

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

**Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»**

**Курс «Парадигмы и конструкции языков программирования»
Отчет по лабораторной работе 2.**

Выполнил:
студент группы ИУ5-33Б
Иванов Николай
Подпись и дата:

Проверил:
преподаватель каф. ИУ5
Нардид А.Н.
Подпись и дата:

Москва, 2023 г.

Текст задания

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](#).
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 («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`.

Текст программы

Abstract.py

```
import abc

class figure(abc.ABC):
    def square(self):
        pass
```

```
def repr(self):
    pass

def get_name(self):
    pass
```

circle.py

```
from python_lab_oop import abstract , color
from math import pi

class circle(abstract.figure):
    def __init__(self, rad = 10, color_ = 'Красный '):
        self.__name = 'Круг'
        self.__color = color.color(color_)
        self.__rad = rad

    def square(self):
        res = pi * self.__rad ** 2
        return res

    def get_name(self):
        return self.__name

    def get_color(self):
        return self.__color

    def repr(self):
        print(f'Название фигуры: {self.get_name()}\n'
              f'Цвет фигуры: {self.get_color()}\n'
              f'Площадь фигуры: {self.square()}\n'
              + '\n'
              )
```

color.py

```
class color:
    def __init__(self, color_='Красный'):
        self.color = color_

    def __str__(self):
        return self.color
```

kvadrat.py

```
from python_lab_oop import abstract, color, rectangle

class kvadrat(rectangle.rectangle):
    def __init__(self, length = 10, color_ = 'Красный'):
        rectangle.rectangle.__init__(self, length=length, color_=color_)
        self.__name = 'Квадрат'

    def square(self):
        res = self._length ** 2
        return res

    def get_name(self):
```

```

        return self.__name

    def get_color(self):
        return self._color

    def repr(self):
        print(f'Название фигуры: {self.get_name()}\n'
              f'Цвет фигуры: {self.get_color()}\n'
              f'Площадь фигуры: {self.square()}\n'
              + '\n'
              )

```

rectangle.py

```

from python_lab_oop import abstract , color

class rectangle(abstract.figure):
    def __init__(self, height = 0, length = 0, color_ = 'Красный'):
        self._length = length
        self.__height = height
        self._color = color.color(color_)
        self._name = 'Прямоугольник'

    def square(self):
        res = self.__height * self._length
        return res

    def get_name(self):
        return self._name

    def get_color(self):
        return self._color

    def repr(self):
        print(f'Название фигуры: {self.get_name()}\n'
              f'Цвет фигуры: {self.get_color()}\n'
              f'Площадь фигуры: {self.square()}\n'
              + '\n'
              )

```

main.py

```

from python_lab_oop import abstract , circle , kvadrat , rectangle
import pygame

def main():
    circle_ = circle.circle(15 , 'Синий')
    circle_.repr()

    kvadrat_ = kvadrat.kvadrat(10 , 'Желтый')
    kvadrat_.repr()

    rectangle_ = rectangle.rectangle(5, 20 , 'Черный')
    rectangle_.repr()

```

```
if __name__ == '__main__':  
    main()
```

Результат работы программы

```
Название фигуры: Круг  
Цвет фигуры: Синий  
Площадь фигуры: 706.8583470577034  
  
Название фигуры: Квадрат  
Цвет фигуры: Желтый  
Площадь фигуры: 100  
  
Название фигуры: Прямоугольник  
Цвет фигуры: Черный  
Площадь фигуры: 100  
  
Process finished with exit code 0
```

