



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)

ФАКУЛЬТЕТ

ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА

СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ

Отчет по лабораторной работе №2

«Объектно-ориентированные возможности»
по курсу «Разработка интернет приложений»

Студент: Федорова Антонина

Группа: ИУ5-53

_____ 23.09.2018

Задание лабораторной работы

1. Программа должна быть разработана в виде консольного приложения на языке Python 3.
2. Все файлы проекта (кроме основного файла main.py) должны располагаться в пакете lab_python_oop.
3. Каждый из нижеперечисленных классов должен располагаться в отдельном файле пакета lab_python_oop.
 1. Абстрактный класс «Геометрическая фигура» содержит абстрактный метод для вычисления площади фигуры. Подробнее про абстрактные классы и методы Вы можете прочитать здесь - <https://docs.python.org/3/library/abc.html>
 2. Класс «Цвет фигуры» содержит свойство для описания цвета геометрической фигуры. Подробнее про описание свойств Вы можете прочитать здесь - <https://docs.python.org/3/library/functions.html#property>
 3. 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.
 4. Класс «Круг» создается аналогично классу «Прямоугольник», задается параметр «радиус». Для вычисления площади используется константа math.pi из модуля math - <https://docs.python.org/3/library/math.html>
 5. Класс «Квадрат» наследуется от класса «Прямоугольник». Класс должен содержать конструктор по длине стороны.
 6. Для классов «Прямоугольник», «Квадрат», «Круг»:
 1. Определите метод "get", который возвращает в виде строки основные параметры фигуры, ее цвет и площадь. Используйте метод format - <https://pyformat.info/>
 2. Название фигуры («Прямоугольник», «Квадрат», «Круг») должно задаваться в виде поля данных класса и возвращаться методом класса.
4. 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:
 1. Прямоугольник синего цвета шириной 3 и высотой 2.
 2. Круг зеленого цвета радиусом 5.
 3. Квадрат красного цвета со стороной 5.

Исходный код:

Файл main.py

```
from lab_python_oop import rectangle
from lab_python_oop import square
from lab_python_oop import circle
from lab_python_oop import colour
import sys

def menu():
    print("Площадь чего вы хотите посчитать?")
    print("1) Прямоугольник")
    print("2) Квадрат")
    print("3) Круг")
    print("4) Выход")
    result = input()
    return result

work = True
# Обработка вводимых значений и вызов методов классов
while work:
    res = menu()
    if (res >= '1') and (res <= '4'):
```

```

# Прямоугольник
if res == '1':

    work1 = True
    while work1:
        print("Введите высоту:")
        try:
            a = int(input())
            work1 = False
        except (TypeError, ValueError):
            print("Введите число!")

    work1 = True
    while work1:
        print("Введите ширину:")
        try:
            b = int(input())
            work1 = False
        except (TypeError, ValueError):
            print("Введите число!")

    print("Введите цвет:")
    cool = colour.Colour()
    cool.v = input()
    rec = rectangle.Rectangle(a, b, cool.v)
    print(rec.gn())
    rec.__repr__()

# Квадрат
elif res == '2':
    work1 = True
    while work1:
        print("Введите сторону квадрата:")
        try:
            a = int(input())
            work1 = False
        except (TypeError, ValueError):
            print("Введите число!")

    print("Введите цвет:")
    cool = colour.Colour
    cool.v = input()
    sq = square.Square(a, cool.v)
    print(sq.gn())
    sq.__repr__()

# Круг
elif res == '3':
    print("Введите радиус:")
    work1 = True
    while work1:
        try:
            a = int(input())
            work1 = False
        except (TypeError, ValueError):
            print("Введите число!")

        print("Введите цвет:")
        cool = colour.Colour
        cool.v = input()
        cc = circle.Circle(a, cool.v)
        print(cc.gn())
        cc.__repr__()

elif res == '4':
    work = False

else:
    print("Вы ввели неверное значение!")
sys.exit()

