

Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»
Кафедра «Автоматизированные системы обработки
информации и управления»
Дисциплина «Разработка интернет-приложений»

Отчёт по лабораторной работе №2

Выполнил:

Якубов А. Р.

ИУ5-53Б

Преподаватель:

Гапанюк Ю. Е

Москва, 2020 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 "get", 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

Main.py

```
from lab_python_oop import circle, square, rect, color
from lab_python_oop.ac import GF
from loguru import logger
RED = color.Color("Red")
BLUE = color.Color("Blue")
GREEN = color.Color("Green")

def printfigure(figure: GF):
    print(f"{figure}\nArea is {figure.area()}")

if __name__ == "__main__":
    r = rect.Rect(25, 25, BLUE)
    printfigure(r)
    c = circle.Circle(25, GREEN)
    printfigure(c)
    s = square.Square(25, RED)
    printfigure(s)
    logger.debug("Hello, cool debugger")
```

AC.py

```
import abc

class GF(abc.ABC):
    @abc.abstractmethod
    def area(self):
        pass
```

Circle.py

```
from lab_python_oop.ac import GF
from math import pi

class Circle(GF):
    def __init__(self, radius, color):
        self.radius = radius
        self.color = color

    def area(self):
        return self.radius * self.radius * pi

    def __str__(self):
        return f"{self.color} colored circle with {self.radius} radius "
```

Rect.py

```
from lab_python_oop.ac import GF
```

```

class Rect (GF) :
    def __init__(self, a, b, color):
        self.a = a
        self.b = b
        self.color = color

    def area(self):
        return self.a * self.b

    def __str__(self):
        return f"{self.color} colored rectangle with {self.a} and {self.b}
parameters "

```

Square.py

```

from lab_python_oop.rect import Rect

```

```

class Square(Rect):
    def __init__(self, a, color):
        self.a = a
        self.b = a
        self.color = color

    def __str__(self):
        return f"{self.color} colored square with {self.a} parameter "

```

Color.py

```

class Color:
    def __init__(self, color):
        self.color = color

    def check(self):
        print(self.color)
        return f"Color of this figure is {self.color}"

    def __str__(self):
        return self.color

```

Примеры выполнения работы программы

Blue colored rectangle with 25 and 25 parameters

Area is 625

Green colored circle with 25 radius

Area is 1963.4954084936207

Red colored square with 25 parameter

Area is 625

2020-09-22 01:19:04.773 | DEBUG | __main__:<module>:20 - Hello, cool debugger