

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

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

**Отчет по лабораторной работе №1
«Основные конструкции языка Python»
по дисциплине
«Парадигмы и конструкции языков программирования»**

Выполнил:
студент группы ИУ5-33Б:
Буйдина К.А.
Подпись и дата:

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю.Е.
Подпись и дата:

Москва, 2023 г.

Описание задания:

Задание:

- Необходимо создать виртуальное окружение и установить в него хотя бы один внешний пакет с использованием `pip`.
- Необходимо разработать программу, реализующую работу с классами. Программа должна быть разработана в виде консольного приложения на языке Python 3.
- Все файлы проекта (кроме основного файла `main.py`) должны располагаться в пакете `lab_python_oop`.
- Каждый из нижеперечисленных классов должен располагаться в отдельном файле пакета `lab_python_oop`.
- Абстрактный класс «Геометрическая фигура» содержит абстрактный метод для вычисления площади фигуры. Подробнее про абстрактные классы и методы Вы можете прочитать [здесь](#).
- Класс «Цвет фигуры» содержит свойство для описания цвета геометрической фигуры. Подробнее про описание свойств Вы можете прочитать [здесь](#).
- 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.
- Класс «Круг» создается аналогично классу «Прямоугольник», задается параметр «радиус». Для вычисления площади используется константа `math.pi` из модуля `math`.
- 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.
- 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`.

- Дополнительное задание. Протестируйте корректность работы Вашей программы с помощью модульного теста.

Текст программы:

1. main

```
from lab_python_oop.rectangle import Rectangle
from lab_python_oop.circle import Circle
from lab_python_oop.square import Square
import pyfiglet
import matplotlib.pyplot as plt

def main():
    r1 = Rectangle("синего", 3, 2)
    r2 = Rectangle("зеленого", 4, 3)
    r3 = Rectangle("красного", 5, 4)
    c1 = Circle("синего", 5)
    c2 = Circle("зеленого", 6)
    c3 = Circle("красного", 7)
    s1 = Square("синего", 5)
    s2 = Square("зеленого", 6)
    s3 = Square("красного", 7)

    #Первый способ
    result = pyfiglet.figlet_format("Laba - 2", font="3-d")
    print(result)
    #Второй способ
    f = pyfiglet.Figlet(font='slant')
    ascii_art = f.renderText('programmers rule the world')
    print(ascii_art)

    #График
    squares = [r1.square(), r2.square(), r3.square(),
               c1.square(), c2.square(), c3.square(),
               s1.square(), s2.square(), s3.square()]
    for sq in squares: print(sq)

    fig, ax = plt.subplots()
    #заголовки
    ax.set_title("Squares", fontsize=14)
    x_value = [r1.FIGURE_TYPE[:2] + '1', r2.FIGURE_TYPE[:2] + '2',
               r3.FIGURE_TYPE[:2] + '3',
               c1.FIGURE_TYPE[:2] + '1', c2.FIGURE_TYPE[:2] + '2',
               c3.FIGURE_TYPE[:2] + '3',
               s1.FIGURE_TYPE[:2] + '1', s2.FIGURE_TYPE[:2] + '2',
               s3.FIGURE_TYPE[:2] + '3']

    plt.plot(x_value, sorted(squares))
    plt.xticks(rotation=90)
    plt.show()

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

2. lab_python_oop.rectangle

```
from .color import Color
from .figure import Figure
class Rectangle(Figure):
```

```

"""
Класс «Прямоугольник» наследуется от класса «Геометрическая фигура».
"""
FIGURE_TYPE = "Прямоугольник"

@classmethod
def get_figure_type(cls):
    return cls.FIGURE_TYPE

def __init__(self, color_param, width_param, height_param):
    """
    Класс должен содержать конструктор по параметрам «ширина», «высота» и
    «цвет». В конструкторе создается объект класса «Цвет фигуры» для хранения
    цвета.
    """
    self.width = width_param
    self.height = height_param
    self.fc = Color()
    self.fc.colorproperty = color_param

def square(self):
    """
    Класс должен переопределять метод, вычисляющий площадь фигуры.
    """
    return self.width*self.height

def __repr__(self):
    return '{} {} цвета шириной {} и высотой {} площадью {}.'.format(
        Rectangle.get_figure_type(),
        self.fc.colorproperty,
        self.width,
        self.height,
        self.square()
    )
"""
o = Rectangle("red", 8, 9)
print(Rectangle.FIGURE_TYPE, Rectangle.get_figure_type(),
      o.FIGURE_TYPE, o.get_figure_type(), "\n", repr(o))
"""

```

3. lab_python_oop.circle

```

from .figure import Figure
from .color import Color
import math

class Circle(Figure):
    """
    Класс «Круг» наследуется от класса «Геометрическая фигура».
    """
    FIGURE_TYPE = "Круг"

    @classmethod
    def get_figure_type(cls):
        return cls.FIGURE_TYPE

    def __init__(self, color_param, r_param):
        """
        Класс должен содержать конструктор по параметрам «радиус» и «цвет». В
        конструкторе создается объект класса «Цвет фигуры» для хранения цвета.
        """
        self.r = r_param
        self.fc = Color()

```

```

        self.fc.colorproperty = color_param

    def square(self):
        """
        Класс должен переопределять метод, вычисляющий площадь фигуры.
        """
        return math.pi*(self.r**2)

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

```

4. lab_python_oop.square

```

from .rectangle import Rectangle

class Square(Rectangle):
    """
    Класс «Квадрат» наследуется от класса «Прямоугольник».
    """
    FIGURE_TYPE = "Квадрат"

    @classmethod
    def get_figure_type(cls):
        return cls.FIGURE_TYPE

    def __init__(self, color_param, side_param):
        """
        Класс должен содержать конструктор по параметрам «сторона» и «цвет».
        """
        self.side = side_param
        super().__init__(color_param, self.side, self.side)

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

```

5. lab_python_oop.figure

```

from abc import ABC, abstractmethod

class Figure(ABC):
    """
    Абстрактный класс «Геометрическая фигура»
    """

    @abstractmethod
    def square(self):
        """
        содержит виртуальный метод для вычисления площади фигуры.
        """
        pass

```

6. lab_python_oop.color

```

class Color:
    """

```

[illegible]

