**Московский государственный технический**

**университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

Кафедра ИУ5. Курс «Базовые компоненты интернет-технологий»

Отчет по лабораторной работе №2

«**Объектно-ориентированные возможности языка Python.**»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-32Б  Казицин Алексей |  | преподаватель кафедры  Гапанюк Ю.Е. |
|  |  |  |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2022 г.

**Цель лабораторной работы:** изучение объектно-ориентированных возможностей языка Python.

**Требования к отчету:**

Отчет по лабораторной работе должен содержать:

1. титульный лист;
2. описание задания;
3. текст программы;
4. экранные формы с примерами выполнения программы.

**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. Подробнее про абстрактные классы и методы Вы можете прочитать [здесь.](https://docs.python.org/3/library/abc.html)
6. 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)
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»:
   * 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.
11. **Дополнительное задание.** Протестируйте корректность работы Вашей программы с помощью модульного теста.

Текст программы:

Main.py

import numpy  
  
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("синего", 8, 8)  
 c = Circle("зеленого", 8)  
 s = Square("красного", 8)  
  
 print(r)  
 print(c)  
 print(s)  
 a = numpy.array([[1, 2, 3], [4, 5, 6]])  
 print(a)  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

Circle.py

import math  
  
from lab\_python\_oop.figure import Figure  
from lab\_python\_oop.color import Color  
  
  
class Circle(Figure):  
 *"""  
 Класс "Круг"  
 """* FIGURE\_TYPE = "Круг"  
  
 @classmethod  
 def get\_figure\_type(cls):  
 return cls.FIGURE\_TYPE  
  
 def \_\_init\_\_(self, color\_param, r\_param):  
 self.fc = Color()  
 self.fc.color\_property = color\_param  
 self.r = r\_param  
  
 def square(self):  
 return 2 \* math.pi \* self.r \*\* 2  
  
 def \_\_repr\_\_(self):  
 return '{} {} цвета, радиуса {}, площадью {:.3f}.'.format(  
 self.get\_figure\_type(),  
 self.fc.color\_property,  
 self.r,  
 self.square()  
 )

Color.py

class Color:  
 *"""  
 Класс "Цвет фигуры"  
 """* def \_\_init\_\_(self):  
 self.\_color = None  
  
 @property  
 def color\_property(self):  
 *"""  
 Get-аксессор  
 """* return self.\_color  
  
 @color\_property.setter  
 def color\_property(self, value):  
 *"""  
 Set-аксессор  
 """* self.\_color = value

Figure.py

from abc import ABC, abstractmethod  
  
  
class Figure:  
 *"""  
 Абстрактный класс "Геометрическая фигура"  
 """* @abstractmethod  
 def square(self):  
 *"""  
 Виртуальный метод вычисления площади фигуры  
 """* pass

Rectangle.py

from lab\_python\_oop.figure import Figure  
from lab\_python\_oop.color import Color  
  
  
class Rectangle(Figure):  
 *"""  
 Класс "Прямоугольник"  
 """* 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 = Color()  
 self.fc.color\_property = color\_param  
  
 def square(self):  
 *"""  
 Вычисление площади фигуры  
 """* return self.width \* self.height  
  
 def \_\_repr\_\_(self):  
 return '{} {} цвета, шириной {} и высотой {}, площадью {:.3f}.'.format(  
 Rectangle.get\_figure\_type(),  
 self.fc.color\_property,  
 self.width,  
 self.height,  
 self.square()  
 )

Square.py

from lab\_python\_oop.rectangle import Rectangle  
  
  
class Square(Rectangle):  
 *"""  
 Класс "Квадрат"  
 """* 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 "цвет" и "сторона"  
 """* self.side = side\_param  
 super().\_\_init\_\_(color\_param, self.side, self.side)  
  
 def \_\_repr\_\_(self):  
 return '{} {} цвета со стороной {}, площадью {:.3f}.'.format(  
 self.get\_figure\_type(),  
 self.fc.color\_property,  
 self.side,  
 self.square()  
 )