```

Файл __init.py__

```
__all__ = ["figure", "rectangle", "circle", "square", "colour"]
```

Файл Figure.py

```
from abc import ABCMeta, abstractmethod
```

```
class Figure:
    __metaclass__ = ABCMeta

    @abstractmethod
    def __init__(self):
        pass

    @abstractmethod
    def area(self):
        return 0

    @abstractmethod
    def __repr__(self):
        pass
```

Файл Rectangle.py

```
from lab_python_oop import figure
```

```
class Rectangle(figure.Figure):

    def __init__(self, height, width, colour, name="Rectangle"):
        self.height = height
        self.width = width
        self.colour = colour
        self.name = name

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

    def __repr__(self):
        print("Высота прямоугольника = {0}, ширина прямоугольника = {1}".format(self.height, self.width))
        print("Площадь прямоугольника = {0}, цвет прямоугольника = {1}\n".format(self.area(), self.colour))

    def gn(self):
        return self.name
```

Файл Square

```
from lab_python_oop import rectangle
```

```
class Square(rectangle.Rectangle):
    def __init__(self, width, col, name="Square"):
        self.width = width
        self.colour = col
        self.name = name

    def area(self):
        return self.width * self.width

    def __repr__(self):
        print("Сторона квадрата = {0}, площадь квадрата = {1}, цвет квадрата = {2}\n".format(self.width, self.area(), self.colour))

    def gn(self):
        return self.name
```

Файл circle

```
from lab_python_oop import figure
import math
```

```
class Circle(figure.Figure):
```

```

def __init__(self, radius, col, name="Circle"):
    self.colour = col
    self.radius = radius
    self.name = name

def area(self):
    return math.pi * self.radius * self.radius

def __repr__(self):
    print("Сторона круга = {0}, площадь круга = {1}, цвет круга = {2}\n".format(self.radius, self.area(), self.colour))

def gn(self):
    return self.name

```

Файл Colour

```

class Colour:
    def __init__(self):
        self.col = None

    def ge(self):
        return self.col

    def se(self, value):
        self.col = value

    def de(self):
        None

    v = property(ge, se, de, "Цвет")

```

Скриншоты выполнения работы

```

C:\Users\User\PycharmProjects\lab2\venv\Scripts\python.exe C:/Users/User/PycharmProjects/lab2/main.py
Площадь чего вы хотите посчитать?
1) Прямоугольник
2) Квадрат
3) Круг
4) Выход
1
Введите высоту:
3
Введите ширину:
2
Введите цвет:
green
Rectangle
Высота прямоугольника = 3, ширина прямоугольника = 2
Площадь прямоугольника = 6, цвет прямоугольника = green

Площадь чего вы хотите посчитать?
1) Прямоугольник
2) Квадрат
3) Круг
4) Выход
2
Введите сторону квадрата:
5
Введите цвет:
red
Square
Сторона квадрата = 5, площадь квадрата = 25, цвет квадрата = red

Площадь чего вы хотите посчитать?
1) Прямоугольник
2) Квадрат
3) Круг
4) Выход
3
Введите радиус:
7
Введите цвет:
blue
Circle
Сторона круга = 7, площадь круга = 153.93804002589985, цвет круга = blue

```

Обработка неверно введенных значений:

```
Площадь чего вы хотите посчитать?
1) Прямоугольник
2) Квадрат
3) Круг
4) Выход
8
Вы ввели неверное значение!
Площадь чего вы хотите посчитать?
1) Прямоугольник
2) Квадрат
3) Круг
4) Выход
3
Введите радиус:
2.5
Введите число!
Введите цвет:
2
Введите цвет:
red
Circle
Сторона круга = 2, площадь круга = 12.566370614359172, цвет круга = red
```

Завершение работы программы:

```
Площадь чего вы хотите посчитать?
1) Прямоугольник
2) Квадрат
3) Круг
4) Выход
4
Process finished with exit code 0
```

Диаграмма классов:

