

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

12) 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.

**2. 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  
  
def main():  
  
 r = Rectangle("оранжевого", 2, 2)  
 c = Circle("белого", 2)  
 s = Square("синего", 2)  
 print(r)  
 print(c)  
 print(s)  
  
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):  
  
 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()  
 )

**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  
  
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):  
 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()  
 )

**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()  
 )

**3. Экранные формы с примерами выполнения программы**

Изображение выглядит как текст

Автоматически созданное описание