

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

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Отчет по лабораторной работе №2

«Основные конструкции языка Python»

Вариант 21

Выполнил:

студент группы ИУ5-35Б

Ханифов С.В.

Подпись и дата:

Проверил:

Подпись и дата:

Москва, 2024 г

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`.

Текст программы

```
import math

from lab_python_oop.color import FigureColor

from lab_python_oop.figure import Figure

class Circle(Figure):

    figure_type="circle"

    def __init__(self,radius,color):

        self.fc=FigureColor()

        self.fc.color=color

        self.radius=radius

    @classmethod

    def get_figure_type(cls): return cls.figure_type

    def square(self): return math.pi*math.pow(self.radius,2)

    def __repr__(self): return 'figure: {}; radius: {}; color: {}; square: {};\n'.format(Circle.get_figure_type(),self.radius,self.fc.color,self.square())

class FigureColor:

    def __init__(self):

        self.__color=None

    @property

    def color(self): return self.__color

    @color.setter
```

```

        def color(self,color): self.__color=color

import abc

class Figure(abc.ABC):

    @abc.abstractmethod

    def square():pass

from lab_python_oop.figure import Figure

from lab_python_oop.color import FigureColor


class Rectangle(Figure):

    figure_type="rectangle"

    def __init__(self,width,height,color):

        self.fc=FigureColor()

        self.fc.color=color

        self.width=width

        self.height=height

    @classmethod

    def get_figure_type(cls): return cls.figure_type

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


    def __repr__(self): return 'figure: {}; width: {}; height: {}; color: {}; square:
    {};'.format(Rectangle.get_figure_type(),self.width,self.height,self.fc.color,self.squ
    are())

from lab_python_oop.rectangle import Rectangle

```

```

class Square(Rectangle):
    figure_type="square"

    def __init__(self, side, color):
        self.side=side
        super().__init__(side, side, color)

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

    def __repr__(self): return 'figure: {}; side: {}; color: {}; square:
{};'.format(Square.get_figure_type(),self.side,self.fc.color,self.square())

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

import requests

N = 21

def main():
    rectangle = Rectangle(N, N, "blue")
    circle = Circle(N, "green")
    square = Square(N, "red")

    print('{}\n{}\n{}\n'.format(rectangle,circle,square))

    response = requests.get("https://jsonplaceholder.typicode.com/todos/1")

```

```
print(f"Status code: {response.status_code}")

print(f"Response body: {response.json()}")


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

Вывод:

```
figure: rectangle; width: 21; height: 21; color: blue; square: 441;
figure: circle; radius: 21; color: green; square: 1385.4423602330987;
figure: square; side: 21; color: red; square: 441;
```

Status code: 200

Response body: {'userId': 1, 'id': 1, 'title': 'delectus aut autem',
'completed': False}