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

|  |  |  |
| --- | --- | --- |
|  |  |  |

**Отчет по лабораторной работе №2 по курсу**

**РИП**

«Python. Объектно-ориентированные возможности»

4

(количество листов)

|  |  |
| --- | --- |
| ИСПОЛНИТЕЛЬ: |  |
| студент группы **ИУ5-52** | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  | (подпись) |
| **Писарчук Н.** | "15" сентября 2018 г. |

Москва, МГТУ - 2018

1. Задание

**Необходимо разработать программу, реализующую работу с классами.**

1. Программа должна быть разработана в виде консольного приложения на языке Python 3.
2. Все файлы проекта (кроме основного файла main.py) должны располагаться в пакете lab\_python\_oop.
3. Каждый из нижеперечисленных классов должен располагаться в отдельном файле пакета lab\_python\_oop.
   1. 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>
   2. 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>
   3. 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.
   4. 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>
   5. Класс «Квадрат» наследуется от класса «Прямоугольник». Класс должен содержать конструктор по длине стороны.
   6. Для классов «Прямоугольник», «Квадрат», «Круг»:
      1. Определите метод "repr", который возвращает в виде строки основные параметры фигуры, ее цвет и площадь. Используйте метод format - <https://pyformat.info/>
      2. 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>). Создайте следующие объекты и выведите о них информацию в консоль:
   1. Прямоугольник синего цвета шириной 3 и высотой 2.
   2. Круг зеленого цвета радиусом 5.
   3. Квадрат красного цвета со стороной 5.

2. Исходный код

***main.py:***

import lab\_python\_oop.geometric\_figure

from lab\_python\_oop.circle import circle

from lab\_python\_oop.square import square

from lab\_python\_oop.rectangle import rectangle

Rectangle = rectangle(3,2,'синий')

Circle = circle(5,"зеленый")

Square = square(5,'красный')

print(Rectangle)

print(Circle)

print(Square)

Пакет ***lab\_python\_oop***:

***circle.py:***

import math

from .geometric\_figure import \*

from .colour\_figure import \*

class circle(geometric\_figure):

def \_\_init\_\_(self, R, colour):

self.radius = R

self.colour = colour\_figure(colour)

def area(self):

return (self.radius \* self.radius \* math.pi)

def name(self):

return('Круг');

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

return '{} цвета {}, радиуса {} и площадью {}'.format(self.name(), self.colour.colour, self.radius, self.area())

***colour\_figure.py***

class colour\_figure:

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

self.col = colour

@property

def colour(self):

return self.col

***geometric\_figure.py***

from abc import \*

class geometric\_figure(metaclass = ABCMeta):

@abstractmethod

def area(self):

pass

***rectangle.py***

from .geometric\_figure import \*

from .colour\_figure import \*

class rectangle(geometric\_figure):

def \_\_init\_\_(self, width, height, colour):

self.width = width

self.height = height

self.colour = colour\_figure(colour)

def area(self):

return (self.width \* self.height)

def name(self):

return('Прямоугольник');

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

return '{} цвета {}, шириной {}, высотой {} и площадью {}'.format(self.name(), self.colour.colour, self.width, self.height, self.area())

***square.py***

from .geometric\_figure import \*

from .colour\_figure import \*

from .rectangle import \*

class square(rectangle):

def \_\_init\_\_(self, height, colour):

self.side = height

self.colour = colour\_figure(colour)

def area(self):

return (self.side \* self.side)

def name(self):

return('Квадрат');

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

return '{} цвета {}, со стороной {} и площадью {}'.format(self.name(), self.colour.colour, self.side, self.area())

3. Скриншот

