



Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

Отчёт по лабораторной работе №2 по курсу
«Разработка интернет-приложений»

Тема работы: «Объектно-ориентированные возможности языка
Python»

Выполнила: Попова Дарья, РТ5-51Б

Проверил: _____

27 октября 2020 г.

ЗАЧТЕНО / НЕ ЗАЧТЕНО _____

(подпись)

Москва, 2020

Задание: создать несколько файлов с программами на языке программирования Python, все, кроме основного файла `main.py` хранятся в пакете `lab_python_oop`. Необходимо реализовать классы:

- «Геометрическая фигура» (с абстрактным методом вычисления площади) – класс `GeometricShape`
- «Цвет фигуры» (со свойством для описания цвета через декоратор `@property`) – класс `ShapeColour`
- «Прямоугольник» (наследуется от класса «Геометрическая фигура», переопределяет метод вычисления площади, создаёт объект класса «Цвет фигуры» для хранения цвета; на вход подаются два числовых параметра – ширина и высота, а также строка с цветом) – класс `Rectangle`
- «Круг» (наследуется от класса «Геометрическая фигура», переопределяет метод вычисления площади, создаёт объект класса «Цвет фигуры» для хранения цвета; на вход подаются радиус круга и строка с цветом) – класс `Circle`
- «Квадрат» (наследуется от класса «Прямоугольник», переопределяет метод вычисления площади, создаёт объект класса «Цвет фигуры» для хранения цвета; на вход подаются сторона квадрата и строка с цветом) – класс `Square`

Выполнение задания:

файл `GeometricShape.py`

```
from abc import ABC, abstractmethod

class GeometricShape(ABC):
    @abstractmethod
    def square(self):
        pass

    @classmethod
    def GetShapeType(cls):
        return cls.shapeType

    def __repr__(shapeObj):
        return "Полученная фигура - это {0}, имеющий {1} цвет и площадь"
```

```

{2}."\
        "Площадь была получена из
измерений:".format(shapeObj.GetShapeType(), shapeObj.Colour.ShapeColour,
shapeObj.square())

@staticmethod
def checkType(measure):
    try:
        value = float(measure)
    except ValueError:
        isNum = False
    if isNum is False:
        print("Введите числовое значение измерения!")
    else:
        return value

```

файл Colour.py

```

class ShapeColour:
    @property
    def colour(self):
        return self.colour

    @colour.setter
    def colour(self, colourValue):
        self.colour = colourValue

```

файл Rectangle.py

```

from lab_python_oop.GeometricShape import GeometricShape
from lab_python_oop.Colour import ShapeColour

class Rectangle(GeometricShape):
    shapeType = "прямоугольник"

    @classmethod
    def GetShapeType(cls):
        return GeometricShape.GetShapeType(cls)

    def __init__(self, **kwargs):
        self.colour = ShapeColour.colour
        self.width = GeometricShape.checkType(kwargs)
        self.height = GeometricShape.checkType(kwargs)

    def square(self):
        return self.height * self.width

    def __repr__(self):
        return GeometricShape.__repr__(self) + "\nВысота = {0}\nШирина = {1}".format(self.height, self.width)

```

файл Square.py

```

from lab_python_oop.Rectangle import Rectangle
from lab_python_oop.GeometricShape import GeometricShape
from lab_python_oop.Colour import ShapeColour

```

```

class Square(Rectangle):
    def __init__(self, **kwargs):
        self.colour = ShapeColour.colour
        self.side = GeometricShape.checkType(kwargs)

    def square(self):
        return self.side ** 2

    def __repr__(self):
        return GeometricShape.__repr__(self) + "\nДлина стороны квадрата = {0}".format(self.side)

```

файл Circle.py

```

from lab_python_oop.GeometricShape import GeometricShape
from lab_python_oop.Colour import ShapeColour
from math import pi

class Circle(GeometricShape):
    shapeType = "круг"

    @classmethod
    def GetShapeType(cls):
        return GeometricShape.GetShapeType(cls)

    def __init__(self, **kwargs):
        self.colour = ShapeColour.colour
        self.radius = GeometricShape.checkType(kwargs)

    def square(self):
        return pi * (self.radius ** 2)

    def __repr__(self):
        return GeometricShape.__repr__(self) + "\nРадиус = {0}".format(self.radius)

```

файл main.py

```

from lab_python_oop.Rectangle import Rectangle
from lab_python_oop.Circle import Circle
from lab_python_oop.Square import Square

if __name__ == '__main__':
    print("Л/р № 2 выполнила Попова Дарья Алексеевна, студентка группы РТ5-51Б")
    print("Введите параметры круга: ")
    circle = Circle(float(input()), input())
    print('Проверка введенных данных:', circle)
    print("Введите параметры квадрата: ")
    square = Square(float(input()), input())
    print('Проверка введенных данных:', square)
    print("Введите параметры прямоугольника: ")
    rectangle = Rectangle(float(input()), float(input()), input())
    print('Проверка введенных данных:', rectangle)

```

Скрин с выполнением программы:

Л/р № 2 выполнила Попова Дарья Алексеевна, студентка группы РТ5-51Б
Введите параметры круга:

16

красный

Проверка введенных данных:

Полученная фигура - это круг с площадью 804.247719318987.

Цвет - красный

Радиус = 16.0

Введите параметры квадрата:

12

лиловый

Проверка введенных данных:

Полученная фигура - это квадрат с площадью 144.0.

Цвет - лиловый

Длина стороны квадрата = 12.0

Введите параметры прямоугольника:

13

18

бордовый

Проверка введенных данных:

Полученная фигура - это прямоугольник с площадью 234.0.

Цвет - бордовый

Высота = 13.0

Ширина = 18.0