



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: "Классы, методы, поля. Наследование,
полиморфизм, инкапсуляция. Модули. Работа с пакетным
менеджером pip."

Выполнил: Брусов Никита, РТ5-51Б

Проверил: _____

12 сентября 2020 г.

ЗАЧТЕНО / НЕ ЗАЧТЕНО _____

(подпись)

1. Цель лабораторной работы.

Изучение основных конструкций и возможностей ЯП Python при объектно-ориентированном подходе.

2. Задание на лабораторную работу.

Для лабораторной работы №2 необходимо разработать консольное приложение, включающее в себя несколько модулей, представляющих из себя классы с общим абстрактным прародительским классом, тема классов – геометрические фигуры. Требуется разработать абстрактный класс, а так же классы для трёх основных геометрических форм – круга, прямоугольника и квадрата – с реализацией функционала:

- 1) Хранения информации об измерениях фигуры и её цвете для экземпляра класса;
- 2) вычисления площади фигуры для экземпляра класса;
- 3) представления экземпляра класса в удобном для чтения виде;
- 4) статического поля, сообщающего о типе фигуры, для каждого класса.

3. Ход выполнения лабораторной работы.

Начнём с самого абстрактного класса. Реализуем весь его функционал в модуле *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 введенных 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
        return value
    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

```

Следующим шагом создаём класс для работы с цветом фигур, *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
```

Теперь создаём классы для самих фигур:

1) Прямоугольник, *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)
```

2) Круг, *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)

```

3) Квадрат, *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)

```

Все рабочие модули созданы, последний шаг – основная программа для проверки работы, в которую так же будет решён вопрос последнего требования к лабработе – загрузки библиотеки через *pip*. *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("Зоров Владислав Витальевич, группа РТ5-51Б")
    print("Лабработа #2")

```

```

print("Программа проверяет функционал созданных по заданию лабораторной
работы модулей.")
print("Вам будет предложено ввести данные для прямоугольника, круга и
квадрата.")
print("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 = '\n\n')
print('\n', np.ones((3, 20), float))

```

4. Результаты работы

```

Run: main
"D:\Study\5 Семестр\ПИП\Lab2\venv\Scripts\python.exe" "D:/Study/5 Семестр/ПИП/Lab2/main.py"
Забурунов Леонид Вячеславович, группа РТ5-515. Лабработа #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:
3 голые коралловый
Записываем ширину...
Записываем высоту...
Ошибка получения величины. Введите значение ещё раз
число
1.63
Записываем цвет...
Запись для прямоугольника завершена!
Введите данные круга:
10 болотный\
Записываем радиус...
Записываем цвет...
Ошибка получения цвета. Введите значение ещё раз
болотный
Запись для круга завершена!
Введите данные квадрата:
36 персиковый
Записываем сторону...
Записываем цвет...
Запись для квадрата завершена!
Итоги ввода:
Данная фигура является прямоугольником, имеет коралловый цвет, а её площадь равняется 4.89 на основании введённых числовых параметров:
Ширина: 3.0
Высота: 1.63
Данная фигура является кругом, имеет болотный цвет, а её площадь равняется 314.1592653589793 на основании введённых числовых параметров:
Радиус: 10.0
Данная фигура является квадратом, имеет персиковый цвет, а её площадь равняется 1296.0 на основании введённых числовых параметров:
Сторона: 36.0

[[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.]]

Process finished with exit code 0

```

Looks like you're using NumPy
Would you like to turn scientific mode on?
[Use scientific mode](#) [Keep current layout...](#)

41:1 CRLF UTF-8 4 spaces Python 3.8 (Lab2)

