

Московский государственный технический университет им. Н.Э. Баумана

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

**Курс «Базовые компоненты интернет-технологий»
Отчет по лабораторной работе №1**

Выполнил:
студент группы ИУ5-31Б
Егошин Дмитрий
Павлович

Подпись: _____

Дата: _____

Проверил:
преподаватель каф. ИУ5
Гапанюк Юрий
Евгеньевич

Подпись: _____

Дата: _____

Москва, 2021 г.

Лабораторная работа №2

Описание задания

Разработать программу для решения биквадратного уравнения.

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 "getr", 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

1.circle.py

```
from lab_python_oop.figure import Figure
from lab_python_oop.color import FigureColor
import math

class Circle(Figure):
    "круг, наследуется от геометрической фигуры"
    FigureType = "Круг"

    @classmethod
    def gettype(cls):
        return cls.FigureType

    def __init__(self, color, r):
        """класс, содержащий цвет и радиус"""
        self.r = r
        self.fc = FigureColor()
        self.fc.colorproperty = color

    def square(self):
        return math.pi*(self.r**2)

    def __repr__(self):
        return '{} {} цвета радиусом {} площадью {}'.format(
            Circle.gettype(),
            self.fc.colorproperty,
            self.r,
            self.square()
        )
```

2.color.py

```
class FigureColor:
    """цвет фигуры"""
    def __init__(self):
        self.color = None

    @property
    def colorproperty(self):
        """getter"""
        return self.color
```

```

@colorproperty.setter
def colorproperty(self, value):
    """setter"""
    self.color = value

```

3.figure.py

```

from abc import ABC, abstractmethod

class Figure(ABC):
    """абстрактный класс геометрической фигуры"""
    @abstractmethod
    def square(self):
        """виртуальный метод для вычисления площади фигуры"""
    pass

```

4.rectangle.py

```

from lab_python_oop.figure import Figure
from lab_python_oop.color import FigureColor

class Rectangle(Figure):
    """класс прямоугольник наследуется от геометрической фигуры"""
    FigureType = "Прямоугольник"

    @classmethod
    def gettype(cls):
        return cls.FigureType

    def __init__(self, color, width, height):
        """содержит конструктор по параметрам: ширина, высота и цвет"""
        self.width = width
        self.height = height
        self.fc = FigureColor()
        self.fc.colorproperty = color

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

    def __repr__(self):
        return '{} {} цвета шириной {} и высотой {} площадью {}'.format(
            Rectangle.gettype(),
            self.fc.colorproperty,
            self.width,
            self.height,
            self.square()
        )

```

5.square.py

```

from lab_python_oop.rectangle import Rectangle

class Square(Rectangle):
    """класс квадрат, наследуется от прямоугольника"""
    FigureType = "Квадрат"

    @classmethod
    def gettype(cls):
        return cls.FigureType

    def __init__(self, color, side):
        """содержит конструктор по параметрам сторона и цвет"""

```

```

        self.side = side
        super().__init__(color, self.side, self.side)

    def __repr__(self):
        return '{} {} цвета со стороной {} площадью {}'.format(
            Square.gettype(),
            self.fc.colorproperty,
            self.side,
            self.square()
        )

```

6. main.py

```

from lab_python_oop.rectangle import Rectangle
from lab_python_oop.circle import Circle
from lab_python_oop.square import Square
from PIL import Image

r = Rectangle('синего', 5, 7.5)
c = Circle('зеленого', 5)
s = Square('красного', 5)
print(r)
print(c)
print(s)

img = Image.open("rab2.jpg")
img.show()

```

Экранные формы с примерами выполнения программы

