**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет информатика и системы управления

Кафедра системы обработки информации и управления

Курс «Парадигмы и конструкции языков программирования»

Отчет по лабораторной работе №2

«Объектно-ориентированные возможности языка Python»

Вариант 21

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-32Б: |  | преподаватель каф. ИУ5 |
| Угрюмов Н.А. |  | Нардид А.Н. |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2025 г.

**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**

* **Файл \_\_init\_\_.py** (в пакете lab\_python\_oop):

*# Делает папку пакетом*

* **Файл circle.py** (в пакете lab\_python\_oop):

from lab\_python\_oop.geom\_figure import GeomFigure  
from lab\_python\_oop.color import FigureColor  
  
from math import pi  
  
class Circle(GeomFigure):  
 name = "круг"  
  
 def \_\_init\_\_(self, radius, color: str):  
 self.radius = radius  
 self.color = FigureColor(color)  
  
 def area(self):  
 return pi \* self.radius\*\*2  
  
 def \_\_repr\_\_(self):  
 return "Фигура: {0}, радиус: {1}, цвет: {2}, площадь: {3}".format(  
 self.get\_name(),  
 self.radius,  
 self.color.color,  
 str(self.area().\_\_round\_\_(2))  
 )

* **Файл color.py** (в пакете lab\_python\_oop):

class FigureColor:  
 def \_\_init\_\_(self, color = "бесцветный"):  
 self.\_color = color  
  
 @property  
 def color(self):  
 return self.\_color  
  
 @color.setter  
 def color(self, new\_color: str):  
 self.\_color = new\_color  
  
 def \_\_repr\_\_(self):  
 return f"{self.\_color}"

* **Файл geom\_figure.py** (в пакете lab\_python\_oop):

from abc import ABC, abstractmethod  
  
class GeomFigure(ABC):  
 *# Абстрактный класс для геометрических фигур* name = "Геометрическая фигура"  
  
 def \_\_init\_\_(self, color: str):  
 self.color = color  
  
 @abstractmethod  
 def area(self):  
 pass  
  
 def get\_name(self):  
 return self.name

* **Файл rectangle.py** (в пакете lab\_python\_oop):

from lab\_python\_oop.geom\_figure import GeomFigure  
from lab\_python\_oop.color import FigureColor  
  
class Rectangle(GeomFigure):  
 name = "прямоугольник"  
  
 def \_\_init\_\_(self, width, height, color: str):  
 self.width = width  
 self.height = height  
 self.color = FigureColor(color)  
  
 def area(self):  
 return self.height \* self.width  
  
 def \_\_repr\_\_(self):  
 return "Фигура: {0}, ширина: {1}, высота: {2}, цвет: {3}, площадь: {4}".format(  
 self.get\_name(),  
 self.width,  
 self.height,  
 self.color.color,  
 str(self.area())  
 )

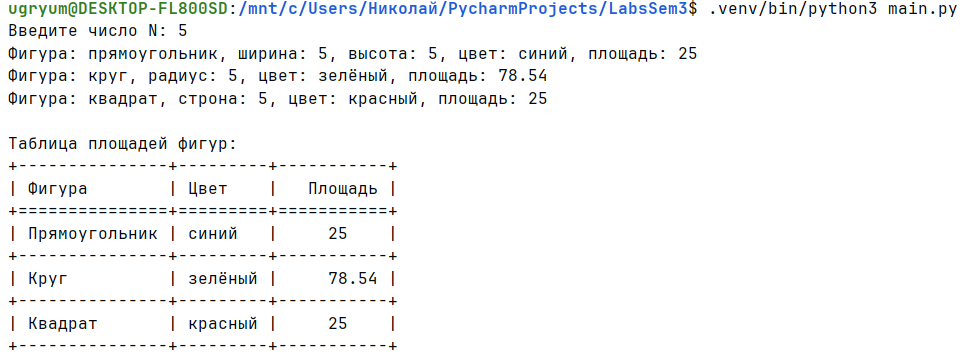
* **Файл square.py** (в пакете lab\_python\_oop):

from lab\_python\_oop.color import FigureColor  
from lab\_python\_oop.rectangle import Rectangle  
  
class Square(Rectangle):  
 name = "квадрат"  
  
 def \_\_init\_\_(self, side, color: str):  
 super().\_\_init\_\_(side, side, color) *# Вызов конструктора базового класса* self.side = side  
  
 def area(self):  
 return self.side \*\* 2  
  
 def \_\_repr\_\_(self):  
 return "Фигура: {0}, строна: {1}, цвет: {2}, площадь: {3}".format(  
 self.get\_name(),  
 self.side,  
 self.color.color,  
 str(self.area())  
 )

* **Файл main.py:**

from lab\_python\_oop.rectangle import Rectangle  
from lab\_python\_oop.circle import Circle  
from lab\_python\_oop.square import Square  
  
from tabulate import tabulate  
  
def main():  
 N = int(input("Введите число N: "))  
  
 rectangle = Rectangle(N, N, "синий")  
 circle = Circle(N, "зелёный")  
 square = Square(N, "красный")  
  
 print(rectangle)  
 print(circle)  
 print(square)  
  
 table = [  
 ["Прямоугольник", rectangle.color.color, rectangle.area()],  
 ["Круг", circle.color.color, f"{circle.area():.2f}"],  
 ["Квадрат", square.color.color, square.area()]  
 ]  
  
 print("\nТаблица площадей фигур:")  
 print(tabulate(table, headers=["Фигура", "Цвет", "Площадь"], tablefmt="grid"))  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

**Скриншоты работы приложения**



**Рис. 1:** работа программы, вывод таблицы площадей

**Ветка с ЛР в моём GitHub:** <https://github.com/ugryum1/LabsSem3/tree/lab2>