



**Министерство науки и высшего образования 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)**

ФАКУЛЬТЕТ

«Радиотехнический»

КАФЕДРА

ИУ-5 «Системы обработки информации и управления»

Отчет по лабораторной работе № 2 по курсу

Разработка интернет-приложений

**Тема работы: "Объектно-ориентированные
возможности языка Python. "**

Выполнил: Лисин. А. В.

Группа: РТ5-51Б

Дата

выполнения: «27» ноября 2020 г.

Подпись:_____

Проверил: Гапанюк Ю. Е.

Дата

проверки: «27» ноября 2020 г.

Подпись:_____

Москва, 2020 г.

Содержание

Описание задания	3
Ход работы	4
Результат работы	7

Цель лабораторной работы: изучение объектно-ориентированных возможностей языка Python.

Описание задания:

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`.

Ход работы:

Текст circle.py:

```
from lab_python_oop.shape import Shape
from lab_python_oop.color import Color
from math import pi as PI

class Circle(Shape):
    shapeType = "Круг"

    @classmethod
    def GetType(cls):
        return Shape.GetType(cls)

    def __init__(self, argsList):
        print("Записываем радиус...")
        self.radius = Shape.CheckNumValue(argsList[0] if len(argsList) >= 1 else '.')
        print("Записываем цвет...")
        self.color = Color(argsList[1] if len(argsList) >= 2 else '0')
        print("Запись для круга завершена!")

    def Square(self):
        return PI * self.radius * self.radius

    def __repr__(self):
        return Shape.__repr__(self) + "\nРадиус: {}".format(self.radius)
```

Текст color.py:

```
class Color:
    # Статический метод проверки корректности ввода строки цвета
    @staticmethod
    def CheckColorValue(value):
        isNum = not value.isalpha()
        # Если введено некорректное значение, то требуем повторного ввода до тех пор,
        # пока не получим буквенную строку
        while isNum:
            print("Ошибка получения цвета. Введите значение ещё раз")
            value = input()
            isNum = not value.isalpha()
        return value

    def __init__(self, colorValue):
        self.color = Color.CheckColorValue(colorValue)

    # Св-во цвета
    @property
    def Color(self):
        return self.color.lower()

    @Color.setter
    def Color(self, colorValue):
        colorValue = self.CheckColorValue(colorValue)
        self.color = colorValue
```

Текст quadrate.py:

```

from lab_python_oop.shape import Shape
from lab_python_oop.rectangle import Rectangle
from lab_python_oop.color import Color

class Quadrate(Rectangle):
    shapeType = "Квадрат"

    @classmethod
    def GetType(cls):
        return Shape.GetType(cls)

    def __init__(self, argsList):
        print("Записываем сторону...")
        self.dim = Shape.CheckNumValue(argsList[0] if len(argsList) >= 1 else '.')
        print("Записываем цвет...")
        self.color = Color(argsList[1] if len(argsList) >= 2 else '0')
        print("Запись для квадрата завершена!")

    def Square(self):
        return self.dim * self.dim

    def __repr__(self):
        return Shape.__repr__(self) + "\nСторона: {}".format(self.dim)

```

Текст rectangle.py:

```

from lab_python_oop.shape import Shape
from lab_python_oop.color import Color

class Rectangle(Shape):
    shapeType = "Прямоугольник"

    @classmethod
    def GetType(cls):
        return Shape.GetType(cls)

    def __init__(self, argsList):
        print("Записываем ширину...")
        self.width = Shape.CheckNumValue(argsList[0] if len(argsList) >= 1 else '.')
        print("Записываем высоту...")
        self.height = Shape.CheckNumValue(argsList[1] if len(argsList) >= 2 else '.')
        print("Записываем цвет...")
        self.color = Color(argsList[2] if len(argsList) >= 3 else '0')
        print("Запись для прямоугольника завершена!")

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

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

```

Текст shape.py:

```

from abc import ABC, abstractmethod

class Shape(ABC):

```

```

# Абстрактный метод вычисления площади
@abstractmethod
def Square(self):
    pass

# Виртуальный метод возвращения типа
def GetType(cls):
    return cls.shapeType.lower()

# Виртуальный метод форматированного представления всех фигур
def __repr__(shapeObject):
    return "Данная фигура является {0}ом, имеет {1} цвет, а её площадь равняется {2} на основании введённых" \
           " числовых параметров:".format(shapeObject.GetType(), shapeObject.color.Color, shapeObject.Square())

# Статический метод проверки корректности ввода числовых значений (для измерений)
@staticmethod
def CheckNumValue(num):
    try:
        # Проверяем возможность перевода в численный формат
        value = float(num)
        # Если возможно, то возвращаем полученное значение без изменений
    except ValueError:
        # Если преобразование типа невозможно, то требуем повторного ввода до тех пор, пока не получим число
        isNum = False
        while not isNum:
            print("Ошибка получения величины. Введите значение ещё раз")
            try:
                value = float(input())
                isNum = True
            except ValueError:
                pass
        return value

```

Текст main.py:

```

from lab_python_oop.rectangle import Rectangle
from lab_python_oop.circle import Circle
from lab_python_oop.quadrate import Quadrate
import numpy as np

if __name__ == '__main__':
    print("Лисин Андрей Валерьевич, группа РТ5-51Б. Лабораторная работа #2")
    print("Программа проверяет функционал созданных по заданию лабораторной работы модулей.")
    print("Вам будет предложено ввести данные для прямоугольника, круга и квадрата. Ввод для каждого объекта "
          "осуществляется в одну строку, через пробел.")
    print("Введите данные прямоугольника: ")
    rctg = Rectangle(input().split())
    print("Введите данные круга: ")
    crc = Circle(input().split())
    print("Введите данные квадрата: ")
    qdr = Quadrate(input().split())
    print('Итоги ввода: ', rctg, crc, qdr, sep = '3\n')
    print('\n', np.ones((3, 20), float))

```

Экранная форма:

```
[toxe@toxubuntu:~/WAO_Labs/Lab2$ python3 main.py]
Лисин Андрей Валерьевич, группа РТ5-516. Лабораторная работа #2
Программа проверяет функционал созданных по заданию лабораторной работы модулей.
Вам будет предложено ввести данные для прямоугольника, круга и квадрата. Ввод для каждого объекта осуществляется в одну строку, через пробел.
Введите данные прямоугольника:
10 3 красный
Записываем ширину...
Записываем высоту...
Записываем цвет...
Запись для прямоугольника завершена!
Введите данные круга:
10 зеленый
Записываем радиус...
Записываем цвет...
Запись для круга завершена!
Введите данные квадрата:
20 черный
Записываем сторону...
Записываем цвет...
Запись для квадрата завершена!
Итоги ввода: 3
Данная фигура является прямоугольником, имеет красный цвет, а её площадь равняется 30.0 на основании введенных числовых параметров:
Ширина: 10.0
Высота: 3.03
Данная фигура является кругом, имеет зеленый цвет, а её площадь равняется 314.1592653589793 на основании введенных числовых параметров:
Радиус: 10.03
Данная фигура является квадратом, имеет черный цвет, а её площадь равняется 400.0 на основании введенных числовых параметров:
Сторона: 20.0

[[[1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]]]
```