**Московский государственный технический**

**университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

Кафедра «Системы обработки информации и управления»

Курс «Разработка Интернет-приложений»

Отчет по лабораторной работе №3

«Python. Объектно-ориентированные возможности»

|  |  |  |
| --- | --- | --- |
| Выполнила: |  | Проверил: |
| студентка группы ИУ5-53 |  | преподаватель каф. ИУ5 |
| Беспалова У.А. |  | Гапанюк Ю.Е. |
| Подпись и дата:  29.09.2019 |  | Подпись и дата: |

Москва, 2019 г.

1. **Содержание задания**

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.
   2. Класс «Цвет фигуры» содержит свойство для описания цвета геометрической фигуры.
   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.
   5. Класс «Квадрат» наследуется от класса «Прямоугольник». Класс должен содержать конструктор по длине стороны.
   6. Для классов «Прямоугольник», «Квадрат», «Круг»:
      * Определите метод "repr", который возвращает в виде строки основные параметры фигуры, ее цвет и площадь. Используйте метод format.
      * 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. **Текст программы**

Main.py

from rect import Rect

from square import Square

from circle import Circle

import unittest

class TestFigure (unittest.TestCase):

def test\_rect (self):

self.assertEqual (str(Rect ("синего", 3, 2)), "Прямоугольник синего цвета, шириной 3 и высотой 2, площадью 6.", "Неверный формат вывода данных о прямоугольнике")

def test\_square (self):

self.assertEqual (str(Square ("красного", 5)), "Квадратик красного цвета, со стороной 5, площадью 25.", "Неверный формат вывода данных о квадрате")

def test\_circle (self):

self.assertEqual (str(Circle ("зелёного", 5)), "Круг зелёного цвета, радиусом 5, площадью 78.53981633974483.", "Неверный формат вывода данных о круге")

# def test\_circle\_must\_be\_fallen (self):

# self.assertEqual (str(Circle ("зелёного", 1)), "Круг зелёного цвета, радиусом 5, площадью 78.53981633974483.", "Неверный формат вывода данных о круге")

def main():

r = Rect ("синего", 3, 2)

print (r)

sq = Square ("красного", 5)

print (sq)

c = Circle ("зелёного", 5)

print (c)

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

main()

print ("\n\nРезультаты выполнения теста")

unittest.main ()

Color.py

class Color:

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

self.\_color = None

@property

def color\_property (self):

return self.\_color

@color\_property.setter

def color\_property (self, value):

self.\_color = value

Figure.py

from abc import ABC, ABCMeta, abstractmethod

class Figure (ABC):

@abstractmethod

def Area (self): #абстрактный метод

pass

@classmethod

def get\_figure\_type (cls):

return cls.Figure\_type

Circle.py

from figure import Figure

from color import Color

import math

class Circle (Figure):

Figure\_type = "Круг"

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

self.r = r\_param

self.fig\_color = Color()

self.fig\_color.color\_property = color\_param

def Area(self):

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

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

return '{} {} цвета, радиусом {}, площадью {}.'.format (Circle.get\_figure\_type(), self.fig\_color.color\_property, self.r, self.Area())

Rect.py

from color import Color

from figure import Figure

class Rect (Figure):

Figure\_type = "Прямоугольник"

def \_\_init\_\_(self, color\_param, w\_param, h\_param):

self.w = w\_param

self.h = h\_param

self.fig\_color = Color()

self.fig\_color.color\_property = color\_param

def Area (self):

return self.w\*self.h

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

return '{} {} цвета, шириной {} и высотой {}, площадью {}.'. format(Rect.get\_figure\_type(), self.fig\_color.color\_property, self.w, self.h, self.Area())

Square.py

from rect import Rect

class Square (Rect):

Figure\_type = "Квадратик"

# @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.fig\_color.color\_property, self.side, self.Area())

test.py

import unittest

class TestFigure (unittest.TestCase):

def test\_rect (self):

self.assertEqual (str(Rect ("синего", 3, 2)), "Прямоугольник синего цвета, шириной 3 и высотой 2, площадью 6.", "Неверный формат вывода данных о прямоугольнике")

def test\_square (self):

self.assertEqual (str(Square ("красного", 5)), "Квадратик красного цвета, со стороной 5, площадью 25.", "Неверный формат вывода данных о квадрате")

def test\_circle (self):

self.assertEqual (str(Circle ("зелёного", 5)), "Круг зелёного цвета, радиусом 5, площадью 78.53981633974483.", "Неверный формат вывода данных о круге")

def test\_circle\_must\_be\_fallen (self):

self.assertEqual (str(Circle ("зелёного", 1)), "Круг зелёного цвета, радиусом 5, площадью 78.53981633974483.", "Неверный формат вывода данных о круге")

1. **Экранные формы с примерами работы программы**

