**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Радиотехнический»

Кафедра «Информатика и вычислительная техника»

Курс «Парадигмы и конструкции языков программирования»

Отчет по лабораторной работе №1

Тема «Объектно-ориентированные возможности языка Python»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы РТ5-31Б: |  | преподаватель каф. ИУ5 |
| Ильина К.А. |  | Гапанюк Ю.Е. |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2023 г.

**Постановка задачи**

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.

6. Класс «Цвет фигуры» содержит свойство для описания цвета

геометрической фигуры. Подробнее про описание свойств Вы можете

прочитать здесь.

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 должно задаваться в виде поля данных класса и

возвращаться методом класса

10.В корневом каталоге проекта создайте файл main.py для тестирования

Ваших классов 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.

**Текст программы**

**main.py**

from lab\_python\_oop.rectangle import Rectangle

from lab\_python\_oop.square import Square

from lab\_python\_oop.circle import Circle

from art import \*

rectangle1 = Rectangle(10,10, "синего")

print(rectangle1)

for i in range(5):

  aprint("coffee", number=5)

circle1 = Circle(10, "зелёного")

print(circle1)

print('     ', end='')

aprint("coffee", number=1,)

aprint("coffee", number=3)

print('     ', end='')

aprint("coffee", number=1)

square1 = Square(5, "красного")

print(square1)

for i in range(7):

  aprint("coffee", number=5)

**circle.py**

from math import \*

from lab\_python\_oop.GeometricFigure import GeometricFigure

from lab\_python\_oop.Color import Color

class Circle(GeometricFigure):

  name = "Круг"

  def \_\_init\_\_(self, radius, color):

      self.radius = radius

      self.color = Color(color)

  def area(self):

      return round(self.radius\*\*2 \* pi, 3)

  def Get\_name(self):

    return self.name

  def \_\_repr\_\_(self):

    return '{} {} цвета радиусом {} и площадью {}.'.format(

      self.Get\_name(),

      self.color.Get\_color(),

      self.radius,

      self.area()

  )

**Color.py**

class Color:

  def \_\_init\_\_(self, color):

      self.color = color

  def Get\_color(self):

    return self.color

**GeometrcicFigure.py**

from abc import ABC, abstractmethod

class GeometricFigure:

  @abstractmethod

  def area(self):

      pass

**\_\_init\_\_.py**

**rectangle.py**

from lab\_python\_oop.GeometricFigure import GeometricFigure

from lab\_python\_oop.Color import Color

class Rectangle(GeometricFigure):

  name = "Прямоугольник"

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

      self.width = width

      self.height = height

      self.color = Color(color)

  def area(self):

    return self.width \* self.height

  def Get\_name(self):

    return self.name

  def \_\_repr\_\_(self):

    return '{} {} цвета шириной {}, высотой {} и площадью {}.'.format(

        self.Get\_name(),

        self.color.Get\_color(),

        self.width,

        self.height,

        self.area()

    )

**square.py**

from lab\_python\_oop.rectangle import Rectangle

class Square(Rectangle):

  name = "Квадрат"

  def \_\_init\_\_(self, side, color):

      self.side = side

      super().\_\_init\_\_(self.side, self.side, color)

  def area(self):

      return self.side\*\*2

  def \_\_repr\_\_(self):

    return '{} {} цвета со стороной {} и площадью {}.'.format(

      self.Get\_name(),

      self.color.Get\_color(),

      self.side,

      self.area()

  )

**Выполнение программы**

