



## ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

## СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ

**Отчет по лабораторной работе №2**  
**по дисциплине Базовые компоненты интернет технологии**

## Тема работы: " Объектно-ориентированные возможности языка Python"

**Выполнил:**

Студент группы ИУ5Ц-54Б

Перевощиков Н.Д.

09.11.22 г.

(дата, подпись)

**Проверил:**

## Преподаватель

Канев А.И.

(дата, подпись)

Москва, 2022

## СОДЕРЖАНИЕ ОТЧЕТА

1. Цель лабораторной работы.....	3
2. Описание задания.....	4
3. Текст программы.....	6
4. Результат.....	13

## **1. Цель лабораторной работы**

Изучение объектно-ориентированных возможностей языка Python.

## 2. 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 `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.
10. В корневом каталоге проекта создайте файл `main.py` для тестирования Ваших классов (используйте следующую конструкцию - [https://docs.python.org/3/library/\\_\\_main\\_\\_.html](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`.

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

### 3. Текст программы

#### LAB\_02.py

```
from lab_python_oop.rectangle import Rectangle
from lab_python_oop.circle import Circle
from lab_python_oop.square import Square

import sys
"""Def - это инструкция, позволяющая создавать функцию."""
def get_coef(index, prompt):
    try:
        #Попытаемся прочесть коэффициент из командной строки
        coef_str = sys.argv[index]
    except:
        while True:
            #Вводим с клавиатуры
            print(prompt)
            coef_str = input()
            # Проверка, есть ли минус и 0?
            if (int(coef_str) > 0 and int(coef_str) < 20):
                break
            else:
                print('Ошибка! Введите номер варианта от 1 до 20')
        #Переводим строку в действительное число
        coef = float(coef_str)
    return coef

def main():
    N = get_coef(1, 'Введите номер вашего варианта по списку журнала:')
    if (N > 0 and N < 20):
        r = Rectangle("синего", N, N)
        c = Circle("зеленого", N)
        s = Square("красного", N)
        print(r)
        print(c)
        print(s)
    else:
        print('Ошибка! Введите номер варианта от 1 до 20')

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

Файлы, содержащие отдельных функций, должны располагаться в пакете lab\_python\_oop.

\_\_init\_\_.py

```
class FigureColor:
    """
    Класс «Цвет фигуры»
    """
    def __init__(self):
        self._color = None

    @property
    def colorproperty(self):
        """
        Get-аксесор
        Get - аксесор, который используется для чтения значения из
        внутреннего поля класса
        """
        return self._color

    @colorproperty.setter
    def colorproperty(self, value):
        """
        Set-аксесор
        Set - аксесор, используемый для записи значения во внутреннее поле
        класса.
        """
        self._color = value
```

## Circle.py

```
from .figure import Figure
from .color import FigureColor
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 = FigureColor()
        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()
        )
)
```



## Color.py

```
class FigureColor:
    """
    Класс «Цвет фигуры»
    """
    def __init__(self):
        self._color = None

    @property
    def colorproperty(self):
        """
        Get-аксесор
        """
        return self._color

    @colorproperty.setter
    def colorproperty(self, value):
        """
        Set-аксесор
        """
        self._color = value
```

## Figure.py

```
from abc import ABC, abstractmethod

class Figure(ABC):
    """
    Абстрактный класс «Геометрическая фигура»
    """
    @abstractmethod
    def square(self):
        """
        содержит виртуальный метод для вычисления площади фигуры.
        """
        pass
```

## Rectangle.py

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

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 = FigureColor()
        self.fc.colorproperty = color_param

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

    def __repr__(self):
        """
        Метод __repr__ в Python выдает текстовое или строковое представление
        сущности или объекта.
        """
        return '{} {} цвета шириной {} и высотой {} площадью {}.'.format(
            Rectangle.get_figure_type(),
            self.fc.colorproperty,
            self.width,
            self.height,
            self.square()
        )
```

## Square.py

```
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()
        )
```

## 4. Результат

### PyCharm

```
Введите номер вашего варианта по списку журнала:  
3  
Прямоугольник синего цвета шириной 3.0 и высотой 3.0 площадью 9.0.  
Круг зеленого цвета радиусом 3.0 площадью 28.274333882308138.  
Квадрат красного цвета со стороной 3.0 площадью 9.0.  
  
Process finished with exit code 0
```

### Командная строка

```
D:\Работа\МГТУ им. Н.Э.Баумана\Программирование\Программы\Программы за 5 семестр\LAB_02>python LAB_02.py  
Введите номер вашего варианта по списку журнала:  
3  
Прямоугольник синего цвета шириной 3.0 и высотой 3.0 площадью 9.0.  
Круг зеленого цвета радиусом 3.0 площадью 28.274333882308138.  
Квадрат красного цвета со стороной 3.0 площадью 9.0.
```

```
D:\Работа\МГТУ им. Н.Э.Баумана\Программирование\Программы\Программы за 5 семестр\LAB_02>python LAB_02.py 3  
Прямоугольник синего цвета шириной 3.0 и высотой 3.0 площадью 9.0.  
Круг зеленого цвета радиусом 3.0 площадью 28.274333882308138.  
Квадрат красного цвета со стороной 3.0 площадью 9.0.
```